

## LAMPIRAN

### Lampiran 1. Lembar Kesediaan Pembimbing

#### SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Yusron Arly Bazarah  
NIM : 20090034  
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama : Muhammad Fikri Hidayattullah, S.T., M.Kom.  
Status : Dosen Tetap  
NIDN : 0623108801  
Jabatan Fungsional : Lektor  
Pangkat/Golongan : Penata Tk. 1/III-d

Pada hari ini Kamis tanggal 14 Maret 2024 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Skripsi Pihak Pertama dengan syarat Pihak Pertama berjanji akan menyelesaikan Skripsi dalam waktu maksimal 3 bulan (awal bulan Juni 2024). Jika syarat tersebut tidak terpenuhi, maka Pihak Kedua berhak untuk tidak melanjutkan proses bimbingan. 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



Yusron Arly Bazarah

Pihak Kedua



Muhammad Fikri Hidayattullah, S.T., M.Kom.

Mengetahui  
Ketua Program Studi Sarjana Terapan Teknik Informatika



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

## SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Yusron Arly Bazarah  
NIM : 20090034  
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama : Dairoh, M.Sc  
Status : Dosen Tetap  
NIDN : 0612108701  
Jabatan Fungsional : Lektor  
Pangkat/Golongan : Penata Tk. 1/III-d

Pada hari ini senin tanggal 25 Maret 2024 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing II 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, 25 Maret 2024

Pihak Pertama



Yusron Arly Bazarah

Pihak Kedua



Dairoh, M.Sc

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 : **Pemilik Bebek Goreng H.Slamet Sultan Agung**  
Yth. : di Kab. Tegal

Dengan hormat, mahasiswa dengan identitas berikut ini:

nama : Yusron Arly Bazarah  
NIM : 20090034  
prodi : Sarjana Terapan Teknik Informatika

Bermaksud melakukan penelitian untuk keperluan Skripsi dengan judul **"APLIKASI RESERVASI MEJA DAN MAKANAN RESTORAN DI KABUPATEN TEGAL BERBASIS WEB"**. 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 Apriyanti S.T., M.Kom**  
NIPY : 09.015.225

### Lampiran 3. Surat Pernyataan HKI

#### SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta:

1. Nama : Yusron Arly Bazarah  
Kewarganegaraan : Indonesia  
Alamat : Jl. Kuwusangka RT 19/03 Kelurahan Pagedangan Kecamatan Adiwerna,  
Kabupaten Tegal, Provinsi Jawa Tengah, 52194
2. Nama : Muhammad Fikri Hidayattullah, S.T., M.Kom  
Kewarganegaraan : Indonesia  
Alamat : Jl. Glatik No.68, Randugunting, Kec. Tegal Selatan, Kota Tegal
2. Nama : Dairoh, M.Sc  
Kewarganegaraan : Indonesia  
Alamat : Jl. Garuda No 3 Rt 02 Rw 09 Randugunting Tegal Selatan Kota Tegal

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan :  
Berupa : Program Komputer  
Berjudul : Risol: Reservasi Restoran *Online*
  - 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:
  - a. permohonan karya cipta yang saya ajukan dianggap ditarik kembali; atau
  - b. 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.



- c. Dalam hal kepemilikan Hak Cipta yang dimohonkan secara elektronik sedang dalam perkara 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 kami buat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Tegal, 01 Juli 2024



(Yusron Arly Bazarah)

(Muhammad Fikri Hidayattullah, S.T., M.Kom)

(Dairoh, M.Sc)

## Lampiran 4. Surat Pengalihan HKI

### SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini:

1. Nama : Yusron Arly Bazarah  
Kewarganegaraan : Indonesia  
Alamat : Jl. Kuwusangka RT 19/03 Kelurahan Pagedangan Kecamatan Adiwerna,  
Kabupaten Tegal, Provinsi Jawa Tengah, 52194
2. Nama : Muhammad Fikri Hidayattullah, S.T., M.Kom.  
Kewarganegaraan : Indonesia  
Alamat : Jl. Glatik No.68, Randugunting, Kec. Tegal Selatan, Kota Tegal
2. Nama : Dairoh, M.Sc  
Kewarganegaraan : Indonesia  
Alamat : Jl. Garuda No 3 Rt 02 Rw 09 Randugunting Tegal Selatan Kota Tegal

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 "Risol: Reservasi Restoran *Online*" 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.

Tegal, 01 Juli 2024  
Pencipta

Pemegang Hak Cipta  
Ketua P3M



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



(Yusron Arly Bazarah)

(Muhammad Fikri Hidayattullah, S.T., M.Kom)



(Dairoh, M.Sc)

## Lampiran 5. Syarat Pengajuan HKI



# Manual Book Risol

Reservasi Restoran Online

Yusron Arly Bazarah  
Muhammad Fikri Hidayattullah, S.T., M.Kom.  
Dairoh, M.Sc



---

## Daftar Isi

Daftar Isi .....	1
Pendahuluan .....	2
Membuka Website Risol .....	3
Tampilan Awal .....	4
Penggunaan Website .....	5
1. Registrasi .....	5
2. Login .....	5
3. Membuat Reservasi Restoran .....	6
4. Daftar Reservasi Saat Ini .....	9
5. Riwayat Reservasi .....	10
6. Mendaftar Sebagai Mitra .....	11
7. Membuat Meja Baru .....	12
8. Mengatur Posisi Meja .....	13
Pemecahan Masalah .....	15

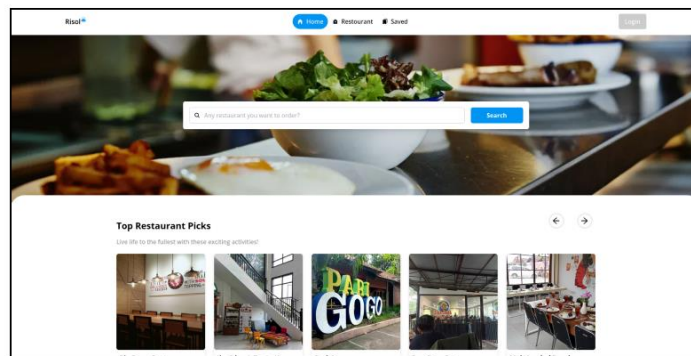
## Pendahuluan

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

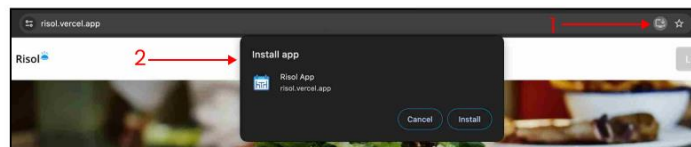


## Membuka Website Risol

1. Buka web browser yang dimiliki, seperti Google Chrome atau Mozilla Firefox.
2. Ketik <https://risol.vercel.app> kemudian tekan *enter*.
3. Halaman pertama [risol.vercel.app](https://risol.vercel.app) adalah halaman beranda seperti berikut.

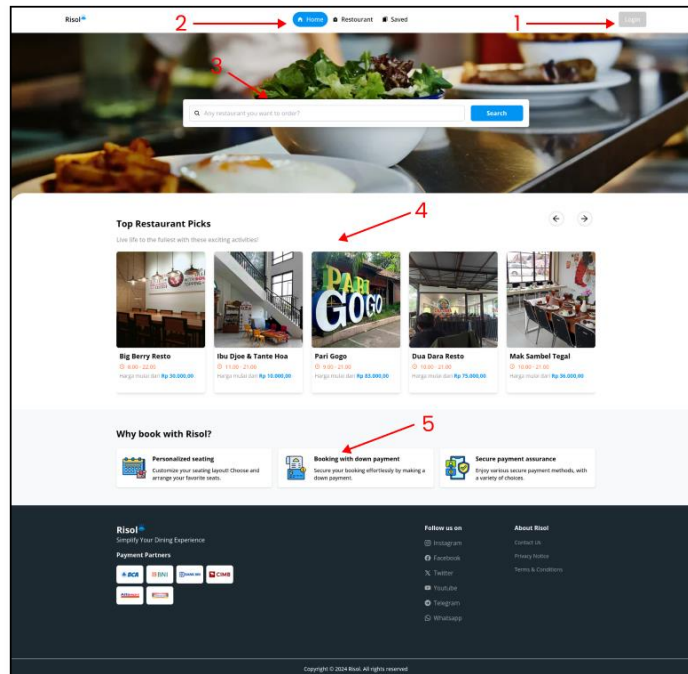


4. Jika menggunakan Google Chrome, pada bagian *address bar* terdapat ikon untuk menginstall aplikasi. Kemudian akan muncul *pop-up* untuk menginstall.



## Tampilan Awal

Saat pertama kali mengunjungi situs web Risol, pengguna akan melihat tampilan seperti berikut, beserta dengan penjelasannya:



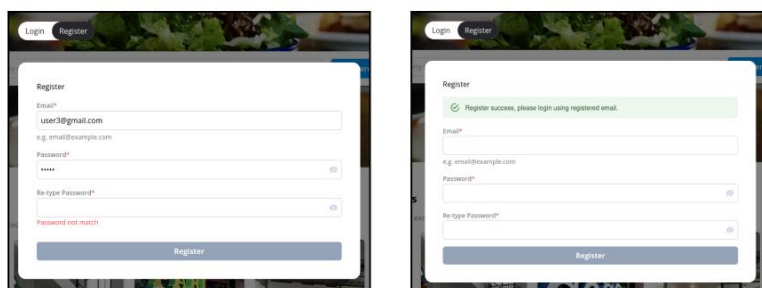
Berikut ini adalah penjelasan beberapa bagian dari halaman utama:

1. Tombol login: Digunakan untuk memunculkan form login sehingga akses fitur penting dalam web dapat dilakukan.
2. Navigation bar: berisi beberapa pilihan yang akan mengarahkan ke masing-masing menu.
3. Search bar: digunakan untuk memunculkan form pencarian sehingga memungkinkan pencarian restoran yang diinginkan.
4. Komponen restoran: menampilkan daftar restoran yang dimiliki oleh website.
5. Komponen reasoning: menampilkan beberapa alasan yang menjadi keunggulan pada website.

# Penggunaan Website

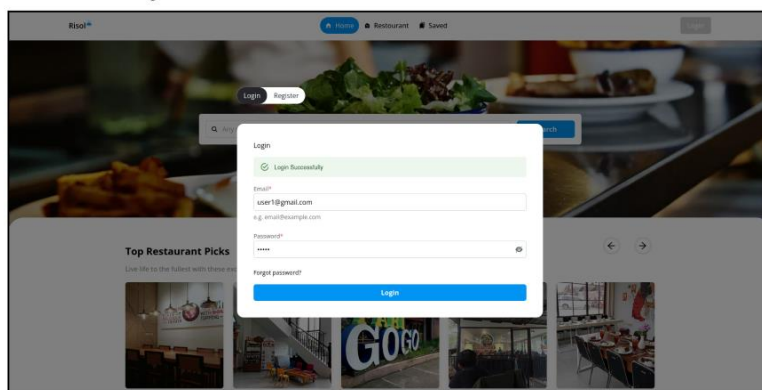
## 1. Registrasi

- Buka website Risol.
- Klik "Login" pada bagian kanan atas.
- Pilih Register.
- Isi form registrasi seperti email dan password.
- Klik "Register" untuk menyelesaikan pendaftaran.



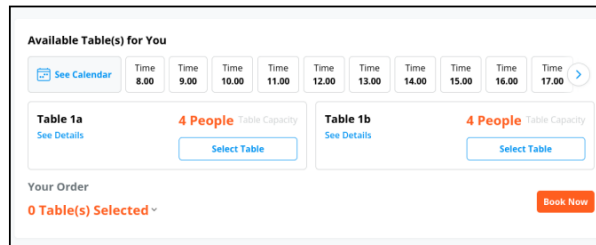
## 2. Login

- Buka website Risol.
- Klik "Login" pada bagian kanan atas.
- Isi form login dengan memasukkan email dan password yang terdaftar.
- Klik "Login".

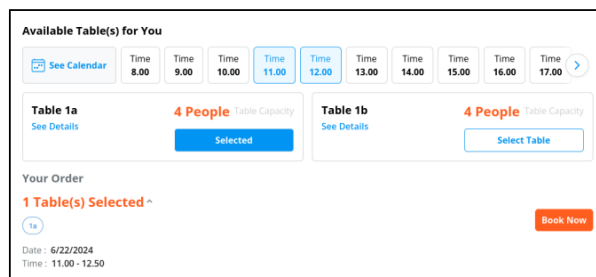


### 3. Membuat Reservasi Restoran

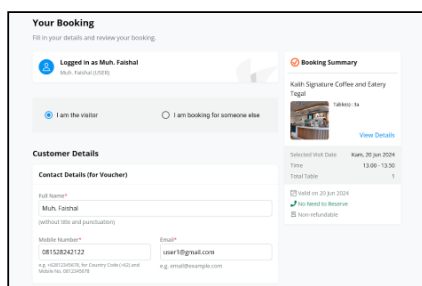
- Pada halaman beranda pilih restoran yang diinginkan. Kemudian akan menuju detail restoran.
- Gulir kebawah hingga sampai pada komponen pengecekan ketersediaan meja.



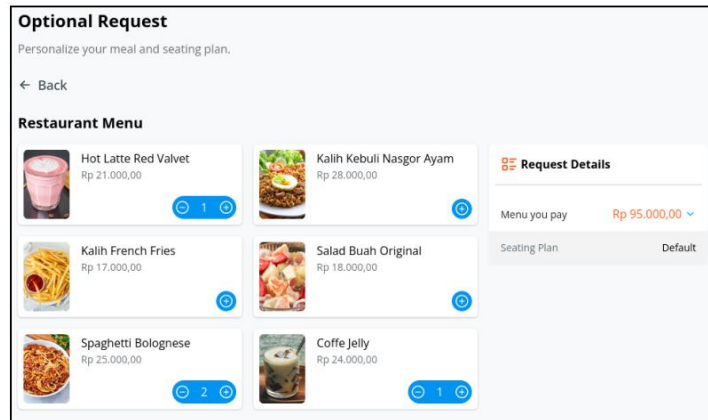
- Klik "See Calendar" untuk memilih tanggal reservasi, kemudian tentukan untuk waktunya dan daftar meja yang tersedia akan ditampilkan. Pilih meja dengan kapasitas yang diinginkan (bisa memilih lebih dari satu meja).



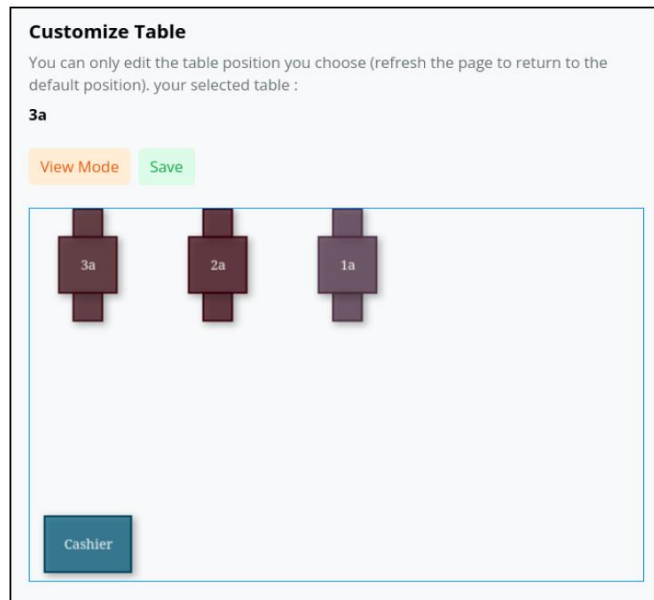
- Klik "Book Now" kemudian akan menuju form reservasi.
- Pada form reservasi, terdapat tiga langkah untuk menyelesaikan proses reservasi. Yang pertama yaitu mengisi data pelanggan.



- Langkah yang kedua merupakan opsional dan bisa dilewati. Disini merupakan form permintaan tambahan jika ingin menyertakan menu saat reservasi.



- Di langkah kedua ini juga bisa melakukan permintaan ketika ingin mengubah posisi tempat duduk yang dipesan.
- Klik "Edit Mode" untuk mengatur tempat duduk lalu klik "Save" untuk menyimpan permintaan tersebut.





- Langkah ketiga adalah memilih metode dan tipe pembayaran. Kemudian klik "Pay" untuk mendapatkan kode reservasi.

**Payment Method**  
How would you like to pay?

← Back

We're holding this price for you! Let's complete your payment in **01:28:31**

**Virtual Account**

- BCA Virtual Account BCA
- BNI Virtual Account BNI
- BRI Virtual Account BRIVA
- CIMB Virtual Account CIMB
- Convenience Store

**Payment Type**

- Full Payment
- Down Payment

**Total Price** Rp 53.500,00

**Pay with BCA Virtual Account**

By continuing to pay, you have agreed to Baidi's Terms & Conditions and Privacy Policy.

**Experience Summary**  
Booking ID: 1144021973

**BOOKING DETAILS**  
**Kalih Signature Coffee and Eatery...**

Valid on: Kam, Jun 20, 2024  
Period: 13.00 - 13.50  
Seating Plan: Default

**CUSTOMER**  
Muhs. Faishal  
081528542122  
user1@gmail.com

- Jika proses berhasil, akan ditampilkan kode untuk membayar biaya reservasi dengan metode pembayaran terkait.

We're holding this price for you! Let's complete your payment in **01:28:33**

**Experience Summary**  
Booking ID: 1718803168605

**Please Transfer to**

**BCA Virtual Account** BCA

You can only transfer from BCA account.

Account Number: 26747049907 Copy

Account Holder Name: Yab Company G558926747

Transfer Amount: Rp 53.500,00 Copy

**How to Transfer**

- m-BCA / BCA Mobile
- Internet Banking
- BCA ATM

**Completed your payment?**

Once your payment is confirmed, we will send your e-booking and receipt in your history.

[Yes, I Have Paid](#)

**BOOKING DETAILS**  
**Kalih Signature Coffee and Eatery...**

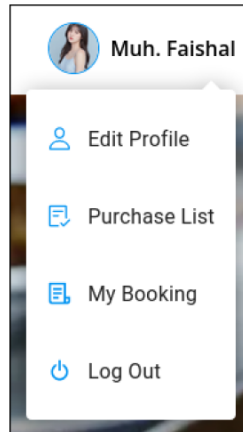
Valid on: Kam, Jun 20, 2024  
Period: 13.00 - 13.50  
Seating Plan: Default

**CUSTOMER**  
Muhs. Faishal  
081528542122  
user1@gmail.com

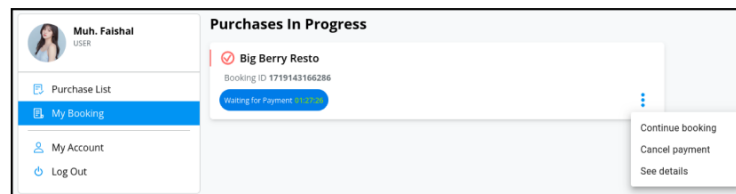
- Proses reservasi selesai, bisa kembali ke beranda.

#### 4. Daftar Reservasi Saat Ini

- Untuk melihat reservasi saat ini, pada bagian navigation bar klik foto profil untuk menampilkan beberapa menu.
- Klik "My Booking".



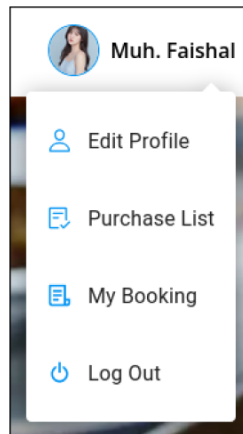
- Pada halaman ini berisi daftar reservasi yang belum terbayar dan memiliki sisa waktu pembayaran.



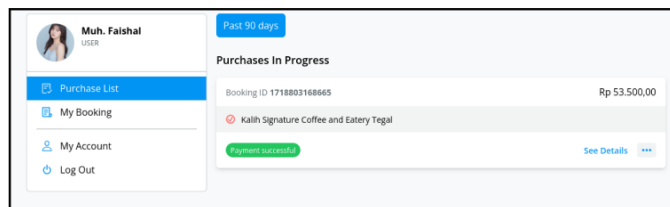
- Terdapat tombol yang berisi beberapa aksi yang bisa diambil, yang pertama melanjutkan pembayaran, kedua membatalkan pemesanan, dan yang ketiga melihat detail pemesanan.
- Memilih "Continue booking" maka akan menuju halaman tata cara melakukan pembayaran.
- Memilih "Cancel payment" akan memunculkan pop-up peringatan tentang pembatalan pesanan.
- Memilih "See details" maka akan ditampilkan detail dari pemesanan.

## 5. Riwayat Pemesanan

- Untuk melihat riwayat pemesanan, pada bagian navigation bar klik foto profil untuk menampilkan beberapa menu.
- Klik "Purchase List".



- Pada halaman ini berisi riwayat pemesanan dengan berbagai status.



Berikut ini adalah beberapa status dan penjelasannya:

1. **Waiting for Payment:** Pembayaran sedang menunggu konfirmasi. Harap segera selesaikan pembayaran untuk melanjutkan reservasi.
2. **Expired:** Waktu pembayaran telah habis. Silakan buat reservasi baru.
3. **Cancelled:** Pembayaran dibatalkan sebelum batas waktu habis.
4. **Payment Success:** Pembayaran berhasil diterima. Reservasi telah dikonfirmasi.
5. **Completed:** Semua proses pembayaran selesai dan dapat memberikan ulasan mengenai pengalaman reservasi.

## 6. Mendaftar Sebagai Mitra

- Pengguna website dapat mendaftarkan restoran mereka sebagai mitra Risol.
- Pada navigation bar pilih menu "Partnership".



- Di halaman ini akan dijelaskan keuntungan dan keunggulan menjadi mitra.
- Untuk melanjutkan, gulir halaman dan tekan tombol "Join Now".



- Kemudian akan diarahkan ke form pendaftaran.

**Become Our Partner**

Who can become a Risol Partner? ▼  
How do I register or join as a Risol partner? ▼  
What additional value that Risol can offer to business partners ▼

**Become Our Partner**  
Reach millions of Risol Xperience users and elevate your business  
[Join Now](#)

**Become Our Partner**

**General Information**

Restaurant Name\*

Contact Person\*

e.g. Mobile No. 0812345678

Operational Hours\*

(Opening) (Closing)

Province\*

District\*

Subdistrict\*

Complete Address\*

RE General information

[Upload image & document](#)

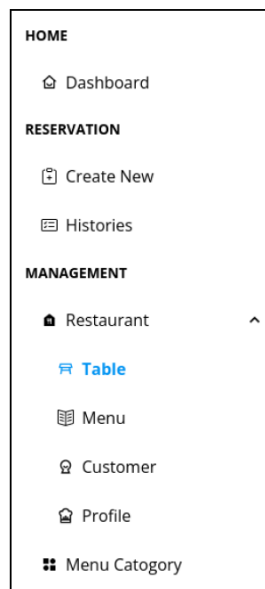
- Isi semua data tersebut lalu klik "Send a Request".
- Pendaftaran selesai, permintaan pendaftaran akan dikonfirmasi oleh admin.

## 7. Membuat Meja Baru

- Jika pengguna website telah terdaftar sebagai owner restoran, mereka bisa menambahkan meja untuk restoran.
- Pada navigation bar pilih menu "Dashboard".



- Jika sudah berada di Dashboard owner, pada navigasi sidebar pilih "Restoran" lalu "Table".



- Di halaman Table, klik "Add Table" untuk menambahkan meja.



A screenshot showing a blue 'Add Table' button at the top left. Below it is a table with the following data:

	Name Table	Capacity	Updated
▼	1b	4 person(s)	4 hari yang lalu
▼	1a	4 person(s)	4 hari yang lalu

At the bottom of the table, there is a footer: 'Rows per page: 5 ▼ Data 1 - 2 of 2'.

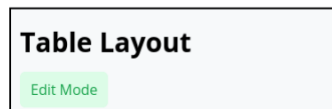


- Kemudian akan muncul pop-up form untuk mengisi data meja.

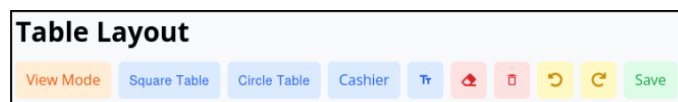
- Isi semua data dan klik "Save" untuk menambahkan meja baru.

## 8. Mengatur Posisi Meja

- Jika pengguna website telah terdaftar sebagai owner restoran, mereka bisa mengatur posisi meja yang dimiliki.
- Pada navigation bar pilih menu "Dashboard".
- Jika sudah berada di Dashboard owner, pada navigasi sidebar pilih "Restoran" lalu "Table".
- Di halaman Table terdapat komponen "Table Layout", yang digunakan untuk mengatur meja restoran.



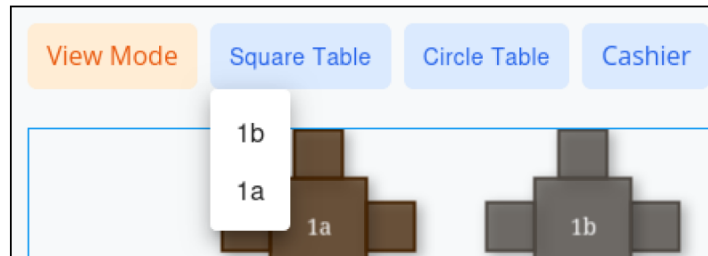
- Klik "Edit Mode" untuk memulai mengatur meja.



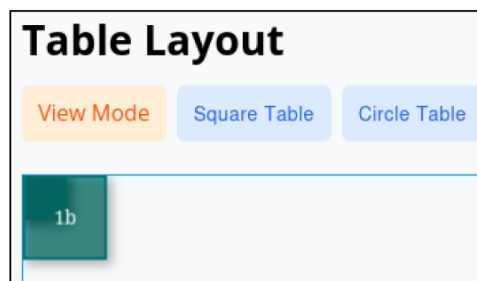
Berikut adalah penjelasan masing-masing tombol:

- Square Table: untuk membuat meja berbentuk persegi.
- Circle Table: untuk membuat meja berbentuk lingkaran.
- Cashier: untuk membuat tempat kasir restoran.

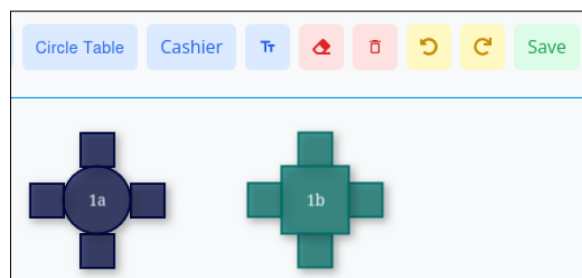
- Pilih "Square Table" atau "Circle Table" untuk meng-generate meja. Ketika salah satu di Klik, maka akan muncul meja dengan label yang dimiliki.



- Jika pilihan label tidak sesuai dengan jumlah label yang dimiliki oleh restoran, maka refresh halaman.
- Setelah memilih label maka akan muncul bentuk-bentuk seperti ini.



- Bentuk-bentuk tersebut bisa digeser dan diposisikan sesuai selera. Jika merasa sudah cukup lalu klik "Save" untuk menyimpan denah posisi meja.



- Setelah proses loading selesai, maka proses menyimpan berhasil.

## Pemecahan Masalah

### 1. Tidak Bisa Login

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

### 2. Website Tidak Merespon

- Periksa Koneksi Internet: Pastikan perangkat Anda terhubung ke internet.
- Lakukan refresh halaman website secara berkala.

### 3. Lainnya

- Kunjungi <https://risol.vercel.app/help> dan ajukan keluhan penggunaan website.



# Dokumen Teknikal Risol

Reservasi Restoran Online

Yusron Arly Bazarah  
Muhammad Fikri Hidayattullah, S.T., M.Kom.  
Dairoh, M.Sc



---

## Daftar Isi

Daftar Isi .....	1
Pendahuluan .....	2
Latar Belakang .....	3
Spesifikasi Teknis .....	4
1. Registrasi .....	5
2. Login .....	6
3. Membuat Reservasi Restoran .....	7
4. Daftar Reservasi Saat Ini .....	18
5. Riwayat Reservasi .....	19
6. Mendaftar Sebagai Mitra .....	20
7. Membuat Meja Baru .....	24
8. Mengatur Posisi Meja .....	26



## Pendahuluan

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

## Latar Belakang

Industri restoran telah mengalami perkembangan yang pesat dalam beberapa tahun terakhir. Pada tahun 2024, Pemerintah juga optimis pertumbuhan ekonomi mampu menyentuh 4,7 hingga 5,5 persen. Kabupaten Tegal juga memiliki potensi bisnis kuliner yang signifikan, restoran dan warung makan memiliki peran penting dalam memperkaya pengalaman kuliner masyarakat setempat dan wisatawan. Namun, sebagian besar dari mereka mungkin masih mengandalkan metode reservasi meja tradisional yang dapat menjadi kendala dalam mengoptimalkan operasional dan memberikan pengalaman terbaik bagi pelanggan.

Cara kerja sistem reservasi tradisional yang dimaksud adalah kunjungan langsung atau panggilan telepon. Biasanya, pelanggan harus menghubungi restoran atau kadang-kadang harus datang langsung ke restoran untuk membuat reservasi. Dimana hal seperti ini memiliki beberapa kelemahan seperti waktu yang terbuang sia-sia karena pelanggan harus pergi ke tempat restoran serta kelemahan informasi seperti meja yang sudah penuh dan tidak dapat menerima pesanan lagi. Kesalahan dalam menu makanan dan minuman serta kesalahan dalam pilihan kursi yang tidak sesuai pesanan. Di sisi lain, restoran juga menghadapi tantangan dalam manajemen reservasi, terutama pada lonjakan pelanggan pada periode tertentu seperti bulan Ramadhan. Sistem reservasi tradisional ini dapat menyebabkan kesalahan dalam penjadwalan dan berpotensi kehilangan pelanggan yang mencari pengalaman yang lebih nyaman.

Berdasarkan permasalahan tersebut, perlu adanya aplikasi reservasi meja dan makanan berbasis web yang dapat digunakan untuk memudahkan kegiatan reservasi serta mengelola restoran sehingga dapat meningkatkan pengalaman pelanggan dan mengoptimalkan operasional restoran. Penggunaan web telah menjadi salah satu alat yang sangat umum digunakan oleh para pengusaha. Pemanfaatan teknologi dalam bentuk website reservasi bukan sekedar penyederhanaan proses pemesanan, melainkan menjadi solusi integral dalam membangun interaksi yang lebih efisien antara pelanggan dan pihak restoran.

# Spesifikasi Teknis

Spesifikasi teknis meliputi :

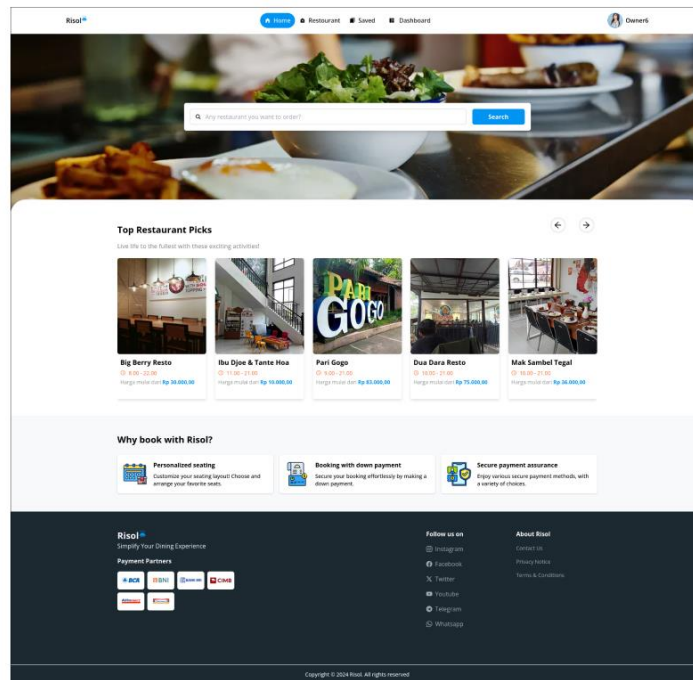
1. Modul Pengguna
2. Source Code

Berikut uraian spesifikasi untuk pembangunan aplikasi :

1. Linux Ubuntu Ram 12 Gb
2. Visual Studio Code
3. Web Browser
4. Postman

Berikut uraian spesifikasi modul :

1. Modul untuk pengguna



Bagian ini adalah bagian yang akan ditampilkan untuk pengguna beserta beragam fitur yang disediakan oleh website seperti membuat reservasi, mencari restoran, mengubah profil dan lainnya.

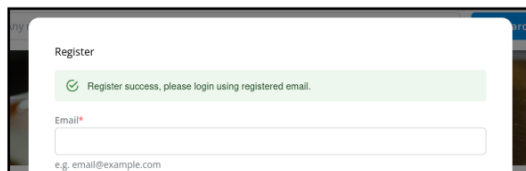
## Source Code

### 1. Registrasi

Proses registrasi akun memungkinkan pengguna baru untuk membuat akun di website "Risol". Dengan akun ini, pengguna dapat mengakses fitur-fitur reservasi restoran.

```
1 export async function POST(request: NextRequest) {
2   const body = await request.json();
3   try {
4     const hashedPassword = await bcrypt.hash(body.password, 10);
5     const isExist = await prisma.user.findUnique({ where: { email: body.email } })
6     if (isExist) return responseBody('Email already used', 400);
7     const user = await prisma.user.create({
8       data: {
9         email: body.email,
10        password: hashedPassword,
11        role: 'USER',
12        image: 'https://res.cloudinary.com/dwtshdu4g/image/upload/v1718630577/
13        awduvprq4fvmffxg@irk.webp',
14        name: formatEmail(body.email)
15      }
16    })
17    if (!user) return responseBody('Create user failed', 401)
18    return responseBody('Success create user', 201, user);
19  } catch (error: any) {
20    return responseBody(error.message, 400);
21  }
22 }
```

Pada kode diatas fungsi register ini menangani request pendaftaran pengguna baru dengan cara menerima dan mengekstrak data dari NextRequest. Password pengguna di-hash menggunakan 'bcrypt' untuk keamanan. Fungsi ini kemudian memeriksa apakah email yang digunakan sudah ada di database menggunakan 'prisma'. Jika email sudah terdaftar, ia mengembalikan pesan error. Jika belum, ia membuat entri pengguna baru dengan email, password yang di-hash, peran sebagai 'USER', gambar profil default, dan nama yang diformat dari email. Setelah sukses, ia mengembalikan pesan keberhasilan dan data pengguna baru. Jika terjadi error, fungsi menangkap dan mengembalikan pesan error yang sesuai.

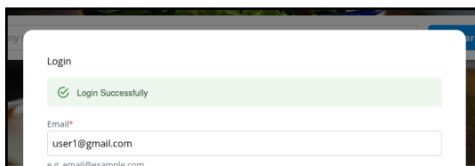


## 2. Login

Proses login memungkinkan pengguna yang sudah terdaftar untuk mengakses akun mereka dan menggunakan fitur-fitur website "Risol". Pada website ini menggunakan library authentication dari Next.js yang populer yaitu next-auth.

```
1 export const authOptions = {
2   adapter: PrismaAdapter(prisma),
3   debug: true,
4   session: { strategy: "jwt" },
5   pages: { error: '/' },
6   secret: "rahasia",
7   providers: [
8     CredentialsProvider({
9       credentials: {
10        email: { label: 'email', type: 'text' },
11        password: { label: 'password', type: 'password' },
12      },
13
14      authorize: async (credential) => {
15        if (!credential?.email || !credential?.password) throw new Error("Credentials not
16        found.");
17
18        const user = await prisma.user.findUnique({ where: { email: credential.email } })
19        if (!user) throw new Error("Login Details Incorrect. Please try again.");
20
21        const decode = await bcrypt.compare(credential.password, user.password)
22        if (!decode) throw new Error("Login Details Incorrect. Please try again.");
23
24        return {
25          id: user.id.toString(),
26          name: user.name,
27          role: user.role,
28          city: user.city || '',
29          email: user.email,
30          image: user.image,
31        }
32      },
33    }),
34  ],
35  callbacks: {
36    jwt: async ({ token, user, session, trigger }) => {
37      if (user) {
38        if (!user) return token
39
40        return {
41          ...token,
42          id: user.id,
43          role: user.role,
44          city: user.city,
45        }
46      }
47      return token;
48    },
49    session: async ({ session, token, user }) => {
50      return {
51        ...session,
52        user: {
53          ...session.user,
54          id: token.id,
55          name: token.name,
56          role: token.role,
57          city: token.city,
58        }
59      };
60    }
61  }
62 }
```

Kode di atas adalah konfigurasi untuk autentikasi dengan penggunaan adapter Prisma untuk integrasi dengan database menggunakan Prisma ORM. Konfigurasi tersebut menetapkan beberapa opsi otentikasi, seperti strategi sesi menggunakan JWT, halaman error khusus, dan rahasia untuk tanda tangan token. Disediakan juga provider otentikasi berbasis kredensial, di mana fungsi 'authorize' memverifikasi email dan password pengguna terhadap data yang ada di database menggunakan Prisma dan bcrypt. Callback 'jwt' digunakan untuk menyimpan informasi pengguna dalam token JWT setelah berhasil login, sementara callback 'session' memperkaya sesi dengan informasi tambahan dari token JWT. Keseluruhan konfigurasi ini memastikan bahwa pengguna dapat login menggunakan email dan password mereka, dengan sesi dan token yang dikelola secara aman dan fleksibel.

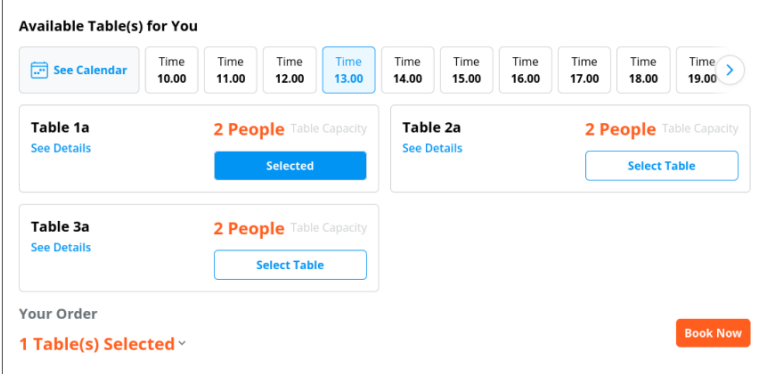


### 3. Membuat Reservasi Restoran

Proses ini memungkinkan pengguna untuk membuat reservasi meja di restoran yang diinginkan melalui aplikasi "Risol".

```
1 const filterHistory = () => {
2   if (!bookingDate) return [];
3
4   const selectedDate = bookingDate.format('YYYY-MM-DD');
5   const filteredHistory = history.filter(data => {
6     const bookingAtDate = dayjs(data.bookingAt).format('YYYY-MM-DD');
7     const bookingAtHour = dayjs(data.bookingAt).hour();
8
9     if (selectedDate !== bookingAtDate) return false;
10
11    if (selectedData.length === 1) {
12      return bookingAtHour === selectedData[0];
13    } else if (selectedData.length > 1) {
14      const minHour = Math.min(...selectedData);
15      const maxHour = Math.max(...selectedData);
16      return bookingAtHour >= minHour && bookingAtHour <= maxHour;
17    }
18
19    return false;
20  });
21
22  return filteredHistory;
23 };
24 const bookedTableIds = filterHistory().flatMap(h => h.bookingTable.map(table => table.id));
25 const filteredTables = tables.filter(table => !bookedTableIds.includes(table.id));
```

Kode diatas adalah fungsi 'filterHistory' menyaring data riwayat pemesanan berdasarkan tanggal dan jam yang dipilih, lalu daftar ID meja yang telah dipesan dari hasil penyaringan ini digunakan untuk memfilter meja yang tersedia, sehingga 'filteredTables' hanya berisi meja-meja yang belum dipesan pada tanggal dan jam tertentu.



The screenshot displays a reservation interface titled "Available Table(s) for You". At the top, there is a "See Calendar" button and a row of time slots from 10.00 to 19.00. The 13.00 slot is selected. Below this, three table options are shown: "Table 1a", "Table 2a", and "Table 3a", each with a "2 People" capacity and a "See Details" link. "Table 1a" has a blue "Selected" button, while "Table 2a" and "Table 3a" have "Select Table" buttons. At the bottom, the "Your Order" section shows "1 Table(s) Selected" and a red "Book Now" button.

Setelah pengguna sudah memilih meja 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, nomor telepon, dan email. Pengguna dapat memesan untuk dirinya sendiri atau memesan untuk orang lain.

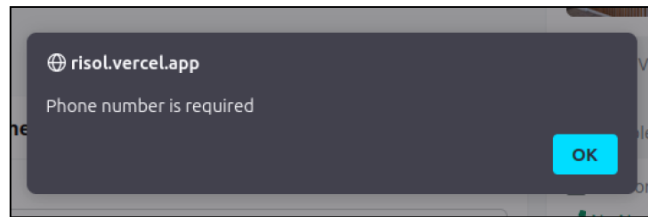
```

1 const nextPageHandler = () => {
2   if (value === 'user') {
3     if (phone === '') return alert('Phone number is required');
4     setBookingData((prevData: any) => {
5       if (prevData) {
6         return {
7           ...prevData,
8           onBehalf: user.user.name,
9           phone: phone,
10          email: user.user.email,
11        }
12      }
13      return prevData
14    })
15  }
16  if (value === 'guest') {
17    if (onBehalf === '') return alert('On behalf name is required');
18    if (phone === '') return alert('Phone number is required');
19    if (email === '') return alert('Email is required');
20
21    setBookingData((prevData: any) => {
22      if (prevData) {
23        return {
24          ...prevData,
25          onBehalf: onBehalf,
26          phone: phone,
27          email: email,
28        }
29      }
30      return prevData
31    })
32  }
33  onHandleNext();
34 }

```

Fungsi 'nextPageHandler' bertanggung jawab untuk memvalidasi data input dan memperbarui data pemesanan berdasarkan peran pengguna, kemudian melanjutkan ke halaman berikutnya. Ketika nilai 'value' adalah 'user', fungsi tersebut memeriksa apakah nomor telepon sudah diisi; jika tidak, fungsi mengeluarkan peringatan. Jika nomor telepon valid, data pemesanan diperbarui dengan nama, nomor telepon, dan email pengguna yang sedang login. Ketika nilai 'value' adalah 'guest', fungsi memvalidasi apakah nama pemesan, nomor telepon, dan email telah diisi. Jika ada yang kosong, fungsi mengeluarkan peringatan yang sesuai. Jika semua data valid, data pemesanan diperbarui dengan nama pemesan, nomor telepon, dan email yang dimasukkan oleh tamu. Setelah proses validasi dan pembaruan selesai, fungsi 'onHandleNext' dipanggil untuk melanjutkan ke langkah berikutnya.

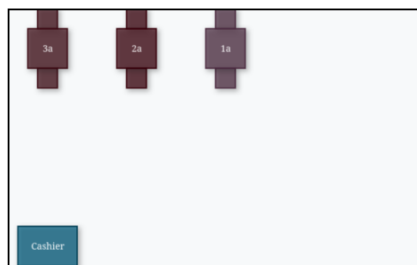




Langkah kedua adalah opsional, namun memberikan fleksibilitas ekstra. Di sini, pengguna dapat langsung melanjutkan ke langkah ketiga atau memilih untuk menambahkan permintaan khusus, seperti menambahkan menu pada reservasi atau mengatur tata letak meja yang dipesan sesuai keinginan.

```
1 const resizeCanvas = () => {
2   const canvas = editor.canvas;
3   const outerCanvasContainer = document.querySelector('.fabric-canvas-wrapper');
4   const containerWidth = outerCanvasContainer!.clientWidth;
5   const scale = containerWidth / canvas.getWidth();
6   const zoom = canvas.getZoom() * scale;
7   canvas.setDimensions({ width: containerWidth, height: 400 });
8   canvas.setViewportTransform([zoom, 0, 0, zoom, 0, 0]);
9 };
10
11 window.addEventListener('resize', resizeCanvas);
12 resizeCanvas();
13
14 if (floorplanFromDB) {
15   const floorplan = JSON.parse(floorplanFromDB);
16   const exampleJson = { objects: floorplan };
17   editor.canvas.loadFromJSON(exampleJson, editor.canvas.renderAll.bind(editor.canvas));
18 }
```

Fungsi 'resizeCanvas' bertanggung jawab untuk mengatur ulang ukuran dan zoom canvas sesuai dengan lebar container luar saat ukuran jendela berubah, dan event listener untuk resize ditambahkan ke window. Selain itu, jika ada data denah lantai yang disimpan di basis data 'floorplanFromDB', kode akan memuat dan merender denah tersebut ke canvas.



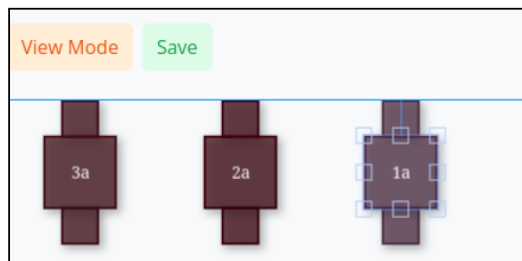
```

1 const toggleEditMode = () => {
2   setEditing(!isEditing);
3
4   if (!editor || !fabric) return;
5   const canvas = editor.canvas;
6
7   if (!isEditing) {
8     canvas.getObjects().map((group) => {
9       const listTable = table.map(item => item.label);
10      if (listTable.includes(group.id!)) {
11        group.selectable = true;
12        group.hasControls = true;
13        group.lockMovementX = false;
14        group.lockMovementY = false;
15        group.borderColor = 'rgba(102, 153, 255, 0.75)';
16        group.borderScaleFactor = 1;
17      }
18    })
19    canvas.selection = true
20    canvas.hoverCursor = 'move'
21    canvas.discardActiveObject()
22    canvas.renderAll()
23  } else {
24    canvas.getObjects().map(o => {
25      o.hasControls = false
26      o.lockMovementX = true
27      o.lockMovementY = true
28      if (!o.id!.includes('table')) {
29        o.selectable = false
30      }
31      o.borderColor = '#38A62E'
32      o.borderScaleFactor = 2.5
33    })
34    canvas.selection = false
35    canvas.hoverCursor = 'pointer'
36    canvas.discardActiveObject()
37    canvas.renderAll()
38  }
39 };

```

Fungsi 'toggleEditMode' dalam kode tersebut bertugas untuk mengatur mode pengeditan pada kanvas yang menggunakan fabric.js. Fungsi ini pertama-tama mengubah status isEditing menjadi kebalikan dari nilai saat ini, kemudian memeriksa keberadaan objek editor dan fabric. Jika salah satunya tidak ada, fungsi akan berhenti. Jika objek-objek tersebut ada, fungsi akan mengakses kanvas dari editor. Saat mode pengeditan diaktifkan

(jika `isEditing` awalnya `false`), fungsi mengizinkan objek-objek di kanvas yang memiliki ID dalam daftar tabel untuk dipilih, dikontrol, dan dipindahkan. Properti seleksi dan kursor pada kanvas juga diatur untuk memungkinkan seleksi dan menggunakan kursor "move".



Sebaliknya, saat mode pengeditan dinonaktifkan (jika `isEditing` awalnya `true`), fungsi mengunci kontrol dan pergerakan objek, serta menonaktifkan seleksi kecuali untuk objek yang merupakan tabel. Warna dan skala border juga disesuaikan kembali. Properti seleksi dan kursor pada kanvas diatur untuk menonaktifkan seleksi dan menggunakan kursor "pointer". Fungsi membatalkan seleksi objek aktif dan memperbarui tampilan kanvas.

```
1 const saveFloorplanHandler = () => {
2   setLoading(true);
3   const { objects } = editor!.canvas.toJSON([
4     'id', 'selectable', 'lockMovementX', 'lockMovementY', 'borderColor', 'hasControls'
5   ]);
6
7   const filteredObjects = objects.filter((item: any) => item.type !== 'line')
8   filteredObjects.map((item: any) => {
9     if (!item.id.includes('table')) {
10      item.selectable = false;
11    }
12    item.hasControls = false
13    item.lockMovementX = true
14    item.lockMovementY = true
15    item.borderColor = '#38A62E'
16    item.borderScaleFactor = 2.5
17  })
18
19  setBookingData((prev: any) => {
20    if (prev) {
21      return {
22        ...prev,
23        floorplan: JSON.stringify(filteredObjects),
24      };
25    }
26    return prev;
27  });
28  setEditing(false);
29  setLoading(false);
30 };
```

---

Kode di atas mendefinisikan fungsi 'saveFloorplanHandler' yang digunakan untuk menyimpan floor plan dalam suatu editor kanvas. Fungsi ini bekerja dengan mengambil data objek dari kanvas editor yang telah difilter berdasarkan properti tertentu seperti id, selectable, lockMovementX, lockMovementY, borderColor, dan hasControls. Setelah mendapatkan daftar objek dari kanvas, fungsi memfilter objek-objek tersebut untuk mengecualikan objek dengan tipe 'line'. Kemudian, fungsi melakukan iterasi pada setiap objek yang tersisa, dan melakukan pengecekan apakah id objek tersebut tidak mengandung 'table'. Jika benar, properti selectable dari objek tersebut diatur ke false. Selain itu, fungsi juga mengatur properti 'hasControls' ke false, 'lockMovementX' dan 'lockMovementY' ke true, serta 'borderColor' menjadi '#38A62E' dan 'borderScaleFactor' menjadi 2.5 untuk setiap objek.

Setelah memproses objek-objek tersebut, fungsi memperbarui state 'bookingData' dengan memasukkan floorplan yang telah difilter dan diubah ke dalam bentuk JSON. Terakhir, fungsi mengubah state editing menjadi false untuk menandakan bahwa proses penyuntingan telah selesai.



Pada langkah ketiga adalah halaman untuk memilih metode dan tipe pembayaran yang diinginkan. Tersedia berbagai metode pembayaran, seperti BCA, BNI, BRI, CIMB, pembayaran melalui konter seperti Indomaret dan Alfamart. Untuk tipe pembayaran, pilih antara pembayaran penuh (full payment) atau menggunakan uang muka (DP).

```

1  const paymentHandler = () => {
2    if (!bookingData) return;
3    setLoading(true);
4    const joinDateHour = dayjs(bookingData.bookingAt).set('hour', bookingData.duration[0])
5    const payload = {
6      bookingAt: joinDateHour.toDate(),
7      duration: bookingData.duration.length,
8      paymentType: [expanded, value],
9      dpOrCash: isPaid,
10     dpAmount: Number(dp),
11   };
12
13   axios.post('/api/v1/charge', { ...bookingData, ...payload })
14     .then(res => {
15       const response = res.data.data;
16
17       if (response.status_code === "201") {
18         if (response.payment_type === "bank_transfer") {
19           setBookingData((prev: any) => {
20             if (prev) {
21               return {
22                 ...prev,
23                 orderId: response.order_id,
24                 paymentMethod: [response.payment_type, response.va_numbers[0].bank,
25                   response.va_numbers[0].va_number],
26                 total: response.gross_amount,
27                 expired: response.expiry_time,
28               };
29             }
30             return prev;
31           });
32         }
33         if (response.payment_type === "cstore") {
34           setBookingData((prev: any) => {
35             if (prev) {
36               return {
37                 ...prev,
38                 orderId: response.order_id,
39                 paymentMethod: [response.payment_type, response.store,
40                   response.payment_code],
41                 total: response.gross_amount,
42                 expired: response.expiry_time,
43               };
44             }
45             return prev;
46           });
47         }
48         onHandleNext();
49       }
50     })
51     .catch(err => alert(err.response.data))
52     .finally(() => setLoading(false));
53   };

```

Fungsi 'paymentHandler' dalam aplikasi ini bertujuan untuk menangani proses pembayaran pemesanan. Fungsi dimulai dengan memeriksa apakah data pemesanan (bookingData) tersedia. Jika tidak, fungsi akan keluar. Setelah validasi, state loading diatur menjadi true untuk menunjukkan bahwa proses sedang berlangsung. Selanjutnya, fungsi menggunakan dayjs untuk mengatur jam pemesanan sesuai dengan data durasi pemesanan yang ada. Data yang sudah dimodifikasi ini kemudian dimasukkan ke dalam objek payload yang berisi detail pemesanan dan pembayaran.

**Convenience Store**

- An admin fee of Rp2.500-Rp7.500 will be charged later when you pay at Convenience Store. But keep in mind, the amount can change anytime without prior notice.

Indomaret

Alfamart

**Payment Type**

Full Payment  Down Payment

**Total Price** Rp 64.500,00

Loading

By continuing to pay, you have agreed to Risol's [Terms & Conditions](#) and [Privacy Policy](#).

Fungsi kemudian mengirimkan permintaan POST ke endpoint /api/v1/charge dengan menggunakan Axios, menyertakan data pemesanan dan payload tersebut. Jika permintaan berhasil dan server mengembalikan status kode "201", fungsi akan memperbarui data pemesanan dengan informasi pembayaran yang diterima dari server, seperti nomor pesanan (orderid), metode pembayaran, jumlah total pembayaran, dan waktu kedaluwarsa (expired). Pembaruan ini dilakukan sesuai dengan jenis pembayaran yang diterima, baik itu transfer bank atau toko konter. Setelah data diperbarui, fungsi memanggil onHandleNext() untuk melanjutkan ke langkah berikutnya yaitu tata cara pembayaran.

G558926747

**How to Pay at Indomaret**

1. Inform the Indomaret cashier that you want to make a payment for a Risol transaction.
2. Show your Payment Code (not Risol Booking ID) to the cashier.
3. An extra fee of Rp2.500/transaction, which is excluded from the total price, will be charged by Indomaret.

---

Price before admin fee: **Rp 64.500,00**

Jika permintaan gagal, fungsi akan menampilkan pesan kesalahan menggunakan alert. Terlepas dari hasil permintaan, state loading diatur kembali menjadi false setelah semua proses selesai. Dengan demikian, fungsi 'paymentHandler' ini mengelola seluruh proses pembayaran dari awal hingga akhir, memastikan bahwa semua data relevan diproses dan di update dengan benar.

```

1 const status = await axios.post('https://api.sandbox.midtrans.com/v2/charge', {
2   payment_type: body.paymentType[0],
3   [body.paymentType[0]]: body.paymentType[0] === 'bank_transfer'
4     ? { bank: body.paymentType[1] }
5     : { store: body.paymentType[1] },
6   transaction_details: {
7     order_id: Date.now().toString(),
8     gross_amount: body.dpOrCash === 'CASH' ? total + 4500 : body.dpAmount,
9   },
10  item_details: itemDetails,
11  customer_details: {
12    first_name: "",
13    last_name: body.onBehalf,
14    email: body.email,
15    phone: body.phone,
16  },
17  custom_expiry: {
18    expiry_duration: 90,
19    unit: "minute"
20  }
21 }, {
22   headers: {
23     'Accept': 'application/json',
24     'Content-Type': 'application/json',
25     'Authorization': basicAuth,
26   }
27 });

```

Kode di atas melakukan permintaan POST ke endpoint Midtrans untuk memproses pembayaran. Kode tersebut memanggil `axios.post` dengan URL `https://api.sandbox.midtrans.com/v2/charge`, mengirimkan beberapa detail pembayaran dan transaksi. Parameter `'payment_type'` diisi dengan tipe pembayaran yang dipilih oleh pengguna, misalnya `'bank_transfer'`. Berdasarkan tipe pembayaran ini, kode menentukan apakah pembayaran dilakukan melalui transfer bank atau toko tertentu dan menyertakan detail yang sesuai.

Bagian `'transaction_details'` mengandung `order_id` yang dihasilkan dari timestamp saat ini dan `gross_amount` yang merupakan jumlah total transaksi. Jumlah ini bisa berupa total keseluruhan ditambah biaya administrasi jika menggunakan metode pembayaran `'CASH'`, atau jumlah pembayaran di muka jika metode `'DP'`.

Kemudian, kode ini juga mengirimkan `item_details`, yang berisi rincian item yang dibeli, dan `'customer_details'`, yang menyertakan informasi pelanggan seperti nama belakang, email, dan nomor

telepon. Selain itu, 'custom\_expiry' disertakan untuk menentukan durasi kadaluarsa transaksi dalam 90 menit. Permintaan ini juga mencakup beberapa header, yaitu Accept, Content-Type, dan Authorization yang diisi dengan nilai 'basicAuth', memastikan permintaan diotorisasi dengan benar.

```
1 if (status.data.status_code === "201") {
2   if (status.data.payment_type === 'bank_transfer') {
3     const vaNumbers = status.data.va_numbers;
4
5     await prisma.reservation.create({
6       data: {
7         code: status.data.order_id,
8         total: total + 4500,
9         email: body.email,
10        phone: body.phone,
11        onBehalf: body.onBehalf,
12        paymentMethod: [body.paymentType[0], body.paymentType[1], vaNumbers[0].va_number],
13        paymentType: body.dpOrCash === 'CASH' ? 'CASH' : 'DOWNPAYMENT',
14        downpay: body.dpAmount,
15        duration: body.duration.toString(),
16        bookingAt: body.bookingAt,
17        notes: body.floorplan ? body.floorplan : null,
18        expiredAt: adjustedDate(new Date(status.data.expiry_time)),
19        bookingTable: {
20          connect: body.tables.map((table: Table) => ({ id: table.id })),
21        },
22        bookingBy: {
23          connect: { id: Number(session!.user.id) }
24        },
25        bookingPlace: {
26          connect: { id: body.restoId }
27        },
28        menus: {
29          create: body.menus.map((menu: any) => {
30            return {
31              quantity: menu.quantity,
32              menu: { connect: { id: menu.id } }
33            }
34          })
35        }
36      }
37    })
38  }
39 }
```

Kode di atas merupakan bagian dari logika pemrosesan setelah menerima respons dari API, khususnya untuk transaksi pembayaran. Jika status respons menunjukkan bahwa transaksi berhasil dengan status kode "201" dan jenis pembayaran adalah "bank\_transfer", maka dilakukan penyimpanan data reservasi ke dalam database menggunakan Prisma ORM.

Pertama, variabel vaNumbers diambil dari data respons untuk mendapatkan nomor rekening virtual (Virtual Account). Kemudian, fungsi prisma.reservation.create digunakan untuk membuat entri baru dalam tabel reservasi dengan data yang sesuai. Data ini



---

mencakup kode pesanan, total pembayaran (ditambah biaya tambahan sebesar 4500), email, nomor telepon, nama pemesan, metode pembayaran, tipe pembayaran (tunai atau uang muka), jumlah uang muka, durasi reservasi, waktu pemesanan, catatan tambahan, waktu kadaluwarsa (yang disesuaikan dengan fungsi `adjustedDate`), tabel yang dipesan (berdasarkan ID tabel yang dikonversi), identitas pengguna yang memesan (berdasarkan ID pengguna yang sedang aktif), tempat pemesanan (berdasarkan ID restoran), serta menu yang dipesan (dengan jumlah dan ID menu).

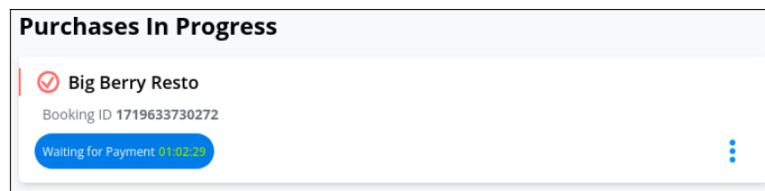
#### 4. Daftar Reservasi Saat Ini

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

```
1 const getPendingBooking = async (userId: number) => {
2   try {
3     const query = await prisma.reservation.findMany({
4       where: {
5         bookingBy: { id: userId },
6         paymentStatus: 'PENDING',
7       },
8       include: { bookingPlace: true },
9       orderBy: { id: 'desc' }
10    })
11
12    return query;
13  } catch (error: any) {
14    console.log('Error fetching', error);
15    return error.message
16  }
17 };
```

Fungsi `getPendingBooking` adalah fungsi asinkron yang dirancang untuk mengambil data pemesanan dengan status pembayaran `PENDING` dari database menggunakan Prisma ORM. Fungsi ini menerima satu parameter `userId`, yang merupakan ID pengguna. Di dalam fungsi, `prisma.reservation.findMany` digunakan untuk mencari semua pemesanan yang dilakukan oleh pengguna dengan `userId` tersebut dan memiliki status pembayaran `PENDING`.

Selain itu, fungsi ini juga menyertakan informasi tentang tempat pemesanan (`bookingPlace`) dan mengurutkan hasil berdasarkan ID secara menurun, sehingga pemesanan terbaru akan berada di awal hasil. Jika terjadi kesalahan selama proses pengambilan data, kesalahan tersebut akan ditangkap oleh blok `catch`, dicetak ke konsol untuk tujuan debugging, dan fungsi akan mengembalikan pesan kesalahan.



## 5. Riwayat Pesanan

Fitur ini menampilkan semua riwayat pesanan yang pernah dibuat oleh pengguna atau pelanggan.

```
1 const getBookingHistories = async (userId: number) => {
2   try {
3     const query = await prisma.reservation.findMany({
4       where: {
5         bookingBy: { id: userId },
6         deletedAt: null,
7       },
8       include: {
9         bookingPlace: true,
10        reviews: true
11      },
12      orderBy: { updatedAt: "desc" }
13    })
14
15    return query;
16  } catch (error: any) {
17    console.log('Error fetching', error);
18    return error.message
19  }
20 };
```

Fungsi `getBookingHistories` adalah fungsi asinkron yang mengambil riwayat pemesanan dari basis data untuk pengguna tertentu, diidentifikasi dengan `userId`, menggunakan Prisma ORM. Fungsi ini mencoba mengeksekusi query pada tabel `reservation` untuk menemukan semua entri yang dibuat oleh pengguna dengan id yang sesuai, yang belum dihapus (`deletedAt null`). Hasil query juga menyertakan data terkait dari tabel `bookingPlace` dan `reviews`, dan diurutkan berdasarkan waktu pembaruan dalam urutan menurun. Jika terjadi kesalahan selama pengambilan data, fungsi ini menangkap kesalahan tersebut, mencetak pesan kesalahan ke konsol, dan mengembalikan pesan kesalahan tersebut sebagai hasil fungsi.

Purchases In Progress	
Booking ID 1719633730272	Rp 64.500,00
Big Berry Resto	
Waiting for payment	<a href="#">See Details</a>
Booking ID 1719147964409	Rp 34.500,00
Big Berry Resto	
Time Limit Exceeded	<a href="#">See Details</a> <a href="#">...</a>
Booking ID 1719146563201	Rp 64.500,00
Big Berry Resto	
Cancelled	<a href="#">See Details</a> <a href="#">...</a>
Booking ID 1719145210974	Rp 34.500,00
Big Berry Resto	
Payment successful	<a href="#">See Details</a> <a href="#">...</a>

## 6. Mendaftar Sebagai Mitra

Fitur ini memungkinkan pengguna website yang memiliki restoran untuk mendaftarkan restoran mereka menjadi mitra Risol.

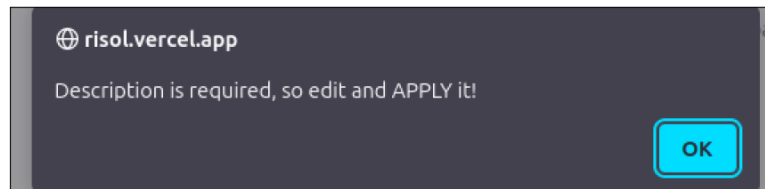
```

1 const handleSubmit = (e: FormEvent<HTMLFormElement>) => {
2   e.preventDefault()
3   setLoading(true);
4   const formData = new FormData();
5
6   if (!datas.desc) {
7     setLoading(false);
8     return alert('Description is required, so edit and APPLY it!');
9   }
10  if (datas.images.length === 0) {
11    setLoading(false);
12    return alert('Images is required');
13  }
14  if (datas.docs.length < 3) {
15    setLoading(false);
16    return alert('All document is rquired');
17  }
18
19  formData.append('name', datas.name);
20  formData.append('desc', datas.desc);
21  formData.append('open', datas.open!.split(':')[0]);
22  formData.append('close', datas.close!.split(':')[0]);
23  formData.append('phone', datas.phone);
24  formData.append('lat', datas.lat);
25  formData.append('long', datas.long);
26  formData.append('address', datas.address);
27  formData.append('province', datas.province!);
28  formData.append('district', datas.district!);
29  formData.append('subdistrict', datas.subdistrict!);
30  datas.docs.forEach(doc => formData.append('doc', doc as Blob))
31  datas.images.forEach(image => formData.append('image', image as Blob))
32
33  axios.post('/api/v1/partner', formData)
34    .then(res => {
35      setSuccess(true);
36      console.log(res.data)
37    })
38    .catch(err => {
39      setSuccess(false);
40      console.log(err.response.data)
41    })
42    .finally(() => {
43      setOpen(true);
44      setLoading(false)
45    })
46 };

```

Fungsi handleSubmit dibuat untuk menangani pengiriman data pada API. Ketika formulir dikirimkan, fungsi ini mencegah perilaku pengiriman default dan mengatur status pemuatan (loading). Fungsi kemudian melakukan beberapa pengecekan validasi pada

objek data: memastikan deskripsi diisi, memastikan ada gambar, dan memastikan ada setidaknya tiga dokumen. Jika salah satu validasi gagal, status pemuatan dihentikan dan pengguna akan menerima peringatan.



Jika semua validasi berhasil, fungsi ini membuat objek FormData dan menambahkan berbagai bidang dari objek data, termasuk nama, deskripsi, jam buka dan tutup, nomor telepon, lintang (latitude), bujur (longitude), alamat, dan detail lokasi (provinsi, distrik, kelurahan). Fungsi ini juga menambahkan setiap dokumen dan file gambar. Terakhir, fungsi ini mengirim FormData tersebut ke endpoint `/api/v1/partner` menggunakan permintaan POST dari axios.

```
1 const images = body.getAll('image') as File[];
2 const imagesPayload = await Promise.all(images.map(async (image, i) => {
3   const res = await uploadToCloudinary(image, { folder: "restos", format: 'webp' });
4   if (res) {
5     return { url: res.secure_url };
6   } else {
7     throw new Error('Fail upload image ${i} to cloudinary');
8   }
9 }));
10
11 const docs = body.getAll('doc') as File[];
12 const docsPayload = await Promise.all(docs.map(async (doc, i) => {
13   const res = await uploadToCloudinary(doc, { folder: "docs" });
14   if (res) {
15     return res.secure_url;
16   } else {
17     throw new Error('Fail upload image ${i} to cloudinary');
18   }
19 }));
```

Kode diatas adalah penanganan dan pemrosesan data formulir untuk pendaftaran restoran sebagai mitra. Pertama, kode mengambil file gambar dari data formulir menggunakan `body.getAll("image")`, dan mengubahnya menjadi array objek File. Kemudian, kode mengunggah gambar-gambar tersebut ke Cloudinary menggunakan fungsi `uploadToCloudinary`, dengan

---

menetapkan folder ke "restos" dan format ke 'webp'. Setiap unggahan yang berhasil mengembalikan URL gambar yang diunggah, yang dikumpulkan ke dalam array `imagesPayload`. Jika ada unggahan yang gagal, sebuah error akan muncul yang menunjukkan gambar mana yang gagal diunggah.

Dengan cara yang sama, kode mengambil file dokumen dari data formulir menggunakan `body.getAll('doc')`, dan mengubahnya menjadi array objek `File`. Kemudian kode mengunggah dokumen-dokumen tersebut ke Cloudinary sebagai sumber daya mentah, dengan menetapkan folder ke "docs". URL dokumen yang berhasil diunggah dikumpulkan ke dalam array `docsPayload`, dengan error yang muncul untuk setiap unggahan yang gagal.

```
1 const resto = await prisma.resto.create({
2   data: {
3     name: body.get('name')!.toString(),
4     phone: body.get('phone')!.toString(),
5     desc: body.get('desc')!.toString(),
6     address: body.get('address')!.toString(),
7     doc: docsPayload,
8     open: body.get('open')!.toString(),
9     close: body.get('close')!.toString(),
10    province: body.get('province')!.toString(),
11    district: body.get('district')!.toString(),
12    subdistrict: body.get('subdistrict')!.toString(),
13    lat: String(body.get('lat')),
14    long: String(body.get('long')),
15    ownerId: Number(session.user.id),
16  },
17  include: { owner: true }
18 })
```

Setelah mengunggah semua gambar dan dokumen, kode membuat entri restoran baru di database menggunakan Prisma ORM. Kode mengumpulkan berbagai data formulir seperti nama restoran, nomor telepon, deskripsi, alamat, jam operasional, lokasi, dan informasi pemilik. URL dokumen yang diunggah dimasukkan ke dalam field `doc`. Entri restoran baru dibuat dengan detail ini dan informasi pemilik juga termasuk dalam respons.

## 7. Membuat Meja Baru

Fitur ini memungkinkan mitra restoran untuk menambah meja baru ke dalam sistem.

```
1 onSubmit: (event: React.FormEvent<HTMLFormElement>) => {
2   event.preventDefault();
3   const formData = new FormData(event.currentTarget);
4   const formJson = Object.fromEntries((formData as any).entries());
5
6   axios.post('/api/v1/table', { body: formJson })
7   .then(res => {
8     setResponse((prev) => ({
9       ...prev,
10      open: true,
11      type: "success",
12      message: res.data.message
13    })))
14  })
15  .catch(err => alert(err.response.data.message))
16  .finally(() => {
17    router.refresh();
18    handleClose();
19  })
20 },
```

Fungsi `onSubmit` menangani pengiriman formulir ke API untuk membuat meja baru. Saat formulir dikirim, fungsi ini mencegah perilaku default pengiriman formulir dengan memanggil `event.preventDefault()`. Selanjutnya, data formulir dikumpulkan menggunakan objek `FormData` dari elemen formulir yang dipicu (`event.currentTarget`). Data formulir ini kemudian diubah menjadi objek JSON dengan `Object.fromEntries((formData as any).entries())`. Objek JSON ini kemudian dikirim ke endpoint `/api/v1/table` menggunakan metode POST dari `axios`. Jika permintaan berhasil, fungsi `then` memutakhirkan state response dengan membuka notifikasi sukses yang menampilkan pesan dari respons server. Jika permintaan gagal, fungsi `catch` akan menampilkan pesan kesalahan dari respons server menggunakan `alert`. Terakhir, di dalam blok `finally`, fungsi `router.refresh()` dipanggil untuk menyegarkan halaman, dan `handleClose()` dipanggil untuk menutup dialog atau komponen terkait.

```

1 export const POST = async (request: NextRequest) => {
2   const { body } = await request.json();
3   const session = await getServerSession(authOptions);
4   if (!session) return responseBody('Unauthorized', 401);
5
6   try {
7     const existingTable = await prisma.table.findFirst({
8       where: {
9         label: body.label,
10        resto: { ownerId: Number(session.user.id) }
11      }
12    })
13    if (existingTable) throw new Error('Label already exists');
14
15    const tableQuery = await prisma.table.create({
16      data: {
17        label: body.label,
18        capacity: Number(body.type),
19        desc: body.desc || null,
20        resto: { connect: {
21          ownerId: Number(session.user.id)
22        } }
23      }
24    })
25    return responseBody('Successfully creating new table 🎉', 201, tableQuery)
26  } catch (error: any) {
27    let message: string = 'Oops 😞';
28    if (error instanceof Prisma.PrismaClientKnownRequestError) {
29      if (error.code === 'P2025') {
30        message += error.message;
31      }
32    } else {
33      message = error.message;
34    }
35    return responseBody(message, 400);
36  }
37 };

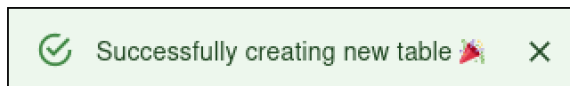
```

Saat menerima permintaan POST, fungsi ini pertama-tama memverifikasi otentikasi pengguna dengan memperoleh sesi dari server menggunakan `getServerSession(authOptions)`. Jika sesi tidak ditemukan, fungsi mengirimkan respons dengan status 401 Unauthorized. Setelah verifikasi sesi, fungsi melakukan pencarian entitas tabel berdasarkan label yang diberikan dan id pemilik restoran (`ownerId`). Jika tabel dengan label yang sama sudah ada, fungsi akan mengembalikan pesan error yang menyatakan 'Label already exists', menunjukkan bahwa label tersebut sudah digunakan sebelumnya. Jika tidak ada konflik dengan label yang ada, fungsi menggunakan Prisma untuk membuat entitas tabel baru dengan data yang diterima dari permintaan POST. Data termasuk label tabel, kapasitas (diambil dari tipe yang disediakan), deskripsi opsional, dan koneksi ke restoran yang dimiliki oleh pengguna sesuai dengan `ownerId`.



---

Hasil dari operasi basis data tersebut kemudian dikembalikan dalam respons dengan status 201 Created, bersama dengan pesan sukses dan data tabel yang baru dibuat. Namun, jika terjadi kesalahan saat berkomunikasi dengan basis data (misalnya kesalahan Prisma), fungsi akan menangkap kesalahan tersebut dan mengirimkan respons dengan status 400 Bad Request, memberikan informasi yang relevan mengenai kesalahan yang terjadi.



## 8. Mengatur Posisi Meja

Fitur ini memungkinkan mitra restoran untuk mengatur dan mengubah posisi meja yang ada.

Kode di bawah mendefinisikan sebuah fungsi bernama `onAddCashier` yang bertujuan untuk menambahkan elemen kasir ke dalam canvas menggunakan library `fabric.js`. Pertama, kode ini membuat objek persegi panjang (`Rect`) dengan properti tertentu seperti lebar, tinggi, warna isi, warna garis tepi, ketebalan garis, bayangan, serta pengaturan pusat rotasi dan sudut snap. Setelah itu, teks (`IText`) dengan properti font, ukuran font, warna, dan perataan teks juga dibuat. Kedua objek tersebut kemudian dikelompokkan (`Group`) menjadi satu entitas yang memiliki properti posisi, rotasi terpusat, sudut snap, keterpilihan, dan identifikasi unik yang dihasilkan oleh fungsi `generateId` dari objek `floorplan`. Elemen grup ini kemudian ditambahkan ke dalam canvas editor untuk ditampilkan. Secara keseluruhan, fungsi ini memfasilitasi penambahan visualisasi elemen kasir yang dapat dipindah-pindahkan dan dirotasi pada `floorplan` editor.

```

1 const onAddCashier = () => {
2   const o = new fabric.Rect({
3     width: 90,
4     height: 60,
5     fill: floorplan.cashierFill,
6     stroke: floorplan.cashierStroke,
7     strokeWidth: 2,
8     shadow: floorplan.cashierShadow,
9     originX: 'center',
10    originY: 'center',
11    centeredRotation: true,
12    snapAngle: 45,
13    selectable: true
14  })
15  const t = new fabric.IText(floorplan.cashierText, {
16    fontFamily: 'Calibri',
17    fontSize: 14,
18    fill: '#fff',
19    textAlign: 'center',
20    originX: 'center',
21    originY: 'center'
22  })
23  const g = new fabric.Group([o, t], {
24    left: 0,
25    top: 0,
26    centeredRotation: true,
27    snapAngle: 45,
28    selectable: true,
29    id: `cashier-${floorplan.generateId()}`,
30    number: number
31  })
32
33  editor!.canvas.add(g);
34 };

```

Fungsi `onAddChair` digunakan untuk menambahkan objek kursi persegi ke dalam kanvas editor dengan pustaka `Fabric.js`. Fungsi ini menerima dua parameter: `chairId` yang merupakan identitas kursi dalam bentuk string, dan `color`, sebuah objek yang berisi properti `rgba` dan `rgb` yang menyimpan nilai warna dalam format `RGBA` dan

RGB. Di dalam fungsi ini, sebuah objek persegi (`fabric.Rect`) dibuat dengan berbagai properti yang telah ditentukan seperti posisi (kiri dan atas diatur ke 0), ukuran (lebar dan tinggi masing-masing 30), warna isian (dari `color.rgba`), warna garis tepi (dari `color.rgb`), ketebalan garis tepi (2), dan bayangan (dari `floorplan.chairShadow`). Selain itu, properti lainnya seperti titik referensi horisontal dan vertikal, rotasi terpusat, sudut rotasi yang terkunci setiap 45 derajat, dan kemampuan untuk dipilih juga diatur. Setelah semua properti ini diatur, objek persegi tersebut ditambahkan ke dalam kanvas editor menggunakan `editor!.canvas.add(o)`. Fungsi ini memungkinkan pengguna untuk melihat dan berinteraksi dengan kursi tersebut dalam editor, sesuai dengan identitas dan warna yang diberikan.

```
1 function onAddChair(chairId: string, color: { rgba: string, rgb: string }) {
2   const o = new fabric.Rect({
3     left: 0,
4     top: 0,
5     width: 30,
6     height: 30,
7     fill: color.rgba,
8     stroke: color.rgb,
9     strokeWidth: 2,
10    shadow: floorplan.chairShadow,
11    originX: 'left',
12    originY: 'top',
13    centeredRotation: true,
14    snapAngle: 45,
15    selectable: true,
16    id: chairId
17  })
18  editor!.canvas.add(o)
19 }
```

Fungsi `createShape` digunakan untuk membuat bentuk lingkaran atau persegi dengan label yang diberikan dan menambahkannya ke dalam canvas menggunakan pustaka `fabric.js`. Fungsi ini pertama-tama menghasilkan warna acak, lalu memeriksa tipe bentuk yang diinginkan (`circle` atau `square`). Berdasarkan tipe tersebut, ia membuat objek `fabric.Circle` atau `fabric.Rect` dengan atribut seperti warna, bayangan, dan pusat rotasi yang telah ditentukan. Selanjutnya, fungsi ini membuat objek teks dengan label yang diberikan dan menggabungkannya dengan bentuk dalam satu grup `fabric.Group`. Fungsi ini juga

menambahkan jumlah kursi yang sesuai dengan kapasitas yang ditemukan dalam dataTable untuk label yang diberikan. Setelah itu, fungsi memperbarui nomor identifikasi dan daftar label, serta menambahkan grup yang telah dibuat ke dalam canvas editor.

```
1 const createShape = ( label: string, updatedLabel: string[], shapeType: 'circle' | 'square' ) => {
2   const color = floorplan.randomColor();
3   let shape: fabric.Circle | fabric.Rect | undefined;
4
5   if (shapeType === 'circle') {
6     shape = new fabric.Circle({
7       radius: 30,
8       fill: color.rgba,
9       stroke: color.rgb,
10      strokeWidth: 2,
11      shadow: floorplan.tableShadow,
12      originX: 'center',
13      originY: 'center',
14      centeredRotation: true
15    });
16   } else if (shapeType === 'square') {
17     shape = new fabric.Rect({
18       width: 60,
19       height: 60,
20       fill: color.rgba,
21       stroke: color.rgb,
22       strokeWidth: 2,
23       shadow: floorplan.tableShadow,
24       originX: 'center',
25       originY: 'center',
26       centeredRotation: true,
27       snapAngle: 45,
28       selectable: true
29     });
30   }
31
32   const text = new fabric.IText(label, {
33     fontFamily: 'Calibri',
34     fontSize: 14,
35     fill: '#fff',
36     textAlign: 'center',
37     originX: 'center',
38     originY: 'center'
39   });
40
41   const group = new fabric.Group([shape!, text], {
42     left: 0,
43     top: 0,
44     centeredRotation: true,
45     snapAngle: 45,
46     selectable: true,
47     id: `${label}`,
48     number: number
49   });
50
51   const chairAmount = dataTable.find((item) => item.label === label)!.capacity;
52   for (let i = 0; i < chairAmount; i++) {
53     onAddChair(`${label}`, color);
54   }
55
56   setNumber(prev => prev + 1);
57   setListLabel(updatedLabel);
58   editor!.canvas.add(group);
59 };
```

```
1 const saveFloorplanHandler = () => {
2   setLoading(true);
3   const { objects } = editor!.canvas.toJSON([
4     'id', 'selectable', 'lockMovementX', 'lockMovementY', 'borderColor', 'hasControls'
5   ]);
6
7   const filteredObjects = objects.filter((item: any) => item.type !== 'line')
8   filteredObjects.map((item: any) => {
9     if (!item.id.includes('table')) {
10      item.selectable = false;
11    }
12    item.hasControls = false
13    item.lockMovementX = true
14    item.lockMovementY = true
15    item.borderColor = '#38A62E'
16    item.borderScaleFactor = 2.5
17  })
18
19  axios.post('/api/v1/floorplan', { data: JSON.stringify(filteredObjects) })
20    .then((res) => console.log(res.data))
21    .catch((err) => console.log(err.response.data.message || err.message))
22    .finally(() => {
23      setLoading(false);
24      router.refresh();
25    })
26  };
```

Fungsi `saveFloorplanHandler` digunakan untuk menyimpan denah lantai dari editor ke server dengan melalui beberapa langkah. Pertama, fungsi ini mengubah status loading menjadi true untuk menandakan proses penyimpanan sedang berlangsung. Selanjutnya, data objek diambil dari kanvas editor dengan hanya menyertakan properti tertentu. Objek-objek yang bukan tipe 'line' kemudian disaring. Objek yang telah difilter dimodifikasi dengan mengatur properti `selectable` menjadi false jika objek bukan meja, menonaktifkan kontrol dengan mengatur `hasControls` menjadi false, mengunci pergerakan pada sumbu X dan Y, mengubah warna border menjadi hijau, dan mengatur skala border menjadi 2.5. Data objek yang telah dimodifikasi ini kemudian dikirim ke server menggunakan `axios.post` ke endpoint `/api/v1/floorplan`. Jika permintaan berhasil, respons data dicatat di konsol, sedangkan pesan kesalahan dicatat jika terjadi error. Terakhir, status loading dikembalikan menjadi false dan halaman disegarkan untuk memperbarui tampilan. Tujuan utama fungsi ini adalah memastikan bahwa denah lantai yang disimpan hanya berisi objek-objek tertentu dengan pengaturan yang tepat sebelum dikirim ke server.

## 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 : EC00202460236, 4 Juli 2024

**Pencipta**

Nama : **Yusron Arly Bazarah, Muhammad Fikri Hidayattullah dkk**  
Alamat : **Jl. Kuwusangka RT 19/03-Kelurahan Pagedangan, Kecamatan Adiwerna, Kabupaten Tegal, Propinsi Jawa Tengah 52194, Adiwerna, Tegal, Jawa Tengah, 52194**  
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 : **Risol: Reservasi Restoran Online**  
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : **4 Juli 2024, di Tegal**  
Jangka waktu perlindungan : **Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.**  
Nomor pencatatan : **000635593**

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	Yusron Arly Bazarah	Jl. Kuwusangka RT 19/03 Kelurahan Pagedangan, Kecamatan Adiwerna, Kabupaten Tegal, Propinsi Jawa Tengah 52194, Adiwerna, Tegal
2	Muhammad Fikri Hidayattullah	Jl. Glatik No.68, Randugunting, Kecamatan Tegal Selatan, Kota Tegal, Propinsi Jawa Tengah 52131, Tegal Selatan, Tegal
3	Dairoh	Jalan Garuda No 3 Rt 02 Rw 09 Randugunting, Kecamatan Tegal Selatan, Kota Tegal, Propinsi Jawa Tengah 52131, Tegal Selatan, Tegal



Lampiran 7. Lembar Bimbingan



**SARJANA TERAPAN TEKNIK INFORMATIKA  
POLITEKNIK HARAPAN BERSAMA**

**LEMBAR BIMBINGAN SKRIPSI**

Nama : Yusron Arly Bazarah  
 Nim : 20090034  
 No. Ponsel : 089675270897  
 Judul TA : Aplikasi Reservasi Meja dan Makanan Restoran di Kabupaten Tegal Berbasis Web  
 Dosen Pembimbing I : Muhammad Fikri Hidayattullah, S.T., M.Kom.

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
1.	21 Maret 2024	Bisnis proses	<ul style="list-style-type: none"> <li>- Tentukan skema monetisasi yg memungkinkan!</li> <li>- Kembangkan fitur drag n drop desain layout!</li> </ul>	
2.	25 April	Prototipe	<ul style="list-style-type: none"> <li>- Perbaiki bug!</li> </ul>	
3.	26 April	prototipe	<ul style="list-style-type: none"> <li>- Pengaturan tempat duduk furnitur ada di level Customer</li> </ul>	
4.	12 Juni	prototipe	<ul style="list-style-type: none"> <li>- Verifikasi kehalusan Customer</li> <li>- Silakan cari celah yg memungkinkan dikembangkan</li> </ul>	





**SARJANA TERAPAN TEKNIK INFORMATIKA  
POLITEKNIK HARAPAN BERSAMA**

5.	20 Juni	Manual Book	- Setiap menu dan fitur di Resol di kirim ke dosen ke Capture.skr	
6.	28 Juni	Manual Book & Technical Doc.	Ajutan ke P3M	
7.	9 Juli	Laporan	Kuarngkan argumen di setiap pembahsan!	
8.	12 Juli	Laporan	Acc. Siap Sidang!	

Tegal,  
Dosen Pembimbing I

Muhammad Fikri Hidayattullah, S.T., M.Kom.  
NIPY. 09.016.307



SARJANA TERAPAN TEKNIK INFORMATIKA  
POLITEKNIK HARAPAN BERSAMA






LEMBAR BIMBINGAN SKRIPSI

Nama : Yusron Arly Bazarah  
Nim : 20090034  
No. Ponsel : 089675270897  
Judul TA : Aplikasi Reservasi Meja dan Makanan Restoran di Kabupaten Tegal Berbasis Web  
Dosen Pembimbing I : Dairoh, M.Sc

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
1.	17/4 2024	diagram	<ul style="list-style-type: none"><li>- membuat uraian diagram</li><li>- membuat sequence diagram</li></ul>	
2.	6/5/2024	Aplikasi	<ul style="list-style-type: none"><li>- use case a c c</li><li>- proses bisnis nya</li><li>- layar</li><li>- aktor dan fungsi</li></ul>	
3.	2/6/2024	Aplikasi	<ul style="list-style-type: none"><li>- Aplikasi e/ bulat pemesanan</li><li>- Kehabisan.</li></ul>	



**SARJANA TERAPAN TEKNIK INFORMATIKA  
POLITEKNIK HARAPAN BERSAMA**

<b>No</b>	<b>Tanggal</b>	<b>Pemeriksaan</b>	<b>Perbaikan yang perlu diperbaiki</b>	<b>Paraf Pembimbing</b>
4	17 Juni 2024	Aplikasi	Acc aplikasi	
5	29 Juni 2024	HKI	Perbaikan manual book dan dokumen teknis	
6	30 Juni 2024	HKI	Acc manual book dan dokumen teknis	
7	12 Juli 2024	Laporan	Cek bab 1-3 laporan	
8	13 Juli 2024	Laporan	Acc laporan	

Tegal, 13 Juli 2024  
Dosen Pembimbing II



Dairoh, M.Sc.  
NIPY. 04.014.178