

Nama : Kamila Fajar Pertiwi

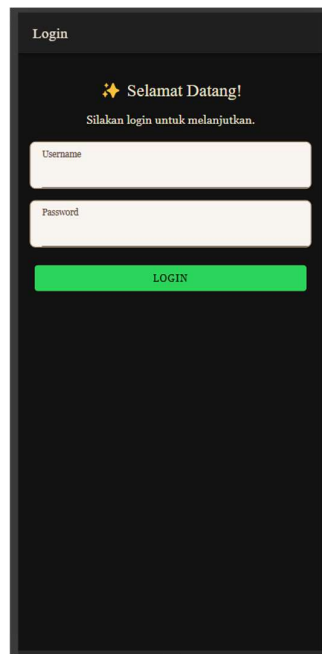
NIM : H1D022035

Shift KRS : A

Shift Baru : B

Responsi Prak PEMOB

1. Login



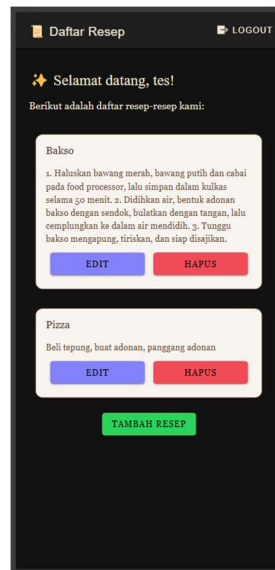
The screenshot shows a mobile application interface for a login page. At the top, it says "Login". Below that is a welcome message "Selamat Datang!" followed by "Silakan login untuk melanjutkan." There are two input fields: "Username" and "Password". Below these fields is a green button labeled "LOGIN".

Sintaks yang bertanggung jawab pada tampilan login berada pada file login.page.html

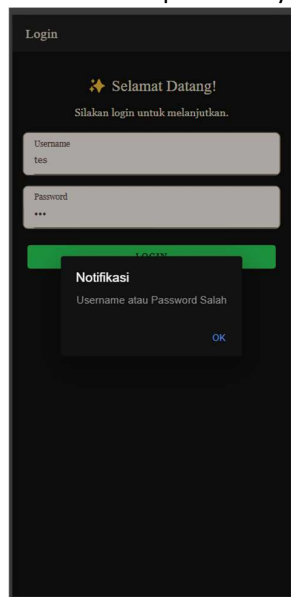
Untuk logikanya, di handle dengan:

```
login() {  
  if (this.username != null && this.password != null) {  
    const data = {  
      username: this.username,  
      password: this.password  
    }  
    this.authService.postMethod(data, 'login.php').subscribe({  
      next: (res) => {  
        if (res.status_login == "berhasil") {  
          this.authService.saveData(res.token, res.username);  
          this.username = '';  
          this.password = '';  
          this.router.navigateByUrl('/home');  
        } else {  
          this.authService.notifikasi('Username atau Password Salah');  
        }  
      },  
      error: (e) => {  
        this.authService.notifikasi('Login Gagal Periksa Koneksi Internet Anda');  
      }  
    })  
  } else {  
    this.authService.notifikasi('Username atau Password Tidak Boleh Kosong');  
  }  
}
```

Percabangan pertama jika pengguna input password serta username yang benar yang sudah terdaftar, maka akan masuk ke halaman home

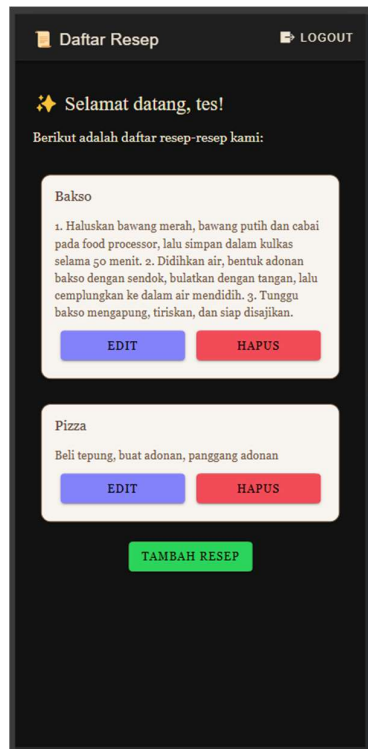


Percabangan else pertama, jika username atau password yang dimasukkan salah



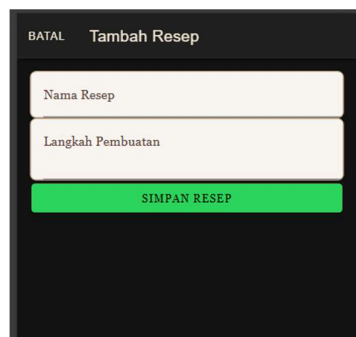
Percabangan else kedua jika gagal login karna koneksinya, dan percabangan terakhir jika username/password tidak dilengkapi

2. CRUD Resep



Sintaks yang bertanggung jawab pada tampilan Home Page berada pada file home.page.html

Create



Sintaks yang bertanggung jawab pada create, yaitu

```
<ion-modal [isOpen]="modalTambah" (ionModalDidDismiss)="cancel()">
  <ng-template>
    <ion-header>
      <ion-toolbar style="background-color: #5A4A83; color: #FFEE60;">
        <ion-title>Tambah Resep</ion-title>
        <ion-buttons slot="start">
          <ion-button (click)="cancel()" style="color: #FFEE60;">BATAL</ion-button>
        </ion-buttons>
      </ion-toolbar>
    </ion-header>
    <ion-content class="ion-padding" style="background-color: #F55C1E;">
      <ion-item style="--background: #F8F4E1; border: 1px solid #808382; border-radius: 8px;">
        <ion-input
          label="Nama Resep"
          labelPlacement="floating"
          required
          [(ngModel)]="nama_resep"
          placeholder="Masukkan Nama Resep"
          style="font-family: 'Georgia', serif; color: #5A4A83;"
        </ion-input>
      </ion-item>
      <ion-item style="--background: #F8F4E1; border: 1px solid #808382; border-radius: 8px;">
        <ion-textarea
          label="Langkah Pembuatan"
          labelPlacement="floating"
          required
          [(ngModel)]="langkah"
          placeholder="Masukkan Langkah Pembuatan"
          autoGrow="true"
          style="font-family: 'Georgia', serif; color: #5A4A83;"
        </ion-textarea>
      </ion-item>
      <ion-button expand="block" color="success" (click)="tambahResep()"
        style="background-color: #607B5E; color: #F8F4E1; border-radius: 6px; font-family: 'Georgia', serif;"
      >Simpan Resep
    </ion-content>
  </ng-template>
</ion-modal>
```

Sedangkan logikanya dihandle di home.pahe.ts, pada bagian

```

tambahResep() {
  if (this.nama_makanan != '' && this.langkah != '') {
    let data = { nama_makanan: this.nama_makanan, langkah: this.langkah };
    this.api.tambah(data, 'tambah.php').subscribe({
      next: (hasil: any) => {
        this.resetModal();
        console.log('berhasil tambah Resep');
        this.getResep();
        this.modalTambah = false;
        this.modal.dismiss();
      },
      error: (err: any) => {
        console.log('gagal tambah resep');
      }
    });
  } else {
    console.log('gagal tambah resep karena masih ada data yg kosong');
  }
}

```

Serta logika untuk membuka modal create, yaitu

```

openModalTambah(isOpen: boolean) {
  this.modalTambah = isOpen;
  this.resetModal();
  this.modalTambah = true;
  this.modalEdit = false;
}

```

Read/Tampil



Sintaks yang bertanggung jawab untuk memunculkan data setelah ditambah data baru, yaitu

```

<!-- Menampilkan List Resep -->
<ion-row *ngFor="let item of dataResep">
  <ion-col size="12">
    <ion-card style="background-color: #F8F4EF; border: 1px solid #6F4E37; border-radius: 12px;">
      <ion-card-header>
        <ion-card-title style="font-family: 'Georgia', serif; color: #5A4033; font-size: 1.2em;">
          {{ item.nama_makanan }}
        <ion-badge color="warning" slot="end" style="font-family: 'Times New Roman', serif;">
          Resep
        </ion-badge>
      </ion-card-title>
    </ion-card-header>
    <ion-card-content>
      <p style="font-family: 'Georgia', serif; color: #6F4E37;">{{ item.langkah }}</p>
      <ion-row>
        <ion-col size="6">
          <ion-button expand="block" color="tertiary" (click)="openModalEdit(true, item.id)"
            style="background-color: #D9B3B2; color: #402A20; border-radius: 6px; font-family: 'Georgia', serif;">
            Edit
          </ion-button>
        </ion-col>
        <ion-col size="6">
          <ion-button expand="block" color="danger" (click)="confirmDelete(item.id)"
            style="background-color: #B3B3B3; color: #F8F4EF; border-radius: 6px; font-family: 'Georgia', serif;">
            Hapus
          </ion-button>
        </ion-col>
      </ion-row>
    </ion-card-content>
  </ion-card>
</ion-col>
</ion-row>

```

Sedangkan logikanya di handle di home.page.ts

```

ngOnInit() {
  this.getResep();
}

getResep() {
  this.api.tampil('tampil.php').subscribe({
    next: (res: any) => {
      console.log('sukses', res);
      this.dataResep = res;
    },
    error: (err: any) => {
      console.log(err);
    },
  });
}

```

Update

BATAL
Edit Resep

Nama Resep

Bakso

Langkah Pembuatan

1. Haluskan bawang merah, bawang putih dan cabai pada food processor, lalu simpan dalam kulkas selama 50 menit.
2. Didihkan air, bentuk adonan bakso dengan sendok, bulatkan dengan tangan, lalu cemplungkan ke dalam air mendidih.
3. Tunggu bakso mengapung, tiriskan, dan siap disajikan.

SIMPAN PERUBAHAN

Sintaks yang bertanggung jawab pada tampilan update yaitu

```

<!-- Modal for Editing a Recipe -->
<ion-modal [isOpen]="modalEdit" (ionModalDidDismiss)="cancel()">
  <ng-template>
    <ion-header>
      <ion-toolbar style="background-color: #5A4033; color: #F0E602;">
        <ion-title>Edit Resep</ion-title>
        <ion-buttons slot="start">
          <ion-button (click)="cancel()" style="color: #F0E602;">Batal</ion-button>
        </ion-buttons>
      </ion-toolbar>
    </ion-header>
    <ion-content class="ion-padding" style="background-color: #F5ECE3;">
      <ion-item style="--background: #F8F4EF; border: 1px solid #D9B382; border-radius: 8px;">
        <ion-input>
          label="Nama Resep"
          labelPlacement="floating"
          required
          [(ngModel)]="nama_makanan"
          placeholder="Masukkan Nama Resep"
          style="font-family: 'Georgia', serif; color: #5A4033;"
        </ion-input>
      </ion-item>
      <ion-item style="--background: #F8F4EF; border: 1px solid #D9B382; border-radius: 8px;">
        <ion-textarea>
          label="Langkah Pembuatan"
          labelPlacement="floating"
          required
          [(ngModel)]="langkah"
          placeholder="Masukkan Langkah Pembuatan"
          autoGrow="true"
          style="font-family: 'Georgia', serif; color: #5A4033;"
        </ion-textarea>
      </ion-item>
      <ion-button expand="block" color="tertiary" (click)="editResep()"
        style="background-color: #D9B382; color: #402A20; border-radius: 6px; font-family: 'Georgia', serif;">
        Simpan Perubahan
      </ion-button>
    </ion-content>
  </ng-template>
</ion-modal>
</ion-content>

```

Sedangkan logikanya di handle di home.page.ts

```

ambilResep(id: any) {
  this.api.lihat(id, 'lihat.php?id=').subscribe({
    next: (hasil: any) => {
      console.log('sukses', hasil);
      let makanan = hasil;
      this.id = makanan.id;
      this.nama_makanan = makanan.nama_makanan;
      this.langkah = makanan.langkah;
    },
    error: (error: any) => {
      console.log('gagal ambil data');
    }
  });
}

modalEdit: any;

openModalEdit(isOpen: boolean, idget: any) {
  this.modalEdit = isOpen;
  this.id = idget;
  console.log(this.id);
  this.ambilResep(this.id);
  this.modalTambah = false;
  this.modalEdit = true;
}

editResep() {
  let data = { id: this.id, nama_makanan: this.nama_makanan, langkah: this.langkah };
  this.api.edit(data, 'edit.php').subscribe({
    next: (hasil: any) => {
      console.log('berhasil edit Resep');
      this.resetModal();
      this.getResep();
      this.modalEdit = false;
      this.modal.dismiss();
    },
    error: (err: any) => {
      console.log('gagal edit Resep');
    }
  });
}

```

Delete



Logika untuk menghandle penghapusan data, yaitu

```
async confirmDelete(id: any) {
  const alert = await this.alertController.create({
    header: 'Konfirmasi Hapus',
    message: 'Apakah Anda yakin ingin menghapus data resep ini?',
    buttons: [
      {
        text: 'Hapus',
        handler: () => {
          this.hapusResep(id);
        }
      },
      {
        text: 'Batal',
        role: 'cancel',
        handler: () => {
          console.log('Hapus dibatalkan');
        }
      }
    ]
  });
  await alert.present();
}

hapusResep(id: any) {
  this.api.hapus(id, 'hapus.php?id=').subscribe({
    next: (res: any) => {
      console.log('berhasil hapus data');
      this.getResep();
    },
    error: (error: any) => {
      console.log('gagal');
    }
  });
}
```

3. Koneksi dengan database (database prak)

```
<?php
header('Access-Control-Allow-Origin: *');
header('Access-Control-Allow-Credentials: true');
header('Access-Control-Allow-Methods: PUT, GET, HEAD, POST, DELETE, OPTIONS');
header('Content-Type: application/json; charset=utf-8');
header('Access-Control-Allow-Headers: X-Requested-With, Content-Type, Origin, Authorization, Accept, Client-Security-Token, Accept-Encoding');
$con = mysqli_connect('localhost', 'root', '', 'prak') or die("koneksi gagal");
```

BE untuk menhandle login

```
<?php
require 'koneksi.php';
$input = file_get_contents('php://input');
$data = json_decode($input, true);
$pesan = [];
$username = trim($data['username']);
$password = md5(trim($data['password']));
$query = mysqli_query($con, "select * from user where username='$username' and password='$password'");
$jumlah = mysqli_num_rows($query);
if ($jumlah != 0) {
    $value = mysqli_fetch_object($query);
    $pesan['username'] = $value->username;
    $pesan['token'] = time() . '_' . $value->password;
    $pesan['status_login'] = 'berhasil';
} else {
    $pesan['status_login'] = 'gagal';
}
echo json_encode($pesan);
echo mysqli_error($con);
```

BE untuk menhandle create/tambah

```
<?php
require 'koneksi.php';
$input = file_get_contents('php://input');
$data = json_decode($input, true);
$nama_makanan = trim($data['nama_makanan']);
$langkah = trim($data['langkah']);
http_response_code(201);
if ($nama_makanan != '' and $langkah != '') {
    $query = mysqli_query($con, "insert into makanan(nama_makanan,langkah) values('$nama_makanan','$langkah')");
    $pesan = true;
} else {
    $pesan = false;
}
echo json_encode($pesan);
echo mysqli_error($con);
```

BE untuk menhandle read/lihat makanan/resep dengan id tertentu


```

lihat.php > ...
1  <?php
2  require 'koneksi.php';
3  $data = [];
4  $id = $_GET['id'];
5  $query = mysqli_query($con, "select * from makanan where id ='$id'");
6  $jumlah = mysqli_num_rows($query);
7  if ($jumlah == 1) {
8      $row = mysqli_fetch_object($query);
9      $data = $row;
10 }
11 echo json_encode($data);
12 echo mysqli_error($con);

```

BE untuk menghandle tampil/read all

```

tampil.php > ...

1  <?php
2  header("Access-Control-Allow-Origin: *");
3  header("Access-Control-Allow-Methods: GET, POST, OPTIONS");
4  header("Access-Control-Allow-Headers: Content-Type, Authorization");
5  require 'koneksi.php';
6  $data = [];
7  $query = mysqli_query($con, "select * from makanan");
8  while ($row = mysqli_fetch_object($query)) {
9      $data[] = $row;
10 }
11 echo json_encode($data);
12 echo mysqli_error($con);

```

BE untuk menghandle delete

```
hapus.php > ...
1  <?php
2  require 'koneksi.php';
3  $input = file_get_contents('php://input');
4  $data = json_decode($input, true);
5  $pesan = [];
6  $id = $_GET['id'];
7  $query = mysqli_query($con, "delete from makanan where id='$id'");
8  if ($query) {
9      http_response_code(201);
10     $pesan['status'] = 'sukses';
11 } else {
12     http_response_code(422);
13     $pesan['status'] = 'gagal';
14 }
15 echo json_encode($pesan);
16 echo mysqli_error($con);
```

BE untuk menghandle edit/update

```
edit.php > ...
1  <?php
2  require 'koneksi.php';
3  $input = file_get_contents('php://input');
4  $data = json_decode($input, true);
5  $id = trim($data['id']);
6  $nama_makanan = trim($data['nama_makanan']);
7  $langkah = trim($data['langkah']);
8  http_response_code(201);
9  if ($nama_makanan != '' and $langkah != '') {
10     $query = mysqli_query($con, "update makanan set nama_makanan='$nama_makanan',langkah='$langkah' where
11 id='$id'");
12     $pesan = true;
13 } else {
14     $pesan = false;
15 }
16 echo json_encode($pesan);
17 echo mysqli_error($con);
```

SOAL

1. Sebuah menu dalam aplikasi memerlukan load data setiap kali menu tersebut diakses, pada fungsi apakah load data tersebut dilakukan untuk framework Angular dan Vue?

Angular

Pada Angular, data biasanya di-load di dalam `ngOnInit()` lifecycle hook.

Contoh:

```
ngOnInit(): void {  
  this.loadData();  
}
```

`ngOnInit()` akan dijalankan setiap kali komponen diinisialisasi, sehingga cocok untuk memuat data saat menu tersebut diakses.

Vue

Pada Vue, data bisa di-load di dalam `mounted()` lifecycle hook.

Contoh:

```
mounted() {  
  this.loadData();  
}
```

`mounted()` dipanggil setelah elemen DOM terpasang, menjadikannya tempat ideal untuk memuat data saat komponen ditampilkan.

Jika menu dapat diakses berkali-kali tanpa me-reload komponen (misalnya dengan router), Angular dapat menggunakan **Router Events**, sementara Vue bisa memanfaatkan **beforeRouteEnter** atau **watch** pada `$route`.

2. Pada menu profil, komponen bawaan Ionic apa saja yang dapat diterapkan untuk membangun menu tersebut? (min. 3 dan sebutkan untuk apa fungsinya)

Komponen bawaan ionic yang dapat diterapkan pada menu profil yaitu:

<ion-avatar>

Untuk menampilkan foto profil pengguna dalam bentuk lingkaran.

<ion-list>

Untuk membuat daftar menu atau informasi profil, seperti nama, email, atau pengaturan akun.

<ion-button>

Untuk menyediakan tombol aksi seperti "Edit Profil" atau "Logout".

<ion-card> (opsional)

Untuk menampilkan informasi profil pengguna dengan desain seperti card tertentu/.

3. Pada Firebase, apa perbedaan antara Cloud Firestore dan Realtime Database?

Aspek	Cloud Firestore	Realtime Database
Struktur Data	Data disimpan dalam bentuk koleksi dan dokumen (mirip JSON)	Data disimpan dalam format JSON tree sederhana.

	dengan hierarki terstruktur).	
Kinerja Query	Mendukung query kompleks dengan filter, sorting, dan kombinasi query.	Mendukung query sederhana (terbatas).
Scalling	Secara otomatis menskalakan untuk aplikasi besar dan mendukung banyak koleksi serta indeks.	Skalabilitas lebih terbatas dibandingkan Firestore.
Realtime	Mendukung pembaruan data realtime tetapi fokus pada efisiensi query.	Berorientasi pada pembaruan realtime, ideal untuk aplikasi dengan sinkronisasi tinggi.
Biaya	Dihitung berdasarkan dokumen yang dibaca, ditulis, atau dihapus.	Dihitung berdasarkan bandwidth dan penyimpanan data.
Keadaan Regional	Replikasi data multi-regional secara default (lebih aman).	Replikasi tunggal (kurang aman terhadap kegagalan regional).