

# LAMPIRAN

## Lampiran 1. Surat Kesiediaan Bimbingan

### SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan di bawah ini :

Pihak Pertama

Nama : Putri Zulfiaturrizki  
NIM : 20090138  
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

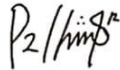
Nama : Dwi Intan Af'idah, S.T., M.Kom.  
Status : Dosen Tetap  
NIDN : 0620089303  
Jabatan Fungsional : Asisten Ahli  
Pangkat/Golongan : Penata Muda Tk.I/III

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

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi

Tegal, 14 Maret 2024

Pihak Pertama



Putri Zulfiaturrizki

Pihak Kedua



Dwi Intan Af'idah, 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 : Putri Zulfiaturrizki  
NIM : 20090138  
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

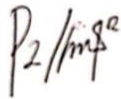
Nama : Rosid Mustofa, S.Kom, M.Kom  
Status : Dosen Praktisi  
NIDN : -  
Jabatan Fungsional : -  
Pangkat/Golongan : -

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

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi

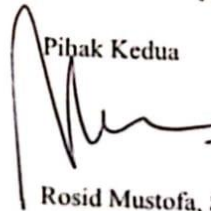
Tegal, 14 Maret 2024

Pihak Pertama



Putri Zulfiaturrizki

Pihak Kedua



Rosid Mustofa, S.Kom, M.Kom

Mengetahui  
Ketua Program Studi Sarjana Terapan Teknik Informatika



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

## Lampiran 2. Surat Pernyataan Pengajuan HKI

### SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta:

1. Nama : Putri Zulfiaturrizki  
Kewarganegaraan : Indonesia  
Alamat : Desa Karangdawa RT 04/RW 03, Kecamatan Margasari, Kabupaten Tegal, 52463.
2. Nama : Dwi Intan Afidah, S.T., M.Kom.  
Kewarganegaraan : Indonesia  
Alamat : Desa Grinting RT 003/RW 001, Kecamatan Bulakamba, Kabupaten Brebes, 52253.
2. Nama : Rosid Mustofa, S.Kom., M.Kom.  
Kewarganegaraan : Indonesia  
Alamat : Desa Kedungbanteng, RT 25/RW 12, Kecamatan Kedungbanteng, Kabupaten Tegal, 52472.

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan :  
Berupa : Program Komputer  
Berjudul : Aplikasi Mobile untuk Deteksi Kualitas Ikan Segar Berdasarkan Kondisi Mata dan Insang
  - 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 melanggar, 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 berperkara dan/atau sedang dalam gugatan di Pengadilan maka status kepemilikan surat pencatatan elektronik tersebut ditangguhkan menunggu putusan Pengadilan yang berkekuatan hukum tetap.

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

Tegal, Juli 2024



(Putri Zulfiaturrizki)

(Dwi Intan Afidah, S.T., M.Kom.)

(Rosid Mustofa, S.Kom., M.Kom.)

### Lampiran 3. Surat Pengalihan Hak Cipta

#### SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini:

1. Nama : Putri Zulfiaturrizki  
Kewarganegaraan : Indonesia  
Alamat : Desa Karangdawa RT 04/RW 03, Kecamatan Margasari, Kabupaten Tegal, 52463,
2. Nama : Dwi Intan Afidah, S.T., M.Kom.  
Kewarganegaraan : Indonesia  
Alamat : Desa Grinting RT 003/RW 001, Kecamatan Bulakamba, Kabupaten Brebes, 52253.
2. Nama : Rosid Mustofa, S.Kom., M.Kom.  
Kewarganegaraan : Indonesia  
Alamat : Desa Kedungbanteng, RT 25/RW 12, Kecamatan Kedungbanteng, Kabupaten Tegal, 52472.

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

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

Adalah Pihak II selaku Pemegang Hak Cipta berupa Program Komputer dengan judul "Aplikasi Mobile untuk Deteksi Kualitas Ikan Segar Berdasarkan Kondisi Mata dan Insang". 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, Juli 2024  
Pencipta

Pemegang Hak Cipta  
Ketua P3M



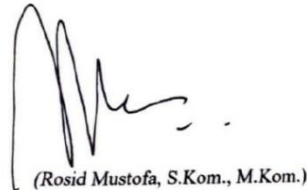
(Dr. Aldi Budi Riyanta, S.Si., M.T)



(Putri Zulfiaturrizki)



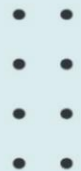
(Dwi Intan Afidah, S.T., M.Kom.)



(Rosid Mustofa, S.Kom., M.Kom.)

**Lampiran 4. Manual Book**

# MANUAL BOOK



## **APLIKASI DETEKSI KUALITAS IKAN SEGAR BERDASARKAN KONDISI MATA DAN INSANG**

**Oleh:**

Putri Zulfiaturrizki

Dwi Intan Afidah, S.T., M.Kom

Rosid Mustofa, S.Kom., M.Kom

## DAFTAR ISI

DAFTAR ISI .....	2
DAFTAR GAMBAR .....	3
1. PENDAHULUAN .....	4
1.1. Tujuan Pembuatan Dokumen .....	4
1.2. Deskripsi Umum Aplikasi .....	4
1.3. Deskripsi Dokumen .....	4
2. PERANGKAT YANG DIBUTUHKAN .....	4
2.1. Perangkat Lunak .....	4
2.2. Perangkat Keras .....	5
3. MENU DAN CARA PENGGUNAAN .....	5
3.1. Struktur Menu .....	5
3.2. Cara Penggunaan .....	7
1. Install Aplikasi .....	7
2. <i>SplashScreen</i> .....	7
3. Halaman <i>Onboarding</i> .....	8
4. Halaman Registrasi .....	10
5. Halaman <i>Login</i> .....	11
6. Halaman Utama .....	12
7. Halaman Deteksi (Fitur Utama) .....	13
8. Halaman <i>History</i> .....	17
9. Halaman <i>Report</i> .....	18
10. Halaman Resep .....	19
11. Halaman Artikel .....	20
12. Halaman Profil Pengguna .....	22

## DAFTAR GAMBAR

Gambar 1. <i>Install</i> Aplikasi .....	7
Gambar 2. Tampilan <i>Splash Screen</i> .....	8
Gambar 3. Tampilan Halaman <i>Onboarding</i> .....	9
Gambar 4. Tampilan Halaman Registrasi .....	10
Gambar 5. Tampilan Verifikasi Email.....	11
Gambar 6. Tampilan Halaman <i>Login</i> .....	12
Gambar 7. Tampilan Halaman Utama.....	13
Gambar 8. Tampilan Deteksi Mata .....	15
Gambar 9. Tampilan Deteksi Insang.....	16
Gambar 10. Tampilan Kesimpulan Hasil Deteksi.....	17
Gambar 11. Tampilan Halaman <i>History</i> .....	18
Gambar 12. Tampilan Halaman <i>Report</i> .....	19
Gambar 13. Tampilan Halaman Resep .....	20
Gambar 14. Tampilan Halaman Artikel .....	21
Gambar 15. Tampilan Halaman Detail Profile.....	22
Gambar 16. Tampilan Halaman Edit <i>Username</i> .....	23
Gambar 17. Tampilan Halaman Edit <i>Password</i> .....	24

## **1. PENDAHULUAN**

### **1.1. Tujuan Pembuatan Dokumen**

Dokumen *user manual* ini dibuat untuk memberikan penjelasan dan gambaran atau sebagai panduan kepada pengguna mengenai penggunaan Aplikasi *Mobile* untuk Deteksi Kualitas Ikan Segar Berdasarkan Kondisi Mata Dan Insang menggunakan Yolov5.

### **1.2. Deskripsi Umum Aplikasi**

Aplikasi ini dirancang untuk membantu pengguna dalam menilai dan mendeteksi kualitas ikan segar berdasarkan kondisi mata dan insang.

### **1.3. Deskripsi Dokumen**

Dokumen ini dibuat untuk memberikan panduan penggunaan Aplikasi *Mobile* Untuk Deteksi Kualitas Ikan Segar Berdasarkan Kondisi Mata Dan Insang.

#### 1. Bagian 1

Bagian ini berisi informasi umum yang meliputi tujuan pembuatan dokumen, deskripsi umum sistem, dan deskripsi dokumen.

#### 2. Bagian 2

Bagian ini berisi perangkat yang dibutuhkan untuk penggunaan aplikasi meliputi perangkat keras dan perangkat lunak.

#### 3. Bagian 3

Bagian ini berisi panduan atau cara penggunaan Aplikasi *Sidefish*.

## **2. PERANGKAT YANG DIBUTUHKAN**

### **2.1. Perangkat Lunak**

Untuk menjalankan aplikasi ini, berikut beberapa perangkat lunak yang dibutuhkan yaitu:

1. Sistem Operasi Android. Aplikasi ini harus dijalankan pada perangkat yang menggunakan sistem operasi Android.
2. Koneksi Internet. Untuk menjalankan aplikasi dengan lancar, diperlukan koneksi internet yang stabil.
3. Penyimpanan Data.



## 2.2. Perangkat Keras

Selain perangkat lunak, aplikasi ini juga membutuhkan perangkat keras yaitu yang paling utama adalah *Smartphone* Android. Perangkat harus dilengkapi dengan kamera yang memiliki resolusi cukup baik untuk menangkap gambar ikan dengan jelas dan detail.

## 3. MENU DAN CARA PENGGUNAAN

### 3.1. Struktur Menu

Berikut struktur menu yang ada pada aplikasi SideFish adalah sebagai berikut:

1. *Splash Screen*  
Tampilan awal saat aplikasi dibuka. Menampilkan logo aplikasi sebelum pengguna diarahkan ke halaman login.
2. *Onboarding*  
Halaman ini akan memperkenalkan fitur-fitur yang disediakan.
3. Registrasi  
Menu ini digunakan untuk mendaftarkan akun baru. Pengguna harus mengisi informasi seperti nama, email, dan kata sandi untuk membuat akun.
4. *Login*  
Menu ini digunakan untuk masuk ke aplikasi dengan akun yang sudah terdaftar. Pengguna harus memasukkan email dan kata sandi yang telah didaftarkan.
5. Deteksi  
Menu ini digunakan untuk melakukan deteksi kualitas ikan. Pengguna akan diarahkan untuk mengambil foto mata dan insang ikan melalui kamera perangkat dan akan muncul hasil deteksi
6. Kesimpulan Deteksi  
Menu ini digunakan untuk menampilkan hasil kesimpulan dari ikan yang sudah di deteksi berdasarkan dua parameter yaitu mata dan insang.
7. *History*  
Menu ini menampilkan riwayat hasil deteksi ikan yang telah dilakukan oleh pengguna. Setiap entri riwayat mencakup informasi mengenai kondisi ikan, tanggal deteksi, dan lokasi deteksi.

8. *Report*

Menu ini menampilkan laporan berdasarkan data riwayat deteksi dari semua pengguna. Laporan disusun berdasarkan lokasi dan memberikan informasi tentang jumlah ikan yang terdeteksi, kondisi ikan segar, dan kondisi ikan tidak segar.

9. *Resep*

Menu ini menyediakan kumpulan resep masakan berbahan dasar ikan. Pengguna dapat mencari resep berdasarkan jenis ikan dan melihat detail cara memasaknya.

10. *Artikel*

Menu ini berisi artikel-artikel informatif seperti cara memilih ikan segar, tips memasak ikan, dan informasi lain.

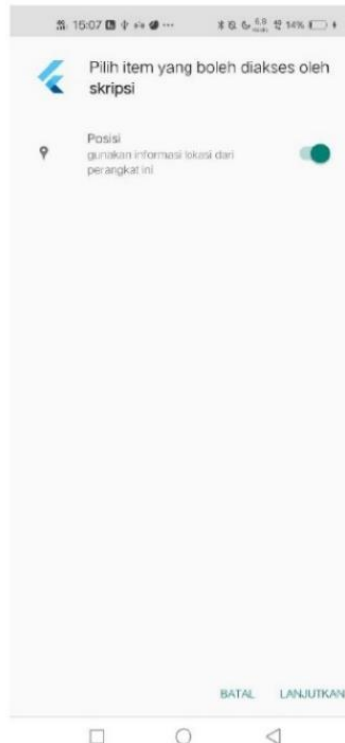
11. *Profile*

Menu ini digunakan untuk melihat dan mengedit informasi profil mereka, seperti nama, email, dan kata sandi.

### 3.2. Cara Penggunaan.

#### 1. Install Aplikasi

Izinkan lokasi saat pertama kali menginstall aplikasi. Klik “Lanjutkan” untuk melanjutkan proses install aplikasi. Berikut contoh tampilan nya dapat dilihat pada Gambar 1.



**Gambar 1.** Install Aplikasi

#### 2. *SplashScreen*

Pastikan *smartphone* terhubung ke internet. Buka aplikasi dan tunggu beberapa saat. Pada saat pertama kali membuka aplikasi, Anda akan disambut dengan *splashscreen*. Layar ini menampilkan logo aplikasi dan berlangsung selama beberapa detik sebelum melanjutkan ke halaman *onboarding*. Layar tersebut bisa dilihat pada Gambar 2.

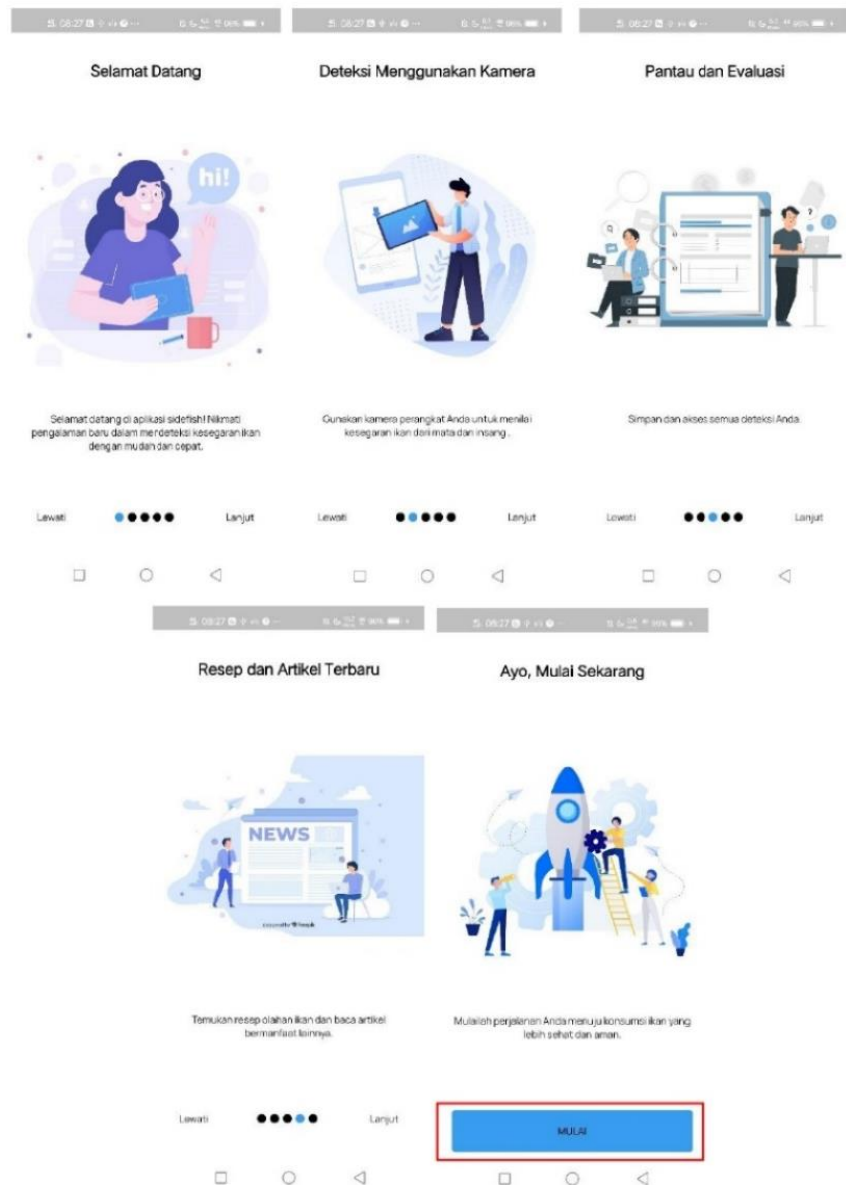


**Gambar 2.** Tampilan *Splash Screen*

### 3. Halaman *Onboarding*

Setelah itu, aplikasi akan menampilkan halaman *onboarding*. Halaman ini akan memperkenalkan fitur-fitur yang disediakan. Halaman *onboarding* bisa dilihat pada Gambar 2. Berikut langkah-langkah nya:

- 1) Halaman ini terdapat text yaitu “Lewati” dan “Lanjut”.
- 2) Ketika Anda ingin melihat semua halaman *onboarding*, Anda bisa geser layer ke kanan atau klik “Lanjut” dan aplikasi akan melanjutkan halaman tersebut. Klik tombol “Mulai” untuk melanjutkan ke halaman login.
- 3) Tetapi ketika Anda ingin melewati halaman *onboarding*, Anda bisa klik “Lewati” dan akan langsung diarahkan ke halaman login.



**Gambar 3.** Tampilan Halaman *Onboarding*

#### 4. Halaman Registrasi

Pada halaman ini, Anda akan diarahkan untuk membuat akun baru supaya dapat mengakses semua fitur pada aplikasi. Berikut langkah-langkah nya:

- 1) Isi form pendaftaran dengan informasi yang diperlukan seperti username, email, dan *password* yang ingin Anda gunakan, serta *ulangi password* untuk memastikan Anda mengingatnya dan tidak ada kesalahan pengetikan.
- 2) Klik tombol "Daftar" untuk membuat akun baru.

Catatan:

- 1) Pada bagian bawah, terdapat text "Sudah Memiliki Akun? Silahkan Masuk". Text tersebut digunakan ketika Anda sudah mempunyai akun dan bisa langsung masuk ke aplikasi. Berikut tampilan halaman registrasi dapat dilihat pada Gambar 4.



**Gambar 4.** Tampilan Halaman Registrasi



- 3) Untuk melanjutkan proses registrasi, pengguna akan diarahkan ke halaman verifikasi email untuk memastikan email yang digunakan benar. Silahkan periksa email Anda dan masukkan kode 4 digit angka dan klik “Verifikasi”.
- 4) Muncul pop up akun berhasil verifikasi, klik “Oke”. Tampilan tersebut dapat dilihat pada Gambar 5.



**Gambar 5.** Tampilan Verifikasi Email

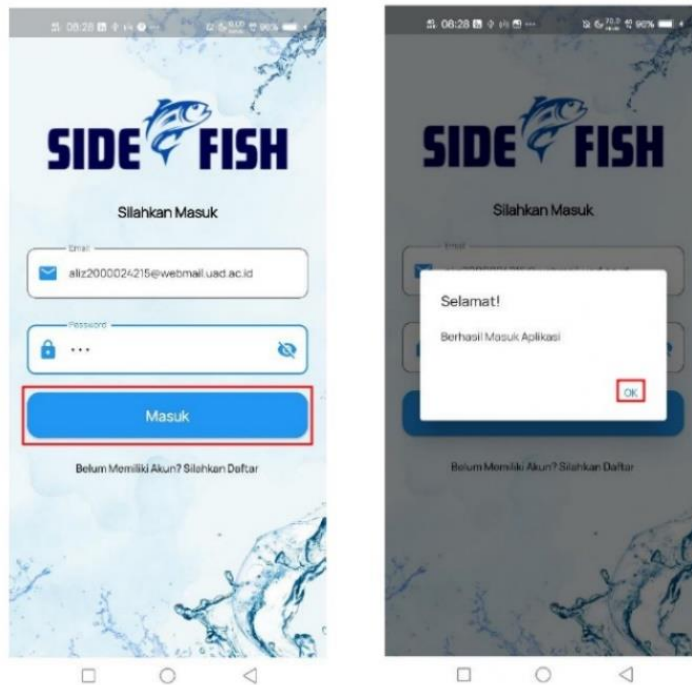
#### 5. Halaman Login

Pada halaman ini, Anda dapat masuk ke aplikasi dengan menggunakan email dan *password* yang telah didaftarkan sebelumnya. Berikut langkah-langkahnya:

- 1) Pada halaman login, aplikasi akan menampilkan dua form yaitu Email dan *Password*.
- 2) Isi form tersebut menggunakan akun yang sudah Anda daftarkan sebelumnya.
- 3) Setelah itu, klik tombol “Masuk” dan aplikasi akan menampilkan popup Berhasil Masuk Aplikasi, klik “Oke” untuk melanjutkan. Kemudian, Anda akan diarahkan ke halaman utama aplikasi.

Catatan:

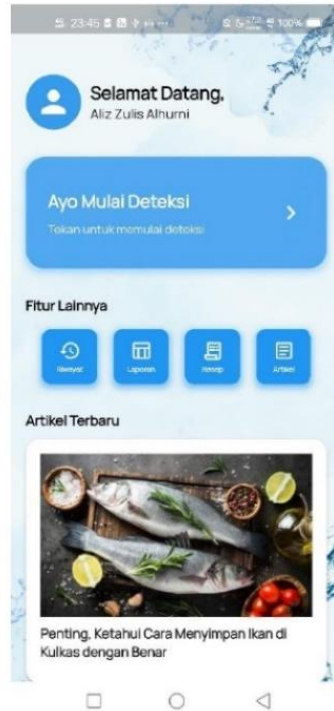
- 1) Pada bagian bawah halaman ini, terdapat Text Tombol **“Belum Memiliki Akun? Silahkan Daftar”**. Tombol tersebut digunakan ketika Anda belum memiliki akun dan akan diarahkan ke halaman registrasi untuk melakukan pendaftaran. Tampilan halaman ini dapat dilihat pada Gambar 6.



**Gambar 6.** Tampilan Halaman Login

## 6. Halaman Utama

Pada halaman utama merupakan halaman yang dilihat setelah Anda berhasil login ke aplikasi. Halaman ini merupakan pusat navigasi yang menyediakan akses ke berbagai fitur aplikasi seperti deteksi, riwayat deteksi, report dan juga melihat daftar resep atau artikel yang bermanfaat. Tampilan halaman ini dapat dilihat pada Gambar 7.



**Gambar 7.** Tampilan Halaman Utama

## 7. Halaman Deteksi (Fitur Utama)

Halaman deteksi merupakan fitur utama dari aplikasi yang memungkinkan pengguna untuk melakukan deteksi kualitas ikan. Deteksi dilakukan melalui dua langkah yaitu berdasarkan mata ikan dan berdasarkan insang ikan. Berikut penjelasan dan langkah-langkah nya:

### 1) Langkah Pertama (Deteksi Bagian Mata).

Berikut langkah-langkah yang harus Anda lakukan, yaitu:

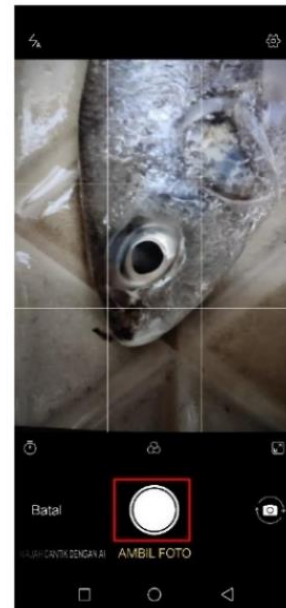
- a. Pada halaman utama, Anda bisa klik tombol “Ayo Mulai Deteksi” dan aplikasi akan menampilkan halaman untuk deteksi.
- b. Klik tombol “Ambil Foto” dan aplikasi akan membuka kamera perangkat yang Anda gunakan.
- c. Arahkan kamera ke bagian **mata** ikan yang akan Anda beli, klik **tombol** bulat pada kamera perangkat Anda untuk ambil foto.

- d. Setelah itu hasil gambar yang Anda ambil akan keluar. Ada tiga opsi yang bisa Anda gunakan yaitu “Batal”, “Icon Ulangi” dan “Oke”.
- Text “Batal” yang ada di bagian kiri bisa Anda gunakan ketika ingin membatalkan proses deteksi yang dilakukan.
  - Ikon Ulangi yang berada di bagian Tengah bisa Anda gunakan ketika ingin mengulangi proses ambil foto.
  - Text “Oke” yang ada di bagian kanan bisa Anda gunakan ketika ingin melanjutkan proses deteksi yang dilakukan.
- e. Untuk melanjutkan proses deteksi, Anda klik “Oke” dan setelah beberapa detik aplikasi akan menampilkan hasilnya. Hasil yang ditampilkan yaitu gambar beserta label dan *bounding box* di area mata ikan. Dibagian bawah hasil deteksi, terdapat bagian Kondisi Mata, bagian ini berfungsi untuk memperjelas hasil deteksi yang sudah dilakukan.
- f. Selanjutnya, Anda klik tombol “Lanjut” untuk melanjutkan proses deteksi selanjutnya yaitu bagian insang ikan. Berikut tampilan dari langkah-langkah yang pertama dapat dilihat pada Gambar 8.

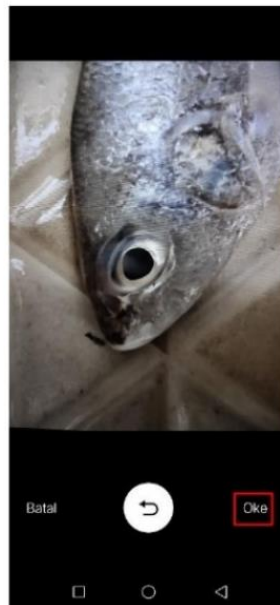
### 1. Klik Ambil Foto



### 2. Klik Icon Bulat



### 3. Klik Tombol Oke



### 4. Klik Tombol Lanjut



Gambar 8. Tampilan Deteksi Mata

## 2) Langkah Kedua (Deteksi Bagian Insang)

Berikut langkah-langkah yang harus Anda lakukan, yaitu:

- a. Setelah Anda sudah selesai melakukan deteksi pada bagian mata, Anda akan diarahkan ke halaman deteksi insang.
- b. Lakukan langkah yang sama ketika Anda melakukan deteksi mata. Berikut tampilan dari langkah-langkah yang pertama dapat dilihat pada Gambar 9.

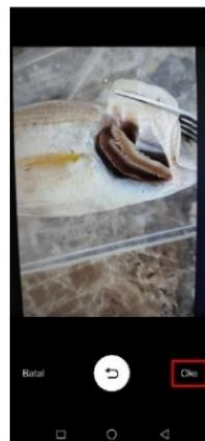
### 1. Klik Ambil Foto



### 2. Klik Icon Bulat



### 3. Klik Tombol Oke



### 4. Klik Tombol Lanjut

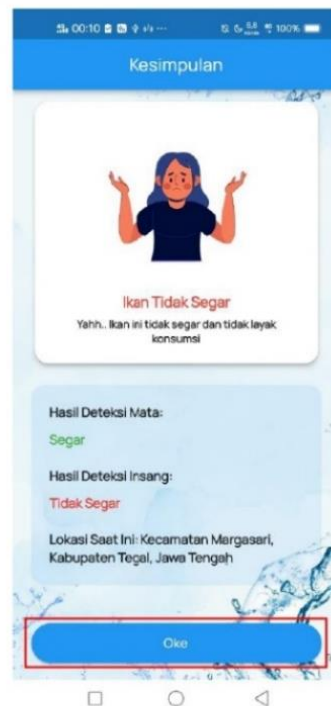


Gambar 9. Tampilan Deteksi Insang



### 3) Langkah Terakhir (Kesimpulan)

Pada halaman ini, Anda akan melihat kesimpulan dari deteksi yang sudah dilakukan. Kesimpulan diambil dari deteksi yang sudah dilakukan yaitu deteksi mata dan insang. Klik tombol “Oke” untuk kembali ke halaman utama. Tampilan halaman ini dapat dilihat pada Gambar 10.



**Gambar 10.** Tampilan Kesimpulan Hasil Deteksi

### 8. Halaman Riwayat Deteksi

Halaman ini digunakan untuk mencatat deteksi yang sudah dilakukan. Informasi yang diambil yaitu **Kesimpulan Hasil Deteksi**, **Tanggal**, dan **Lokasi** pada saat Anda melakukan deteksi ikan. Jika sudah, klik tombol “Kembali” untuk kembali ke halaman utama. Berikut tampilan pada halaman ini dapat dilihat pada Gambar 11.



**Gambar 11.** Tampilan Halaman Riwayat Deteksi

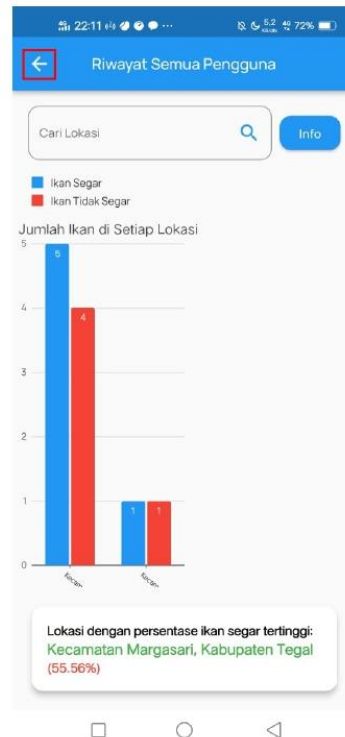
#### 9. Halaman Report

Halaman ini memberikan gambaran menyeluruh mengenai riwayat deteksi ikan dari semua pengguna aplikasi berdasarkan lokasi. Tampilan pada halaman ini dapat dilihat pada Gambar 12.

Berikut penjelasan mengenai halaman report, yaitu:

- 1) Jika Anda ingin melihat dan mempercepat untuk melihat lokasi yang Anda inginkan, Anda bisa menggunakan fitur pencarian.
- 2) Jika sudah, Anda bisa melihat report tersebut.
  - a. Lokasi. Kolom ini menunjukkan lokasi tempat deteksi dilakukan. Lokasi ditampilkan secara lengkap hingga level kecamatan dan kabupaten/kota.
  - b. Jumlah Ikan. Kolom ini menunjukkan total jumlah ikan yang telah dideteksi di lokasi tersebut.

- c. Ikan Segar: Kolom ini menunjukkan jumlah ikan yang terdeteksi dalam kondisi segar di lokasi tersebut.
  - d. Ikan Tidak Segar. Kolom ini menunjukkan jumlah ikan yang terdeteksi dalam kondisi tidak segar di lokasi tersebut.
- 3) Klik tombol “Kembali” yang ada di atas dan akan Kembali ke halaman utama.



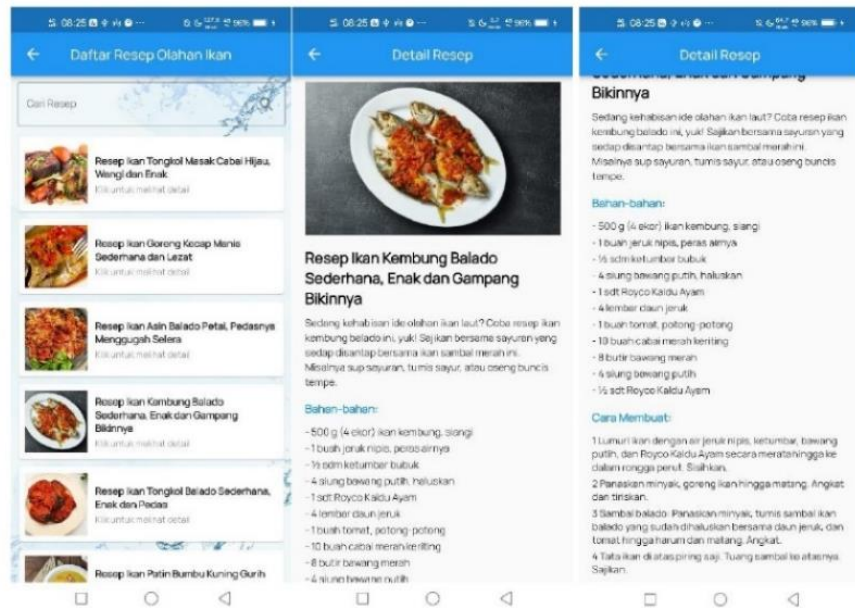
**Gambar 12.** Tampilan Halaman Report

#### 10. Halaman Resep

Halaman ini berisi kumpulan resep olahan tentang ikan yang diambil dari sebuah website. Berikut langkah-langkah nya:

- 1) Pilih resep yang Anda inginkan.
- 2) Klik resep tersebut.
- 3) Aplikasi menampilkan detail resep dari resep yang Anda pilih. Jika sudah, klik icon Kembali untuk Kembali ke halaman daftar resep.

- 4) Jika sudah, pada halaman daftar resep, klik icon kembali untuk kembali ke halaman utama. Tampilan pada halaman ini dapat dilihat pada Gambar 13.

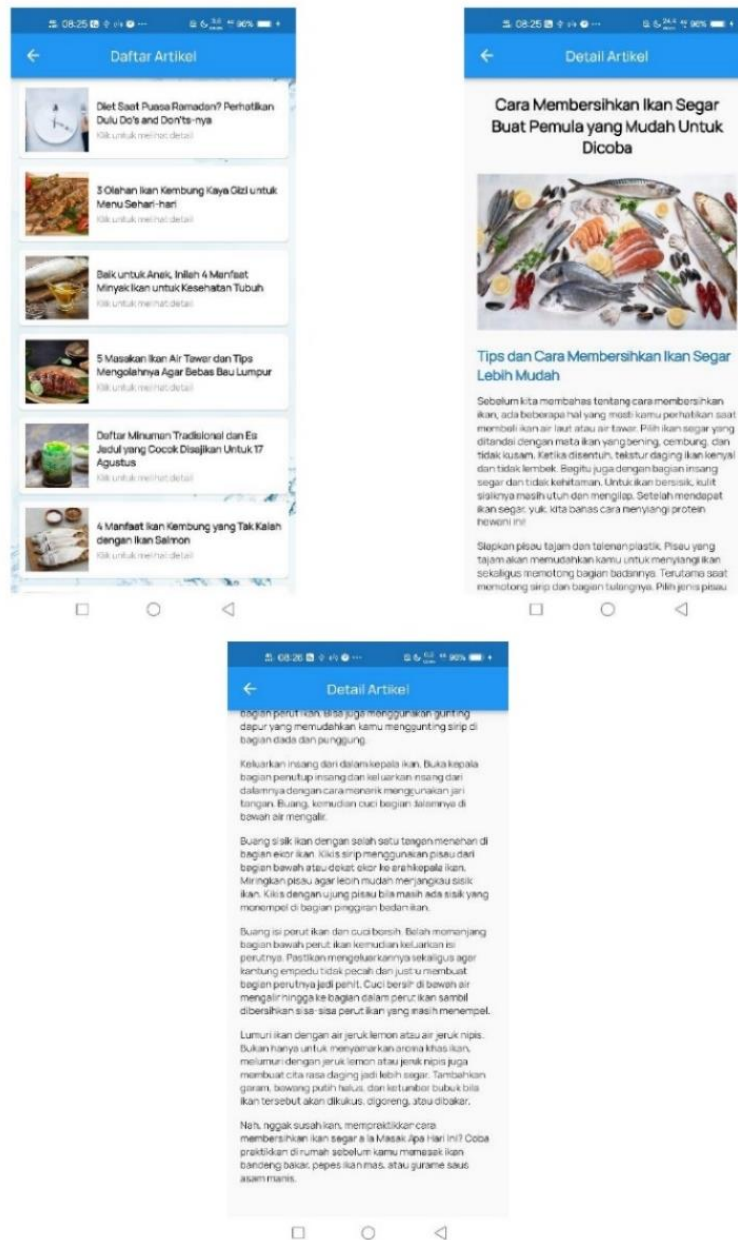


Gambar 13. Tampilan Halaman Resep

## 11. Halaman Artikel

Halaman ini menampilkan artikel bermanfaat yang bisa Anda baca. Berikut langkah-langkah nya:

- 1) Pilih artikel yang Anda inginkan.
- 2) Klik artikel tersebut.
- 3) Aplikasi menampilkan detail artikel yang Anda pilih. Jika sudah, klik icon kembali untuk kembali ke halaman daftar artikel.
- 4) Jika sudah, pada halaman daftar artikel klik icon kembali untuk kembali ke halaman utama. Tampilan pada halaman ini dapat dilihat pada Gambar 14.



Gambar 14. Tampilan Halaman Artikel

## 12. Halaman Profil Pengguna

Halaman ini menampilkan detail dari profil Anda. Terdapat *username* dan email yang Anda gunakan. Selain itu, Anda juga bisa melakukan edit *username* dan edit *password*. Berikut tampilan halaman profile dapat dilihat pada Gambar 15.

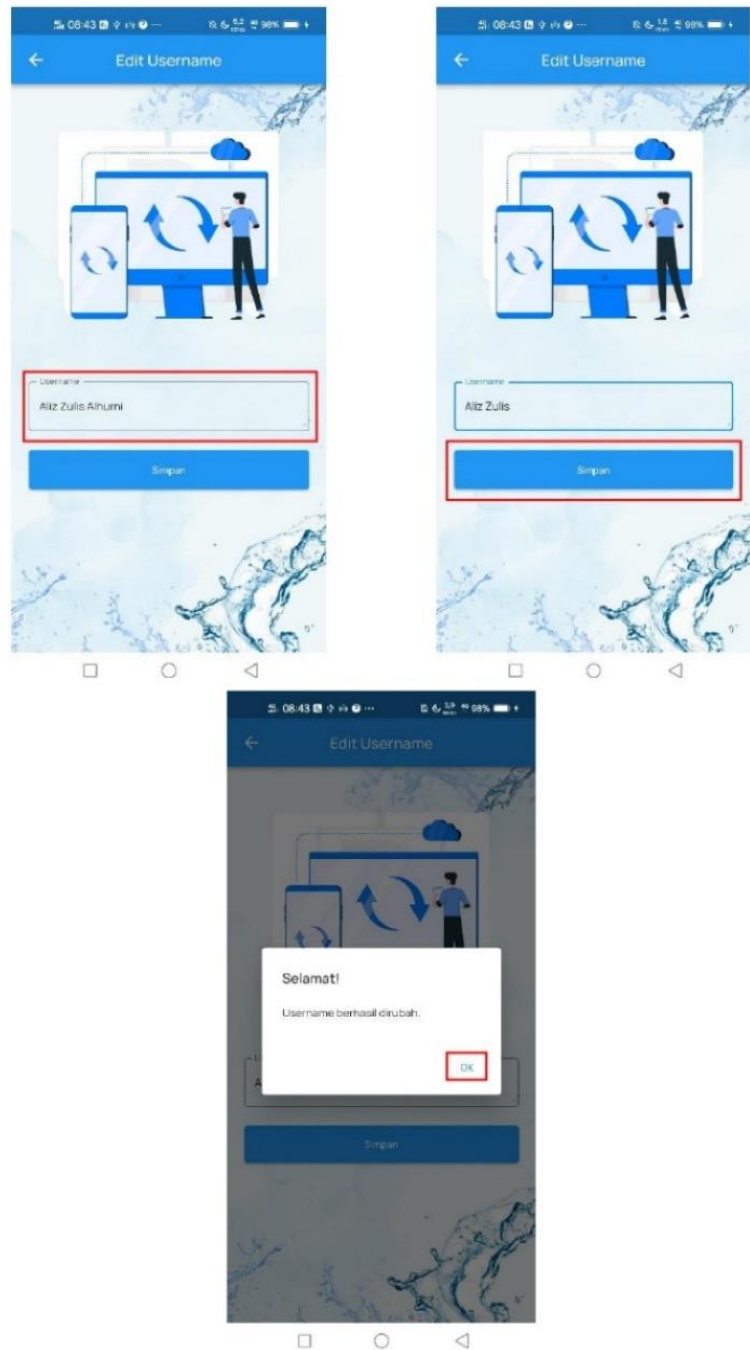


**Gambar 15.** Tampilan Halaman Detail Profile

### 1) Edit *Username*

Anda bisa melakukan edit *username* dengan klik “Edit *Username*” dan akan muncul *username* lama dan Anda bisa langsung menggantinya. Ketika sudah selesai, klik tombol “Simpan” dan akan muncul pop up “*Username* berhasil dirubah”. Berikut tampilan halaman edit *username* bisa dilihat pada Gambar 16.

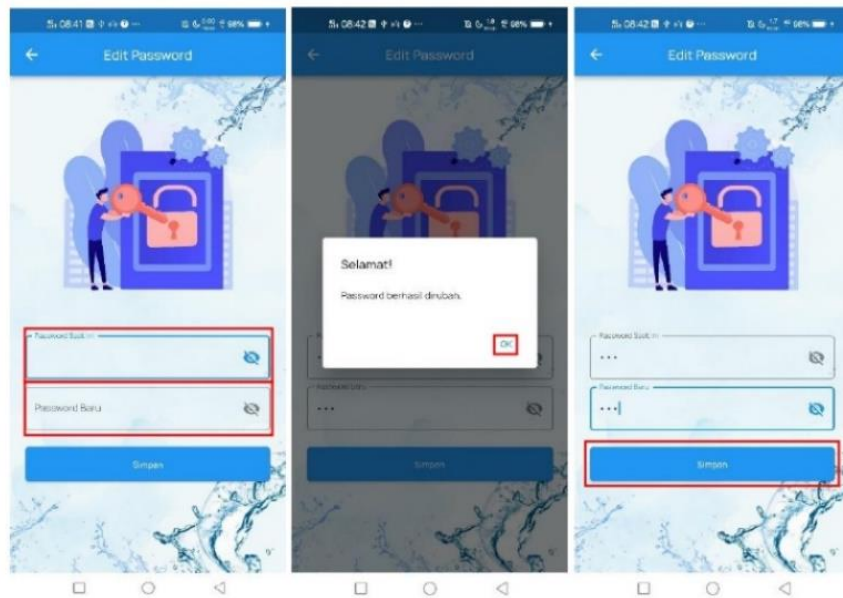




**Gambar 16.** Tampilan Halaman Edit *Username*

## 2) Edit *Password*

Anda bisa melakukan edit *username* dengan klik “Edit *Password*” dan akan muncul dua form untuk input “*Password Lama*” yang Anda gunakan dan juga ”*Password Baru*” yang akan Anda gunakan. Ketika sudah selesai, klik tombol “Simpan” dan akan muncul pop up “*Password berhasil dirubah*”. Berikut tampilan halaman edit *username* bisa dilihat pada Gambar 17.



**Gambar 17.** Tampilan Halaman Edit *Password*

# DOKUMEN TEKNIS



## APLIKASI DETEKSI KUALITAS IKAN SEGAR BERDASARKAN KONDISI MATA DAN INSANG

**Oleh:**

Putri Zulfiaturrizki

Dwi Intan Afidah, S.T., M.Kom

Rosid Mustofa, S.Kom., M.Kom

## DAFTAR ISI

DAFTAR ISI .....	2
DAFTAR GAMBAR .....	3
Profile.....	4
Latar Belakang .....	4
Manfaat .....	5
Spesifikasi Teknis .....	5

## DAFTAR GAMBAR

Gambar 1. 1. Source Code Tampilan Registrasi .....	6
Gambar 1. 2. Source Code Tampilan Verifikasi Email .....	7
Gambar 1. 3. Source Code Tampilan <i>Login</i> .....	8
Gambar 1. 4. Source Code Tampilan Utama .....	8
Gambar 1. 5. Source Code Tampilan Profil Pengguna .....	9
Gambar 1. 6. Source Code Tampilan Edit <i>Username</i> .....	10
Gambar 1. 7. Source Code Tampilan Edit <i>Password</i> .....	10
Gambar 1. 8. Source Code Tampilan Deteksi .....	11
Gambar 1. 9. Source Code Tampilan Kesimpulan Deteksi .....	11
Gambar 1. 10. Source Code Tampilan Riwayat Deteksi .....	12
Gambar 1. 11. Source Code Tampilan Report .....	13
Gambar 1. 12. Source Code Tampilan Daftar Resep .....	13
Gambar 1. 13. Source Code Tampilan Detail Resep .....	14
Gambar 1. 14. Source Code Tampilan Daftar Artikel .....	15
Gambar 1. 15. Source Code Tampilan Detail Artikel .....	15
Gambar 2. 1. Import Library .....	16
Gambar 2. 2. Route Registrasi .....	17
Gambar 2. 3. Route Verifikasi Email .....	18
Gambar 2. 4. Route <i>Login</i> .....	20
Gambar 2. 5. Route Edit <i>Username</i> .....	21
Gambar 2. 6. Route Edit <i>Password</i> .....	22
Gambar 2. 7. Route Riwayat Deteksi .....	24
Gambar 2. 8. Route Report .....	24
Gambar 2. 9. Route Scraping Website Resep .....	25
Gambar 2. 10. Route Scraping Detail Resep .....	27
Gambar 2. 11. Route Scraping Artikel .....	27
Gambar 2. 12. Route Scraping Detail Artikel .....	28

## **Profile**

SideFish adalah aplikasi untuk mendeteksi kualitas ikan segar berdasarkan kondisi mata dan insang menggunakan algoritma YOLOv5. Aplikasi ini digunakan untuk menilai kualitas ikan yang akan dibeli dengan cepat dan tepat hanya dengan mengambil gambar melalui kamera dan bisa langsung melihat hasil deteksi dengan memunculkan *bounding box* serta label pada mata ikan tanpa perlu khawatir terdapat kesalahan dalam memilih ikan yang akan dikonsumsi.

## **Latar Belakang**

Indonesia merupakan negara kepulauan terbesar di dunia yang terdiri dari 17.500 pulau dan bergaris pantai sepanjang 81.000 km. Menurut data Kementerian Kelautan dan Perikanan (KKP) sekitar 62% luas wilayah Indonesia adalah laut dan perairan. Berdasarkan data Angka Konsumsi Ikan (AKI) dari Kementerian Kelautan dan Perikanan (KKP), mencatat bahwa rata-rata konsumsi ikan di Indonesia pada tahun 2023 mencapai 58,48 kg per orang per tahun. Angka ini menunjukkan peningkatan sebesar 2,11% dibandingkan dengan tahun sebelumnya yaitu pada tahun 2022, dimana angka rata-rata konsumsi ikan di Indonesia mencapai 57,27 kg per orang per tahun. Data tersebut menunjukkan adanya peningkatan masyarakat dalam mengonsumsi ikan sebagai sumber protein yang sehat dan bergizi tinggi.

Ikan merupakan bahan pangan yang terkenal dengan kandungan gizi cukup tinggi yang dibutuhkan oleh tubuh, salah satunya yaitu protein. Pada dasarnya, manusia membutuhkan banyak protein untuk menjaga kesehatan tubuhnya. Akan tetapi, ikan termasuk bahan pangan yang mudah rusak dan mempunyai masa simpan yang pendek jika dibiarkan lama di suhu ruang dan tidak segera diolah. Hal tersebut berisiko membawa bakteri yang dapat menyebabkan seseorang keracunan makanan yang diakibatkan oleh histamin setelah mengonsumsi ikan yang tidak segar. Keracunan ini terjadi akibat mengonsumsi ikan yang tidak segar. Untuk menghindari risiko tersebut, sangat penting untuk dapat mengidentifikasi ciri-ciri kualitas ikan yang baik. Beberapa cara untuk melakukan identifikasi kualitas ikan segar yaitu dengan memperhatikan berbagai aspek seperti aroma, tekstur, warna fisik, kondisi mata dan insang dari ikan. Dari ciri-ciri tersebut yang paling mudah untuk dilihat adalah warna mata dan insang.

Berdasarkan latar belakang penelitian yang ada, diperlukan aplikasi yang dapat mendeteksi kualitas ikan segar berdasarkan warna mata dan insang. Dalam penelitian ini dilakukan deteksi kualitas ikan yaitu ikan segar dan ikan tidak segar menggunakan model YOLO (You Only Look Once). Melalui aplikasi ini, pengguna dapat menilai kualitas ikan dengan cepat dan tepat hanya dengan mengambil gambar melalui kamera dan bisa langsung melihat hasil deteksi dengan memunculkan bounding box serta label pada mata ikan tanpa perlu khawatir terdapat kesalahan dalam memilih ikan yang akan dikonsumsi.

### **Manfaat**

1. Aplikasi ini dapat membantu masyarakat dalam menilai kualitas ikan yang akan dikonsumsi sehingga dapat meningkatkan keamanan pangan dan kesehatan konsumen.
2. Meningkatkan kesadaran akan pentingnya konsumsi ikan yang berkualitas dalam menjaga kesehatan tubuh.
3. Mengurangi pemborosan makanan dengan memastikan bahwa ikan yang dibeli dan dikonsumsi berada dalam kondisi segar.

### **Spesifikasi Teknis**

Spesifikasi Teknis, meliputi:

1. Modul Pengguna
2. *Source Code*

Berikut uraian spesifikasi untuk pembangunan aplikasi:

1. Flutter
2. Python Flask
3. Google Collaboration
4. Flask-SQLAlchemy
5. Visual Studio Code
6. Google Chrome

Berikut uraian spesifikasi modul:

### 1. Modul untuk pengguna

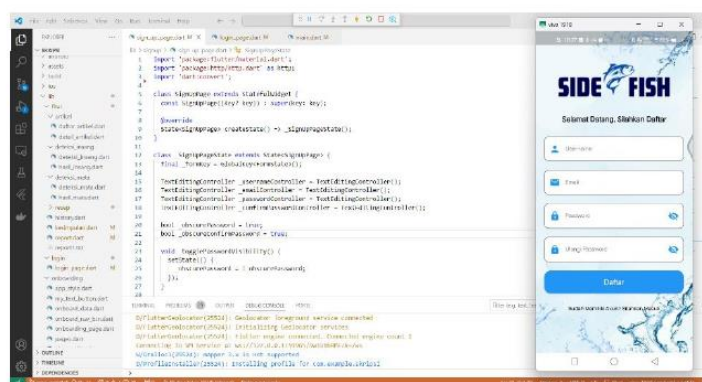
Bagian ini adalah aplikasi SideFish. Aplikasi ini menampilkan halaman utama dimana halaman tersebut menampilkan berbagai fitur yaitu fitur deteksi, riwayat pengguna, *report*, kumpulan resep olahan ikan, dan juga artikel yang dapat menambah wawasan bagi pengguna.

### 2. Source Code

#### 1) User Interface.

##### 1. Tampilan Registrasi

Tampilan registrasi ini menampilkan formulir yang terdiri dari empat input yaitu username, email, password, dan ulangi password. Masing-masing dikelola oleh 'TextEditingController' untuk mengontrol teks yang diinput pengguna, serta divalidasi menggunakan '\_formKey' untuk memastikan semua data diisi dengan benar. Halaman ini juga menyediakan tombol untuk mendaftar, yang memanggil fungsi '\_signUp()' untuk mengirim data ke server, dan tombol untuk beralih ke halaman login jika pengguna sudah memiliki akun. Fitur untuk mengubah visibilitas password menggunakan variabel boolean '\_obscurePassword' dan '\_obscureConfirmPassword' memungkinkan pengguna melihat atau menyembunyikan teks password dengan klik ikon mata. Selain itu, terdapat tombol navigasi berupa TextButton yang mengarahkan pengguna ke halaman login jika mereka sudah memiliki akun. *Source Code* tersebut dapat dilihat pada Gambar 1. 1.

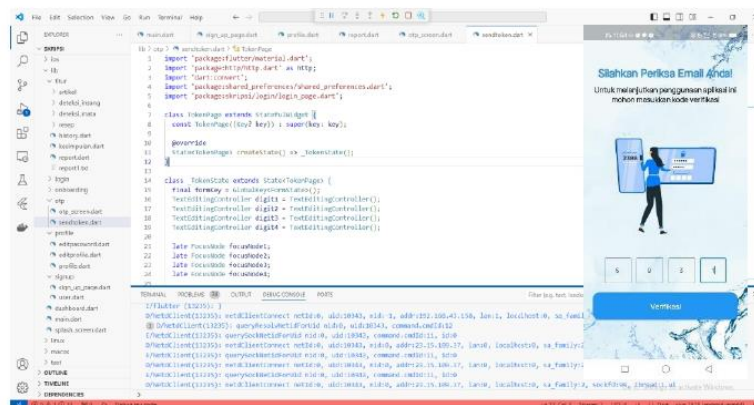


Gambar 1. 1. Source Code Tampilan Registrasi



## 2. Tampilan Verifikasi Email

Tampilan Verifikasi Email pada aplikasi ini menampilkan UI yang memungkinkan pengguna memasukkan kode verifikasi untuk memverifikasi email mereka. Konten utama terdiri dari teks instruksi, gambar ilustrasi, dan empat TextFormField untuk memasukkan digit kode verifikasi, masing-masing diatur dengan kontroler TextEditingController dan FocusNode untuk mengelola fokus input. Tombol verifikasi di bagian bawah mengirim permintaan verifikasi ke server. Jika verifikasi berhasil, pengguna akan melihat dialog konfirmasi dan diarahkan ke halaman login LoginPage. *Source code* tersebut dapat dilihat pada Gambar 1. 2.

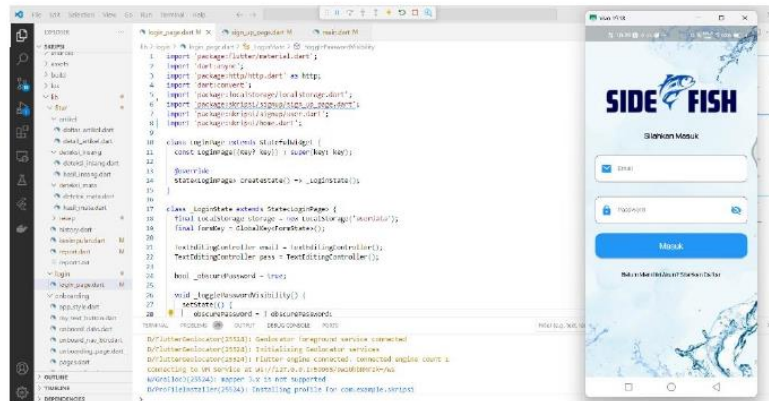


Gambar 1. 2. *Source Code* Tampilan Verifikasi Email

## 3. Tampilan Login

Tampilan *login* ini menyajikan sebuah formulir dengan dua input utama, yaitu email dan *password*, yang masing-masing dikelola oleh 'TextEditingController' untuk mengatur dan mengontrol teks yang dimasukkan pengguna. Formulir ini menggunakan 'formKey' untuk validasi, memastikan bahwa semua kolom terisi dengan benar sebelum melakukan proses *login*. Pengguna dapat mengubah visibilitas teks password dengan menekan ikon mata pada sisi kanan input password. Tombol "Masuk" dilakukan untuk proses *login* saat diklik, yang memanggil fungsi 'signIn()' untuk mengirim data ke server. Selain itu, terdapat tombol navigasi berupa TextButton yang mengarahkan

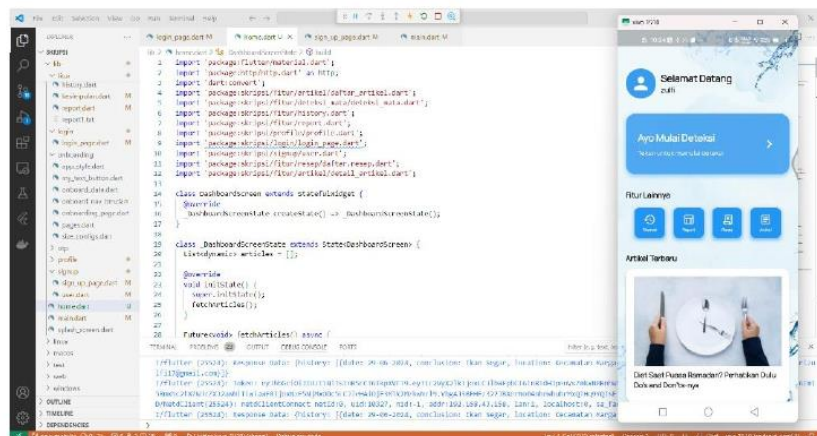
pengguna ke halaman registrasi jika mereka belum memiliki akun. *Source Code* tersebut dapat dilihat pada Gambar 1. 3.



**Gambar 1. 3. Source Code Tampilan Login**

#### 4. Tampilan Utama

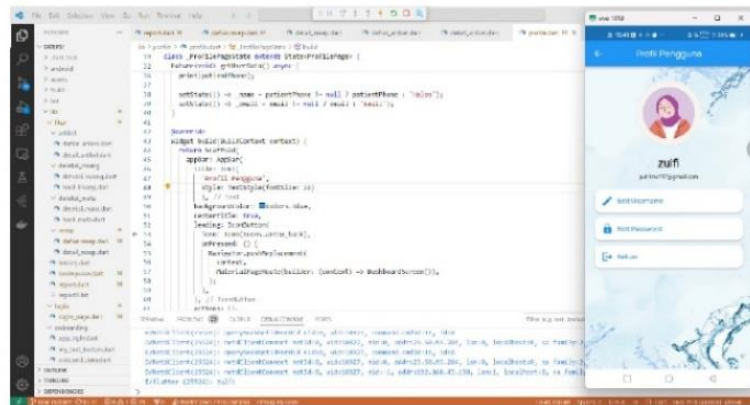
Tampilan utama menampilkan berbagai fitur yang tersedia. Terdapat tombol yang memungkinkan pengguna untuk memulai deteksi ikan dengan menavigasi ke halaman Deteksi Mata Ikan. Fitur lainnya ditampilkan dalam bentuk kartu dengan ikon yang menggambarkan fungsinya, seperti Riwayat, Report, Resep, dan Artikel. Halaman ini juga menampilkan daftar tiga artikel terbaru yang diambil dari server dengan menggunakan permintaan HTTP. *Source Code* tersebut dapat dilihat pada Gambar 1. 4.



**Gambar 1. 4. Source Code Tampilan Utama**

## 5. Tampilan Profil Pengguna

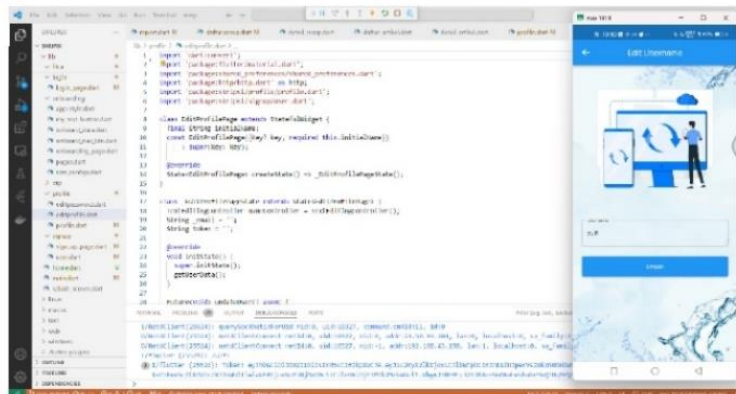
Halaman ini menampilkan profil pengguna, termasuk nama dan email. Setelah data berhasil diambil, halaman ini akan menampilkan foto profil pengguna, nama pengguna ‘\_name’, dan email pengguna ‘\_email’. Halaman ini juga menyediakan tombol untuk mengedit profil, mengedit kata sandi, dan logout. *Source Code* tersebut dapat dilihat pada Gambar 1. 5.



**Gambar 1. 5.** *Source Code* Tampilan Profil Pengguna

## 6. Tampilan Edit Username

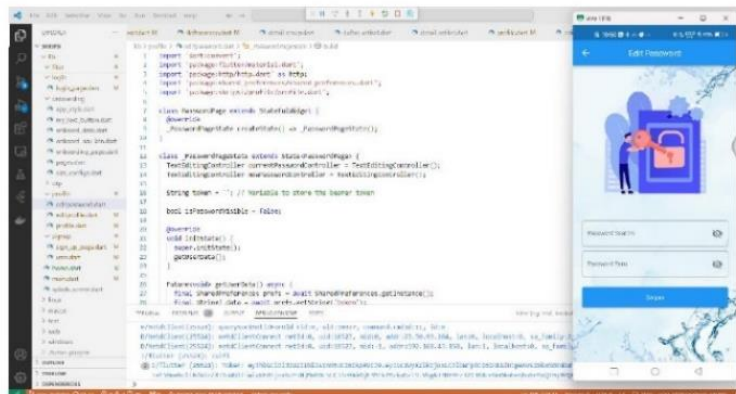
Halaman ini menampilkan UI untuk mengedit profil pengguna, dengan fokus pada perubahan username. Saat halaman diinisialisasi, data pengguna seperti nama, email, dan token akan diperbarui dikirim ke server melalui permintaan HTTP PUT ke URL yang ditentukan. Variabel utama yang digunakan adalah nameController untuk mengontrol input teks username, \_email untuk menyimpan email pengguna. *Source Code* tersebut dapat dilihat pada Gambar 1. 6.



**Gambar 1. 6. Source Code Tampilan Edit Username**

### 7. Tampilan Edit Password

Halaman ini menampilkan UI untuk mengubah kata sandi. Terdapat dua TextField untuk memasukkan kata sandi saat ini dan kata sandi baru dengan fitur toggle visibility yang diatur melalui variabel isPasswordVisible. Variabel yang digunakan adalah currentPasswordController dan newPasswordController untuk mengontrol input teks kata sandi. Source Code tersebut dapat dilihat pada Gambar 1. 7.



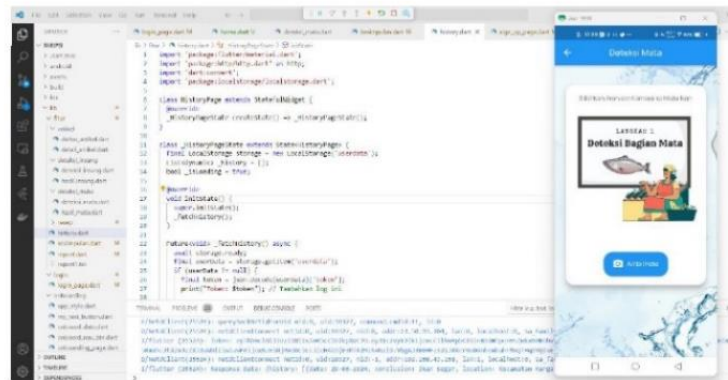
**Gambar 1. 7. Source Code Tampilan Edit Password**

### 8. Tampilan Deteksi

Tampilan deteksi ini menampilkan tombol “Ambil Foto” dengan ikon kamera, mengaktifkan fungsi untuk memulai proses pengambilan gambar dari kamera dan menavigasi pengguna ke halaman Hasil Deteksi Mata dengan hasil



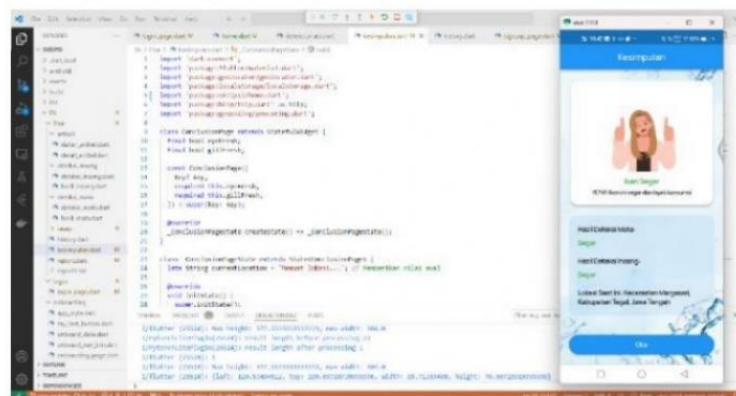
gambar yang ditangkap dan model deteksi objek PyTorch yang telah dimuat sebelumnya. *Source Code* tersebut dapat dilihat pada Gambar 1. 8.



**Gambar 1. 8.** *Source Code* Tampilan Deteksi

#### 9. Tampilan Kesimpulan Deteksi

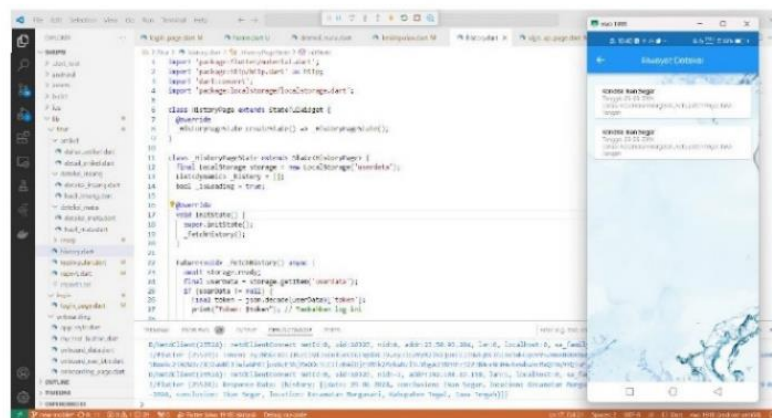
Halaman kesimpulan pada aplikasi ini menampilkan hasil deteksi kesegaran mata dan insang ikan berdasarkan variabel `eyeFresh` dan `gillFresh` yang bertipe boolean. Saat pengguna menavigasi ke halaman ini, aplikasi akan secara otomatis memuat lokasi saat ini menggunakan variabel `currentLocation`. Selain itu, terdapat lokasi saat ini yang diperoleh melalui proses asynchronous menggunakan `Geolocator`. Ketika pengguna menekan tombol “Oke”, informasi hasil deteksi dan lokasi saat ini disimpan ke dalam database server. *Source Code* tersebut dapat dilihat pada Gambar 1. 9.



**Gambar 1. 9.** *Source Code* Tampilan Kesimpulan Deteksi

## 10. Tampilan Riwayat Deteksi

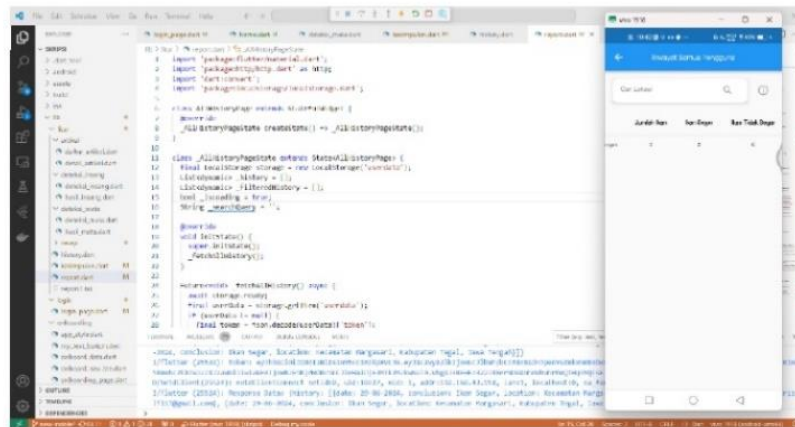
Ketika halaman ini dibuka, aplikasi mengambil token yang disimpan di LocalStorage dan menggunakan token tersebut untuk melakukan permintaan HTTP ke server. Permintaan ini bertujuan untuk mengambil data riwayat deteksi dari database server. Jika permintaan berhasil dan data riwayat ditemukan, data tersebut disimpan dalam variabel '\_history', berupa daftar yang berisi informasi mengenai kesimpulan kondisi ikan, tanggal deteksi, dan lokasi deteksi. *Source Code* tersebut dapat dilihat pada Gambar 1. 10.



**Gambar 1. 10.** *Source Code* Tampilan Riwayat Deteksi

## 11. Tampilan Report

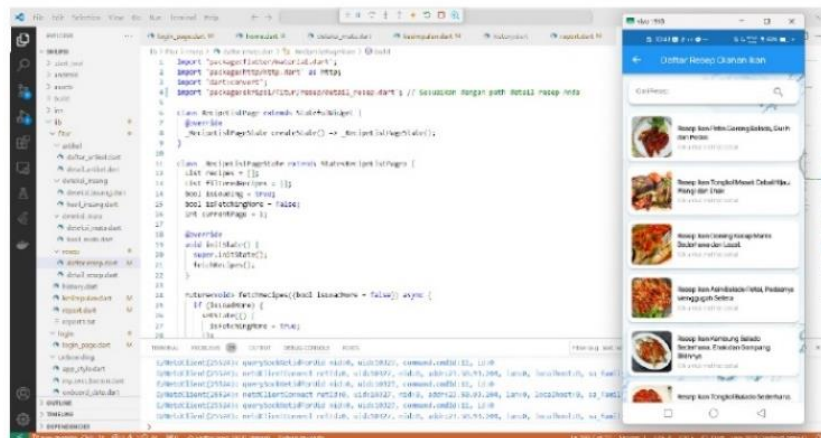
Saat halaman dibuka, aplikasi menampilkan indikator loading menggunakan variabel `_isLoading` untuk memberi tahu pengguna bahwa data sedang diambil. Data riwayat disimpan dalam variabel `_history`, yang diperoleh melalui permintaan HTTP ke server menggunakan token yang disimpan di local storage. Data yang diambil kemudian di filter berdasarkan lokasi menggunakan metode `_aggregateHistory`, dengan hasil yang disimpan dalam variabel `_filteredHistory`. *Source Code* tersebut dapat dilihat pada Gambar 1.10.



Gambar 1. 11. Source Code Tampilan Report

## 12. Tampilan Daftar Resep

Halaman ini menampilkan daftar resep olahan ikan yang diambil dari sebuah website. Terdapat sebuah kotak pencarian dalam card untuk pengguna mencari resep dan akan memfilter daftar resep dalam variabel 'filteredRecipes'. Variabel utama yang digunakan adalah 'recipes' (daftar semua resep), 'filteredRecipes' (daftar resep yang difilter), 'isLoading' (status pengambilan data), 'isFetchingMore' (status pengambilan data tambahan), dan 'currentPage' (halaman saat ini). Source Code tersebut dapat dilihat pada Gambar 1. 12.



Gambar 1. 12. Source Code Tampilan Daftar Resep

### 13. Tampilan Detail Resep

Halaman ini menampilkan informasi lengkap tentang resep yang dipilih, diambil dari server berdasarkan URL yang diberikan. Setelah data berhasil diambil, halaman ini akan menampilkan gambar resep ('recipeDetail['image\_url']'), judul ('recipeDetail['title']'), dan deskripsi ('recipeDetail['desc']"). Selanjutnya, terdapat daftar bahan-bahan ('recipeDetail['bahan']) yang ditampilkan sebagai teks dengan format bulat point, diikuti dengan langkah-langkah pembuatan ('recipeDetail['cara\_membuat']). Variabel utama yang digunakan adalah 'recipeDetail', sebuah map yang menyimpan detail resep termasuk gambar, judul, deskripsi, bahan-bahan, dan cara pembuatan. *Source Code* tersebut dapat dilihat pada Gambar 1. 13.

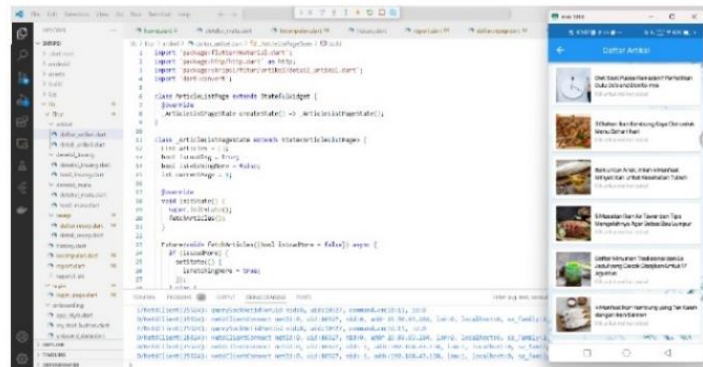


Gambar 1. 13. *Source Code* Tampilan Detail Resep

### 14. Tampilan Daftar Artikel

Halaman ini menampilkan daftar artikel yang diambil dari server. Setelah data berhasil diambil, halaman ini akan menampilkan daftar artikel berisi gambar, judul artikel, dan teks penjelasan. Variabel utama yang digunakan adalah 'articles', yang menyimpan daftar artikel yang diambil, 'isLoading', yang menunjukkan status pengambilan data awal, 'isFetchingMore', yang menunjukkan status pengambilan data tambahan saat menggulir, dan 'currentPage', yang melacak halaman saat ini. *Source Code* tersebut dapat dilihat pada Gambar 1. 14.

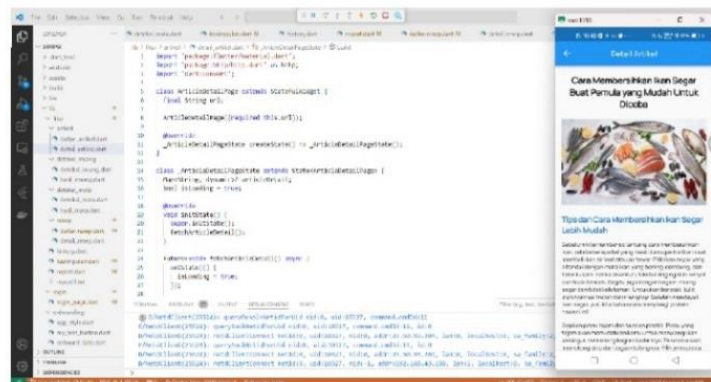




Gambar 1. 14. Source Code Tampilan Daftar Artikel

15. Tampilan Detail artikel

Halaman ini menampilkan informasi detail artikel yang diambil dari server berdasarkan URL yang diberikan. Setelah data berhasil diambil, halaman ini akan menampilkan judul artikel ('articleDetail['title']'), gambar ('articleDetail['image\_url']'), dan daftar sub-judul dengan deskripsinya ('articleDetail['sub\_title']'). Variabel utama yang digunakan adalah 'articleDetail', untuk menyimpan detail artikel termasuk judul, gambar, dan sub-judul dengan deskripsinya, serta 'isLoading', yang menunjukkan status pengambilan data. Source Code tersebut dapat dilihat pada Gambar 1. 15.



Gambar 1. 15. Source Code Tampilan Detail Artikel

## 2) Logic

### 1. Import Library

Bagian import pada kode Flask mencakup berbagai modul dan pustaka yang diperlukan untuk membangun aplikasi. Seperti 'random' yang digunakan untuk menghasilkan angka acak, sementara 'jwt' digunakan untuk mengelola JSON Web Tokens untuk autentikasi pengguna. Flask dan Flask-RESTX ('Flask', 'request', 'jsonify', 'Resource', 'Api', 'reqparse') memberikan kerangka kerja untuk membangun aplikasi dan API RESTful. 'SQLAlchemy' memfasilitasi interaksi dengan database menggunakan ORM, dan 'Flask-Mail' ('Mail', 'Message') memungkinkan pengiriman email dari aplikasi. 'werkzeug.security' menyediakan fungsi untuk hashing dan verifikasi password, sedangkan modul 'datetime' menangani operasi terkait tanggal dan waktu. Modul 'os' digunakan untuk interaksi dengan sistem operasi, terutama untuk membaca variabel lingkungan yang disimpan dalam file '.env' melalui 'dotenv'. 'requests' memungkinkan pengambilan data dari *website*, dan 'BeautifulSoup' digunakan untuk menguraikan HTML dan XML yang diambil, memungkinkan ekstraksi informasi yang dibutuhkan dari halaman web. *Source Code* tersebut dapat dilihat pada Gambar 2. 1.

```
import random
import jwt
from flask import Flask, request, jsonify
from flask_restx import Resource, Api, reqparse
from flask_sqlalchemy import SQLAlchemy
from flask_mail import Mail, Message
from werkzeug.security import generate_password_hash, check_password_hash
from datetime import datetime, timedelta, date
import os
import re
from dotenv import load_dotenv
import requests
from bs4 import BeautifulSoup
```

**Gambar 2. 1.** Import Library

## 2. Route Registrasi

Pada bagian ini, terdapat endpoint untuk registrasi pengguna baru yang diakses melalui URL `/register` dengan metode HTTP POST, dengan input JSON yang berisi email, nama, dan password pengguna. Pertama, kode ini memeriksa apakah email yang diberikan sudah digunakan oleh pengguna lain. Jika sudah digunakan, respons yang akan diterima yaitu pesan "Email ini sudah digunakan!" akan dikirim. Jika belum digunakan, objek `User` baru dibuat dengan email, nama, dan password yang di-hash untuk keamanan. Pengguna baru kemudian ditambahkan ke session database dan perubahan disimpan menggunakan `db.session.commit()`. Jika registrasi berhasil, respons yang akan diterima yaitu pesan "Berhasil!" dengan status HTTP 201 akan dikirim, menandakan bahwa pengguna baru telah berhasil didaftarkan dalam sistem. *Source Code* tersebut dapat dilihat pada Gambar 2.2.

```
@api.route('/register')
class Registration(Resource):
    @api.expect(parser4Reg)
    def post(self):
        args = parser4Reg.parse_args()
        email = args['email']
        name = args['name']
        password = args['password']

        user = db.session.execute(db.select(User).filter_by(email=email)).first()

        if user:
            return "Email ini sudah digunakan!"

        user = User()
        user.email = email
        user.name = name
        user.password = generate_password_hash(password)

        db.session.add(user)
        db.session.commit()

        return {'message': 'Successful!'}, 201
```

**Gambar 2. 2.** Route Registrasi

### 3. Route Verifikasi Email

Pada bagian ini, terdapat endpoint untuk verifikasi email pengguna yang diakses melalui URL '/otp-verif' dengan metode HTTP POST, dengan input JSON yang berisi email dan OTP. Pertama, parser parser4OTP dibuat untuk memarsing argumen email dan otp dari input JSON. Endpoint ini kemudian memeriksa apakah OTP yang diberikan oleh pengguna sesuai dengan OTP yang tersimpan di database untuk email tersebut. Jika OTP tidak sesuai, respons yang akan diterima adalah pesan error. Jika OTP sesuai, status verifikasi pengguna diperbarui di database dengan mengubah nilai `verified_user` menjadi 1. Perubahan ini kemudian disimpan menggunakan `db.session.commit()`. Jika verifikasi berhasil, respons yang akan diterima adalah pesan "Successful!" dengan status HTTP 201 yang menandakan bahwa verifikasi OTP berhasil dilakukan. *Source code* tersebut dapat dilihat pada Gambar 2. 3.

```
parser4OTP = reqparse.RequestParser()
parser4OTP.add_argument('email', type=str, help='Email', location='json', required=True)
parser4OTP.add_argument('otp', type=str, help='OTP', location='json', required=True)

@api.route('/otp-verif')
class OtpVerif(Resource):
    @api.expect(parser4OTP)
    def post(self):
        args = parser4OTP.parse_args()
        email = args['email']
        otpRequest = args['otp']

        user = User.query.filter_by(email=email).first()
        user_otp = user.otp

        if user_otp != otpRequest:
            return "PLEASE INSERT VALID OTP"

        user.verified_user = 1
        db.session.commit()
        return {'message': 'Successful!'}, 201
```

**Gambar 2. 3.** Route Verifikasi Email

### 4. Route *Login*

Pada bagian ini, terdapat endpoint untuk proses login pengguna yang diakses melalui URL '/login' dengan metode HTTP POST, dengan input JSON yang berisi email dan password pengguna. Pertama, kode ini memeriksa apakah email dan password sudah diisi. Jika tidak diisi, akan dikirimkan pesan kesalahan "Silahkan masukkan Email atau Password!" dengan status HTTP 400. Selanjutnya, kode ini memeriksa apakah pengguna dengan email yang

diberikan ada di database. Jika tidak ada, akan dikirimkan pesan kesalahan “Email atau Password Salah!” dengan status HTTP 400.

Jika pengguna ditemukan, password yang di-hash akan diperiksa. Jika password cocok, payload JWT akan dibuat dengan informasi pengguna seperti ID, email, audiens, issuer, waktu pembuatan, dan waktu kadaluarsa. Token ini kemudian di-encode dengan ‘SECRET\_KEY’ dan dikirim sebagai bagian dari respons sukses dengan status HTTP 200, dan akan menampilkan pesan “Berhasil Login!” dan nama pengguna. Jika password tidak cocok, akan dikirimkan pesan kesalahan “Ada kesalahan Email atau Password!” dengan status HTTP 400. *Source Code* tersebut dapat dilihat pada Gambar 2. 4.

```
@api.route('/login')
class LogIn(Resource):
    @api.expect(parser4LogIn)
    def post(self):
        args = parser4LogIn.parse_args()
        email = args['email']
        password = args['password']

        if not email or not password:
            return {
                'message': 'Silahkan masukkan Email atau Password!'
            }, 400

        user = db.session.execute(db.select(User).filter_by(email=email)).first()

        if not user:
            return {
                'message': 'Email atau Password Salah!'
            }, 400
        else:
            user = user[0]
```



```

if check_password_hash(user.password, password):
    payload = {
        'user_id': user.id,
        'email': user.email,
        'aud': AUDIENCE_MOBILE,
        'iss': ISSUER,
        'iat': datetime.utcnow(),
        'exp': datetime.utcnow() + timedelta(hours=2)
    }
    token = jwt.encode(payload, SECRET_KEY)
    return {
        'message': 'Berhasil Login!',
        'token': token,
        'name': user.name
    }, 200
else:
    return {
        'message': 'Wrong email or password!'
    }, 400

```

**Gambar 2. 4.** Route *Login*

#### 5. Route Edit *Username*

Pada bagian ini, terdapat endpoint untuk mengedit informasi pengguna yang diakses melalui URL `/edit-user` dengan metode HTTP PUT, dengan input JSON yang berisi email dan nama baru pengguna, serta header `Authorization` yang berisi token Bearer. Pertama, kode ini mengambil argumen yang dikirimkan dan mengekstrak token Bearer dari header `Authorization`. Kemudian, token ini di-decode untuk mendapatkan informasi pengguna. Kode ini mencari pengguna berdasarkan email yang terdapat dalam token dan memperbarui email dan nama pengguna dengan nilai baru yang diberikan, serta menambahkan timestamp saat ini sebagai `updatedAt`.

Jika semua proses berhasil, perubahan disimpan dalam database dan respons sukses dengan pesan `Update User Sukses` dan status HTTP 200 dikirimkan. Jika terjadi kesalahan, seperti token yang tidak valid, akan dikirimkan pesan kesalahan `Token Tidak valid, Silahkan Login Terlebih Dahulu!` dengan status HTTP 401. *Source Code* tersebut dapat dilihat pada Gambar 2. 5.

```

@api.route('/edit-user')
class EditUser(Resource):
    @api.expect(editParser)
    def put(self):
        args = editParser.parse_args()
        bearerAuth = args['Authorization']
        email = args['email']
        name = args['name']
        datenow = datetime.today().strftime('%Y-%m-%d %H:%M:%S')

        try:
            jwtToken = bearerAuth[7:]
            token = decodetoken(jwtToken)
            user = User.query.filter_by(email=token.get('email')).first()
            user.email = email
            user.name = name
            user.updatedAt = datenow
            db.session.commit()
        except:
            return {
                'message': 'Token Tidak valid, Silahkan Login Terlebih Dahulu!'
            }, 401

        return {'message': 'Update User Sukses'}, 200

```

**Gambar 2. 5.** Route Edit Username

## 6. Route Edit Password

Pada bagian ini, terdapat endpoint untuk mengubah password pengguna yang diakses melalui URL '/edit-password' dengan metode HTTP PUT, dengan input JSON yang berisi password saat ini ('current\_password'), password baru ('new\_password'), serta header 'Authorization' yang berisi token Bearer. Pertama, kode ini mengambil argumen yang dikirimkan dan mengekstrak token Bearer dari header 'Authorization'. Token ini kemudian di-decode untuk mendapatkan informasi pengguna. Kode ini mencari pengguna berdasarkan 'user\_id' yang terdapat dalam token dan memverifikasi apakah password saat ini sesuai dengan hash yang disimpan dalam database. Jika verifikasi berhasil, password baru di-hash dan diperbarui dalam database.

Jika password saat ini salah, akan dikirimkan pesan kesalahan "Password Lama Salah" dengan status HTTP 400. Jika terjadi kesalahan lain, seperti token yang tidak valid, akan dikirimkan pesan kesalahan "Token Tidak valid! Silahkan, Sign in!" dengan status HTTP 401. Jika semua proses berhasil,

respons sukses dengan pesan “Password Berhasil Diubah” dan status HTTP 200 dikirimkan. *Source Code* tersebut dapat dilihat pada Gambar 2. 6.

```
@api.route('/edit-password')
class Password(Resource):
    @api.expect(parser4Bearer, editPasswordParser)
    def put(self):
        args = editPasswordParser.parse_args()
        argss = parser4Bearer.parse_args()
        bearerAuth = argss['Authorization']
        cu_password = args['current_password']
        newpassword = args['new_password']

        try:
            jwtToken = bearerAuth[7:]
            token = decodetoken(jwtToken)
            user = User.query.filter_by(id=token.get('user_id')).first()

            if check_password_hash(user.password, cu_password):
                user.password = generate_password_hash(newpassword)
                db.session.commit()
            else:
                return {'message': 'Password Lama Salah'}, 400
        except:
            return {
                'message': 'Token Tidak valid! Silahkan, Sign in!'
            }, 401

        return {'message': 'Password Berhasil Diubah'}, 200
```

**Gambar 2. 6.** Route Edit Password

#### 7. Route Riwayat Deteksi

Pada bagian ini, terdapat dua endpoint Flask, yaitu ‘/save-history’ dan ‘/get-history’, yang berfungsi untuk menyimpan dan mengambil riwayat aktivitas pengguna. Endpoint ‘/save-history’ menggunakan metode HTTP POST untuk menyimpan data riwayat baru ke dalam database berdasarkan token Bearer yang disertakan dalam header ‘Authorization’. Setelah mendekode token tersebut, kode memperoleh ‘user\_id’ dari pengguna yang terautentikasi dan menyimpan detail riwayat seperti kesimpulan (‘conclusion’) dan lokasi (‘location’). Jika operasi penyimpanan berhasil, endpoint mengirimkan respons dengan pesan sukses dan status HTTP 201.



Sementara itu, endpoint `‘/get-history’` menggunakan metode HTTP GET untuk mengambil semua entri riwayat yang terkait dengan `‘user_id’` yang dikirimkan melalui token Bearer. Setelah mendekode token dan mengambil `‘user_id’`, kode menjalankan query ke database untuk mengambil semua riwayat pengguna. Setiap entri riwayat diambil dalam format tanggal (`‘date’`), kesimpulan (`‘conclusion’`), dan lokasi (`‘location’`), yang kemudian disusun dalam bentuk daftar (`‘history_list’`). Jika berhasil, endpoint mengirimkan respons dengan daftar riwayat dalam format JSON dan status HTTP 200. Namun, jika terjadi kesalahan dalam pengambilan data, endpoint memberikan respons dengan pesan kesalahan yang mencakup detail kesalahan yang terjadi dan status HTTP 400. *Source Code* tersebut dapat dilihat pada Gambar 2. 7.

```
@api.route('/save-history')
class SaveHistory(Resource):
    @api.expect(parser4Bearer, history_parser)
    def post(self):
        args = history_parser.parse_args()
        auth_args = parser4Bearer.parse_args()
        bearerAuth = auth_args['Authorization']
        jwtToken = bearerAuth[7:]

        try:
            token = decodetoken(jwtToken)
            user_id = token.get('user_id')
            conclusion = args['conclusion']
            location = args['location']

            new_history = History(user_id=user_id, conclusion=conclusion, location=location)
            db.session.add(new_history)
            db.session.commit()

            return {'message': 'History saved successfully!'}, 201
        except Exception as e:
            db.session.rollback()
            print(f"Error: {str(e)}")
            return {'message': 'Error saving history: ' + str(e)}, 400

@api.route('/get-history')
class GetHistory(Resource):
    @api.expect(parser4Bearer)
    def get(self):
        auth_args = parser4Bearer.parse_args()
        bearerAuth = auth_args['Authorization']
        jwtToken = bearerAuth[7:]

        try:
            token = decodetoken(jwtToken)
            user_id = token.get('user_id')
```

```

histories = History.query.filter_by(user_id=user_id).all()
history_list = []
for history in histories:
    history_data = {
        'date': history.date.strftime('%d-%m-%Y'),
        'conclusion': history.conclusion,
        'location': history.location
    }
    history_list.append(history_data)

return {'history': history_list}, 200
except Exception as e:
    return {'message': 'Error fetching history: ' + str(e)}, 400

```

**Gambar 2. 7.** Route Riwayat Deteksi

## 8. Route Report

Pada bagian ini, terdapat endpoint `/get-all-history` menggunakan metode HTTP GET untuk mengambil seluruh riwayat dari tabel `'History'`. Dalam metode ini, semua entri riwayat diambil dari database menggunakan `'History.query.all()'`. Setiap entri riwayat disusun ke dalam dictionary yang berisi tanggal riwayat, kesimpulan, lokasi, dan email pengguna yang terkait. Semua entri riwayat kemudian disusun dalam bentuk list dan dikembalikan sebagai respons JSON dengan status HTTP 200. Jika ada kesalahan selama proses pengambilan atau pengolahan data, endpoint akan mengembalikan pesan kesalahan dengan detail kesalahan yang terjadi dan status HTTP 400. *Source Code* tersebut dapat dilihat pada Gambar 2. 8.

```

@api.route('/get-all-history')
class GetAllHistory(Resource):
    def get(self):
        try:
            histories = History.query.all()
            all_history_list = []
            for history in histories:
                user = User.query.filter_by(id=history.user_id).first()
                history_data = {
                    'date': history.date.strftime('%d-%m-%Y'),
                    'conclusion': history.conclusion,
                    'location': history.location,
                    'user': user.email if user else 'Unknown'
                }
                all_history_list.append(history_data)

            return {'history': all_history_list}, 200
        except Exception as e:
            return {'message': 'Error fetching history: ' + str(e)}, 400

```

**Gambar 2. 8.** Route Report

## 9. Route Scraping Website Resep

Pada bagian ini, terdapat endpoint `/resep` yang menggunakan metode HTTP GET untuk mengambil daftar resep dari situs `https://www.masakapahariini.com`. Pertama, endpoint membangun URL berdasarkan nilai `page` yang diberikan dalam permintaan. Kemudian, menggunakan pustaka `requests` dan `BeautifulSoup`, kode mengirimkan permintaan GET ke URL yang telah disiapkan dengan menyertakan header `User-Agent` untuk mensimulasikan permintaan dari browser. Jika permintaan berhasil akan muncul status code 200, kode melakukan parsing halaman menggunakan `BeautifulSoup` untuk mengekstrak tautan, nilai, dan gambar dari setiap entri resep yang ditemukan. Setiap entri resep yang berhasil diekstrak diambil dari elemen HTML dengan kelas `_recipe-card`. Untuk setiap entri, tautan ke resep, nilai atau judul resep, dan URL gambar diekstrak dan disimpan dalam bentuk daftar objek Python. Jika ada kesalahan dalam proses pengambilan data, endpoint akan mengembalikan pesan kesalahan yang mencakup detail kesalahan yang terjadi, bersama dengan status HTTP 400. *Source Code* tersebut dapat dilihat pada Gambar 2. 9.

```
@api.route('/resep')
class GetResep(Resource):
    def get(self):
        try:
            page = request.args.get('page', 1)
            url = f'https://www.masakapahariini.com/page/{page}?s=ikan'

            headers = {
                'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36'
            }

            response = requests.get(url, headers=headers)
            if response.status_code != 200:
                return {'message': 'error fetching recipe list'}, response.status_code
            soup = BeautifulSoup(response.text, 'html.parser')

            # Mengambil tautan dan nilai dari setiap resep
            recipe_links_and_values = []
            for recipe in soup.find_all('div', class_='recipe-card'):
                link_element = recipe.find('a', class_='stretched-link')
                image_element = recipe.find('img')
                image = image_element['src'] if image_element else None
                link = link_element.get('href')
                value = link_element.text.strip()
                recipe_links_and_values.append({
                    'link': link,
                    'value': value,
                    'image': image
                })
            return {'resep': recipe_links_and_values}, 200
        except Exception as e:
            return {'message': 'Error fetching resep: ' + str(e)}, 400
```

**Gambar 2. 9.** Route Scraping Website Resep

## 10. Route Scraping Detail Resep

Pada bagian ini, terdapat endpoint `/resep-detail` menggunakan metode HTTP GET untuk mengambil detail dari sebuah resep berdasarkan URL yang diberikan sebagai parameter query. Pertama, endpoint ini mengambil URL resep dari parameter query dan mengirimkan permintaan GET ke URL tersebut menggunakan pustaka `requests`, termasuk header `User-Agent` untuk mensimulasikan permintaan dari browser. Jika permintaan berhasil akan muncul pesan status code 200, halaman web resep diambil menggunakan `BeautifulSoup`.

Bahan-bahan resep diambil dari elemen HTML dengan kelas `_recipe-ingredients`. Langkah-langkah pembuatan resep diambil dari elemen HTML dengan kelas `_recipe-steps`. Semua informasi ini kemudian disusun dalam sebuah dictionary `recipe_detail`, yang berisi judul, URL gambar, deskripsi, bahan-bahan, dan langkah-langkah pembuatan resep. *Source Code* tersebut dapat dilihat pada Gambar 2. 10.

```
@api.route('/resep-detail')
class GetResepDetail(Resource):
    def get(self):
        try:
            # Mengambil URL resep dari parameter query
            url = request.args.get('url')
            if not url:
                return ('message': 'URL parameter is required'), 400

            headers = {
                'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.'
            }

            response = requests.get(url, headers=headers)
            if response.status_code != 200:
                return ('message': 'Error fetching recipe detail'), response.status_code

            soup = BeautifulSoup(response.text, 'html.parser')

            for portions_div in soup.find_all('div', class_='portions'):
                portions_div.decompose()
            # Mengambil detail resep
            recipe_detail = {}

            # Contoh scraping untuk mendapatkan judul dan gambar
            title_element = soup.find('h1')
            title = title_element.text.strip() if title_element else 'no title found'
            desc_element = soup.find('div', class_='excerpt text-center text-md-start')
            desc = desc_element.text.strip() if desc_element else 'no desc found'
            image_element = soup.find('img', class_='image')
            image_url = image_element['src'] if image_element else None

            ingredients = []
            ingredients_container = soup.find('div', class_='recipe-ingredients')
            if ingredients_container:
                for item_element in ingredients_container.find_all('div', class_='d-flex'):
                    item = re.sub(r'\s+', ' ', item_element.text.strip())
                    ingredients.append({
                        'item': item
                    })

            instructions = []
            instructions_container = soup.find('div', class_='recipe-steps')
            if instructions_container:
                for step_element in instructions_container.find_all('div', class_='step'):
                    step = re.sub(r'\s+', ' ', step_element.text.strip())
                    instructions.append({
                        'step': step
                    })
        }
```

```

# Menyimpan detail resep dalam dictionary
recipe_detail = {
    'title': title,
    'image_url': image_url,
    'desc': desc,
    'bahan': ingredients,
    'cara_membuat': instructions
}

return ('recipe_detail': recipe_detail), 200
except Exception as e:
    return {'message': 'Error fetching recipes: ' + str(e)}, 400

```

**Gambar 2. 10.** Route Scraping Detail Resep

## 11. Route Scraping Artikel

Pada bagian ini, terdapat endpoint '/artikel' menggunakan metode HTTP GET untuk mengambil daftar artikel. Endpoint ini mengirimkan permintaan GET ke URL yang dibentuk dengan menambahkan parameter halaman ke URL dasar situs. Endpoint ini kemudian mencari elemen artikel pada halaman web tersebut, yang diwakili oleh elemen HTML dengan kelas '\_article-card'. Informasi artikel yang diekstraksi kemudian disusun dalam bentuk list of dictionaries, masing-masing berisi 'link', 'value', dan 'image'. *Source Code* tersebut dapat dilihat pada Gambar 2. 11.

```

@api.route('/artikel')
class GetArtikel(Resource):
    def get(self):
        try:
            page = request.args.get('page', 1)
            url = f'https://www.masakapaharini.com/page/{page}?s=ikan'

            headers = {
                'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4012.101 Safari/537.36'
            }

            response = requests.get(url, headers=headers)
            if response.status_code != 200:
                return {'message': 'error fetching article list'}, response.status_code

            soup = BeautifulSoup(response.text, 'html.parser')

            artikel_links_and_values = []
            for artikel in soup.find_all('div', class_='article-card'):
                link_element = artikel.find('a', class_='stretched-link')
                img_element = artikel.find('img')
                link = link_element['href'].strip() if link_element else None # menggunakan strip() di sini
                value = link_element.text.strip() # menggunakan strip() di sini juga
                artikel_links_and_values.append({
                    'link': link,
                    'value': value,
                    'image': img
                })

            # Tambahkan logging untuk respons
            print('artikel': artikel_links_and_values)
            return ('artikel': artikel_links_and_values), 200
        except Exception as e:
            print('Error fetching artikel: {str(e)}')
            return {'message': 'Error fetching artikel: ' + str(e)}, 400

```

**Gambar 2. 11.** Route Scraping Artikel



## 12. Route *Scraping* Detail Artikel

Pada bagian ini, terdapat endpoint `/artikel-detail` menggunakan metode HTTP GET untuk mengambil detail sebuah berdasarkan URL artikel yang diberikan sebagai parameter query `url`. Detail artikel yang diambil meliputi judul artikel, gambar utama, dan subjudul dengan deskripsinya. Judul artikel diambil dari elemen HTML `<h1>`, dan URL gambar diambil dari elemen `<img>` dengan kelas `image`. Subjudul dan deskripsi artikel diambil dari elemen HTML `<h2>` dan paragraf-paragraf berikutnya sampai elemen `<h2>` berikutnya ditemukan. Setiap subjudul dan deskripsinya disusun dalam bentuk list of dictionaries yang berisi `sub` (subjudul) dan `desc` (deskripsi). *Source Code* tersebut dapat dilihat pada Gambar 2. 12.

```
@api.route('/artikel-detail')
class GetArtikelDetail(Resource):
    def get(self):
        try:
            # Mengambil URL resep dari parameter query
            url = request.args.get('url')
            if not url:
                return {'message': 'URL parameter is required'}, 400

            headers = {
                'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36'
            }

            response = requests.get(url, headers=headers)
            if response.status_code != 200:
                return {'message': 'Error fetching artikel detail'}, response.status_code

            soup = BeautifulSoup(response.text, 'html.parser')

            # Mengambil detail resep
            artikel_detail = {}

            # Contoh scraping untuk mendapatkan judul dan gambar
            title_element = soup.find('h1')
            title = title_element.text.strip() if title_element else 'No title found'
            image_element = soup.find('img', class_='image')
            image_url = image_element['src'] if image_element else None

            sub_title = []
            sub_title_container = soup.find("div", class_='rich-content')
            if sub_title_container:
                for item_element in sub_title_container.find_all("h2"):
                    item = item_element.text.strip()
                    steps = []

                    # Find all sibling <p> elements until the next <h2> is encountered
                    for sibling in item_element.find_next_siblings():
                        if sibling.name == 'h2':
                            break
                        if sibling.name == 'p':
                            steps.append(sibling.text.strip())

            # Menyimpan detail resep dalam dictionary
            artikel_detail = {
                'title': title,
                'image_url': image_url,
                'sub_title': sub_title,
            }

            return {'artikel_detail': artikel_detail}, 200
        except Exception as e:
            return {'message': 'Error fetching artikel detail: ' + str(e)}, 400
```

**Gambar 2. 12.** Route *Scraping* Detail Artikel

## Lampiran 6. Sertifikat HKI yang terbit

REPUBLIC INDONESIA KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA	
<b>SURAT PENCATATAN CIPTAAN</b>	
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	: EC00202459834, 4 Juli 2024
<b>Pencipta</b>	
Nama	: Putri Zulfiaturrizki, Dwi Intan Afidah, S.T., M.Kom. dkk
Alamat	: Jalan Angrek 1, RT 04/ RW 03, Desa Karangdawa, Kecamatan Margasari, Kabupaten Tegal, Propinsi Jawa Tengah 52463, Margasari, Tegal, Jawa Tengah, 52463
Kewarganegaraan	: Indonesia
<b>Pemegang Hak Cipta</b>	
Nama	: Pusat Penelitian dan Pengabdian Masyarakat (P3M) Politeknik Harapan Bersama
Alamat	: Jalan Mataram No. 9, Pesurungan Lor, Kecamatan Margadana 52142, Margadana, Tegal, Jawa Tengah 52142
Kewarganegaraan	: Indonesia
Jenis Ciptaan	: Program Komputer
Judul Ciptaan	: Aplikasi Mobile Untuk Deteksi Kualitas Ikan Segar Berdasarkan Kondisi Mata Dan Insang
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	: 000635192
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	Putri Zulfiaturrizki	Jalan Angrek 1, RT 04/ RW 03, Desa Karangdawa, Kecamatan Margasari, Kabupaten Tegal, Propinsi Jawa Tengah 52463, Margasari, Tegal
2	Dwi Intan Afidah, S.T., M.Kom.	Desa Grinting RT 003/RW 001, Kecamatan Bulakamba, Kabupaten Brebes, Propinsi Jawa Tengah 52253, Bulakamba, Brebes
3	Rosid Mustofa, S.Kom., M.Kom.	Desa Kedungbanteng, RT 25/RW 12, Kecamatan Kedung Banteng, Kabupaten Tegal, Propinsi Jawa Tengah 52472, Kedung Banteng, Tegal





## Lampiran 7. Lembar Bimbingan



**SARJANA TERAPAN TEKNIK INFORMATIKA  
POLITEKNIK HARAPAN BERSAMA**

### LEMBAR BIMBINGAN SKRIPSI

Nama : Putri Zulfiaturrizki  
Nim : 20090138  
No. Ponsel : 082314930434  
Judul TA : Aplikasi Mobile untuk Deteksi Kualitas Ikan Segar Berdasarkan Kondisi Mata dan Insang menggunakan YOLOv5  
Dosen Pembimbing I : Dwi Intan Afidah, S.T., M.Kom.

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
1.	19 Maret 2024	Konsep produk skripsi	Pastikan bisnis proses pada saat scan mata ikan, scan insang ikan, dan hasil klasifikasi ikannya	
2.	04 April 2024	Model Deteksi Objek	Pastikan akurasi modelnya sudah mencapai 90%	
3.	25 April 2024	Implementasi Model	Dilanjutkan untuk implementasi model ke aplikasi.	
4.	17 Mei 2024	Produk	Tambahkan history dan juga report jumlah ikan segar dan tidak segar per wilayah.	
5.	31 Mei 2024	Produk	<ul style="list-style-type: none"><li>- Dilanjutkan untuk menambahkan report berdasarkan wilayah.</li><li>- Tambahkan detail resep dan artikel.</li></ul>	
6.	28 Juni 2024	Pesyaratan Hak Cipta	Lengkapi persyaratan Hak Cipta	



**SARJANA TERAPAN TEKNIK INFORMATIKA  
POLITEKNIK HARAPAN BERSAMA**

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
7.	05 Juli 2024	Produk	Testing Aplikasi	
8.	12 Juli 2024	Laporan	- Bab 1. Perbaiki tujuan, manfaat dan tinjauan pustaka. - Bab 2. Lengkapi Saran dan Kesimpulan.	
9.	15 Juli 2024	Laporan	Tambahkan Lampiran	
10.	16 Juli 2024	Laporan	ACC	

Tegal, 16 Juli 2024  
Dosen Pembimbing I

Dwi Intan Afidah, S.T., M.Kom.  
NIPY. 11.020.470



**SARJANA TERAPAN TEKNIK INFORMATIKA  
POLITEKNIK HARAPAN BERSAMA**

**LEMBAR BIMBINGAN SKRIPSI**

Nama : Putri Zulfiaturrizki  
Nim : 20090138  
No. Ponsel : 082314930434  
Judul TA : Aplikasi Mobile untuk Deteksi Kualitas Ikan  
Berdasarkan Kondisi Mata dan Insang menggunakan  
YOLOv5  
Dosen Pembimbing II : Rosid Mustofa, S.Kom, M.Kom

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
1	21/3/21	konsep	Tuliskan biar lebih jernih	Az.
2	16/5/21	produk	model di bahas kembali model tising	Az.
3	22/5/21	produk	History di selesaikan Papan home ke database.	Az.
4	12/6/21	Aplikasi	udah ok lagi di jaga di test - test Lagi yah	Az.



SARJANA TERAPAN TEKNIK INFORMATIKA  
POLITEKNIK HARAPAN BERSAMA

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing
5	27/21 6	Tulisan Bab 1-2	cek lagi typo nya lagi yaah bab 3	<i>[Signature]</i>
6	29/21 6	Tulisan	udaah ok	<i>[Signature]</i>
7	3/21 7	Praker.	diTS - ya lagi yah	<i>[Signature]</i>
8	16/21 7	Praker + Tulisan	Acc mag- sidang.	<i>[Signature]</i>



**SARJANA TERAPAN TEKNIK INFORMATIKA  
POLITEKNIK HARAPAN BERSAMA**

No	Tanggal	Pemeriksaan	Perbaikan yang perlu dilakukan	Paraf Pembimbing

Tegal, 16 Juli 2024  
Dosen Pembimbing II

Rosid Mustofa, S.Kom, M.Kom.