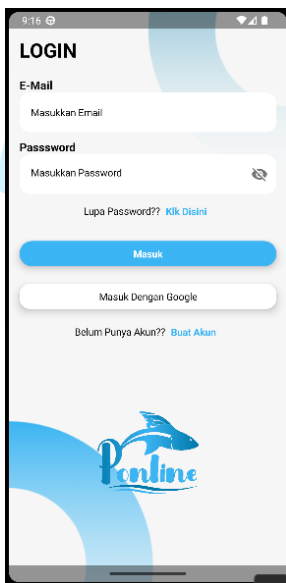


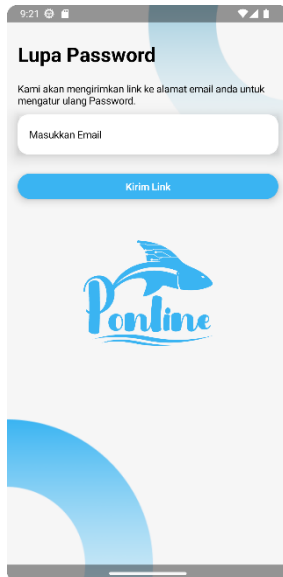
## Penjelasan dari hasil program client sesi 1



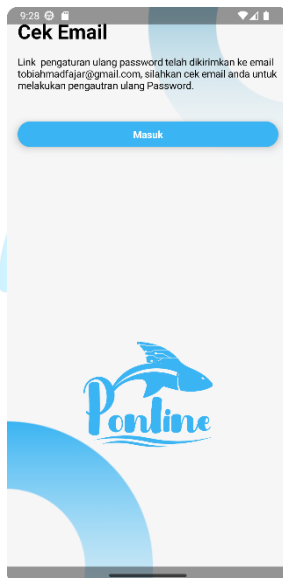
Splashscreen Untuk Frontend dengan platform mobile Menggunakan Framework Jetpack Compose Serta Kotlin.



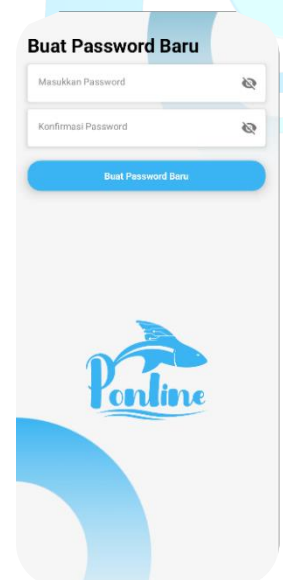
Setelah Halaman SplashScreen Jika User Belum Login Maka Akan Diarahkan ke Page login, diisini Terdapat Form Email dan Form Password, Serta terdapat dua text button dan dua button dimana text button yang pertama untuk mengarahkan ke halaman lupa password text button kedua untuk mengarahkan user ke halaman register, Serta button terdapat button login dan login with google



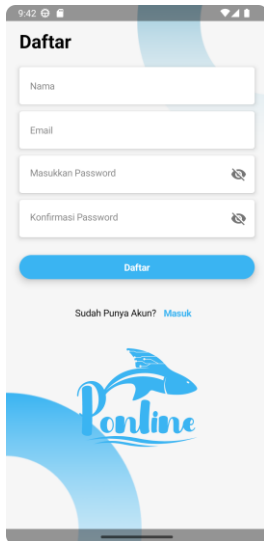
Merupakan Halaman Forgot Password, dimana nantinya user akan diminta memasukkan alamat email yang sebelumnya telah didaftarkan, agar mendapatkan url untuk reset password, setelah user mengklik tombol kirim link maka akan diarahkan ke page info yang nantinya user dapat kembali ke halaman login.



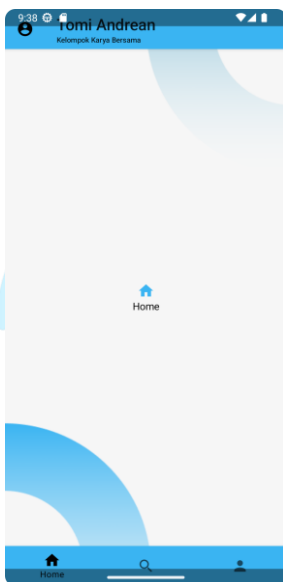
Setelah user memasukkan alamat email dan mengklik kirim link maka akan diarahkan ke halaman info ini yang nantinya user diarahkan untuk cek email serta kembali ke halaman login.



Setelah User Mengklik URL yang telah dikirimkan ke alamat emai, user akan diarahkan ke halaman ini untuk membuat password baru, yang kemudian akan diarahkan ke halaman awal login,



Jika User mengklik tombol Register maka akan diarahkan ke halaman daftar, disini terdapat beberapa form yang harus diisi seperti nama, email, dan password



Halaman ini yang nantinya akan dijadikan dashboard, setelah user sukses untuk login nantinya akan terdapat 3 halaman pada halaman dashboard yaitu home, search, serta management akun

## Penjelasan dari hasil pengujian program server sesi 1

Dalam pengujian untuk hasil konsumsi api server sesi 1 ini adalah mengenai autentikasi. Autentikasi yang di gunakan dalam program ada adalah http basic autentikasi dan OAuth2. http basic autentikasi di gunakan untuk user yang login atau mendaftar secara lokal sedangkan OAuth2 adalah untuk user yang login/mendaftar menggunakan provider pihak ketiga seperti yang telah dikerjakan adalah google.

Untuk pengujiannya menggunakan aplikasi insomnia sebagai REST API Client. Untuk import dokumen insomnia dapat di download pada [link Google Drive](#) ini.

1. Mendaftar pengguna baru dengan email dan password.

Metode : POST

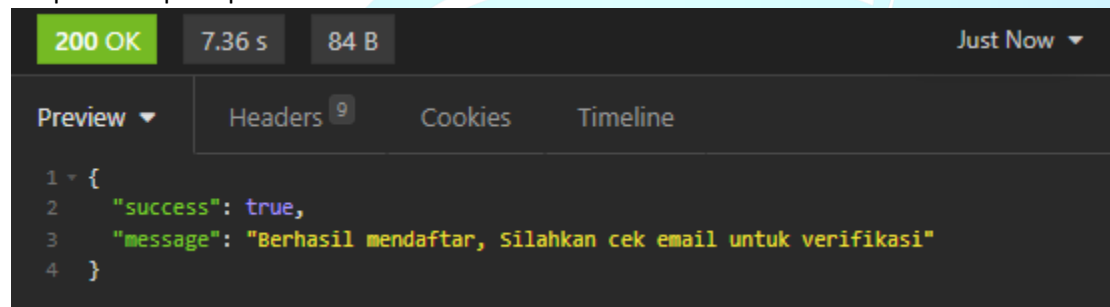
URLs endpoint : <http://localhost:8080/auth/signup>

Autentikasi : none

Request Body :

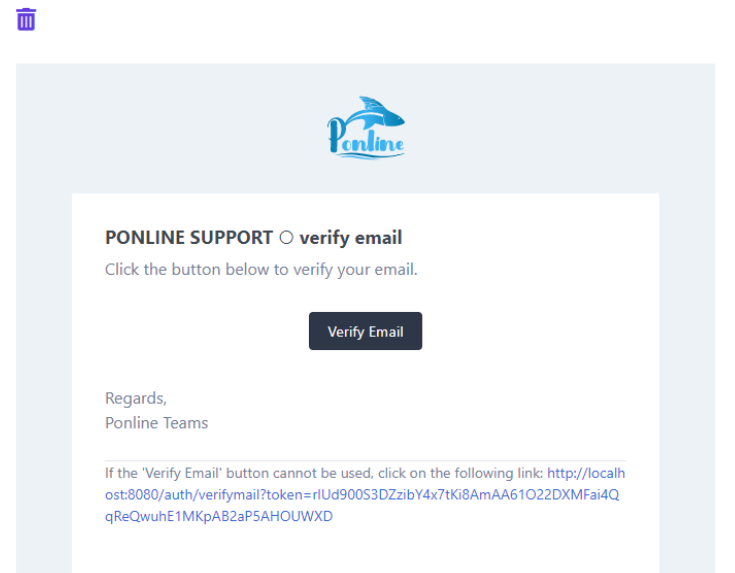
```
{
  "name": "PONLINE USER 1",
  "email": "pon_user1@quickat.work",
  "password": "123456"
}
```

Response : ApiResponse



Email Verifikasi :

PONLINE SUPPORT ○ verify email  
PONLINE SUPPORT - fredihermawan1211@gmail.com 2022-12-11 04:28 PM



Database : status akun tidak aktif

Token verifikasi email

ofvdobanjli	fbfewjerbe	ifomfluvud	kwbrfpgtm	ggxahwykfd	jltxnfodmw	osoumbbxli	oebwazzvdi	urwagexwxut	xvfeumirr	typhbwagm
1 pon_user1@quickat.work	1 pon_user1@quickat.work	1	NULL	NULL	NULL	NULL	NULL	ROLE_USER	NULL	NULL

## 2. Memverifikasi Email

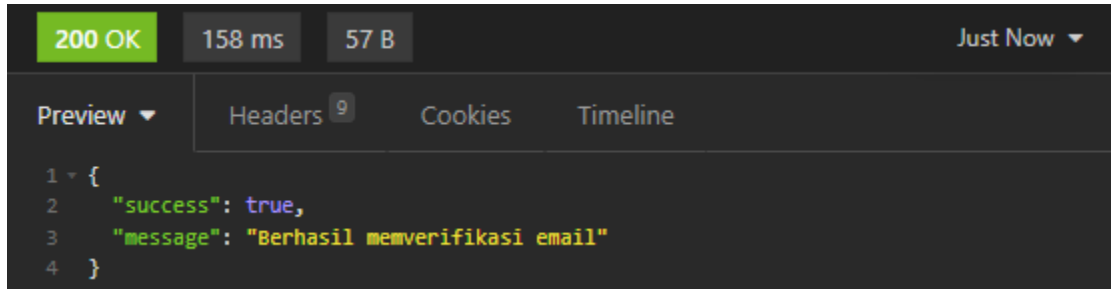
Metode: GET

URLs Endpoint : <http://localhost:8080/auth/verifymail?token={tokenVerifyEmail}>

Autentikasi : none

Request Body : none

Response : ApiResponse



Database : status akun aktif

Token email dihapus setelah verifikasi

ofvdobanjli	fbfewjerbe	ifomfluvud	kwbrfpgtm	ggxahwykfd	jltxnfodmw	osoumbbxli	oebwazzvdi	urwagexwxut	xvfeumirr	typhbwagm
1 pon_user1@quickat.work	1 pon_user1@quickat.work	1	NULL	NULL	NULL	NULL	NULL	ROLE_USER	NULL	NULL

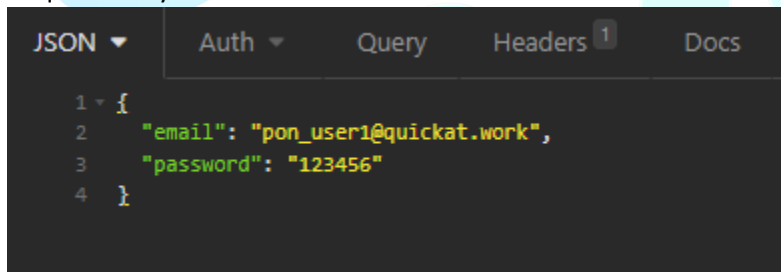
## 3. Login menggunakan email dan password

Metode : POST

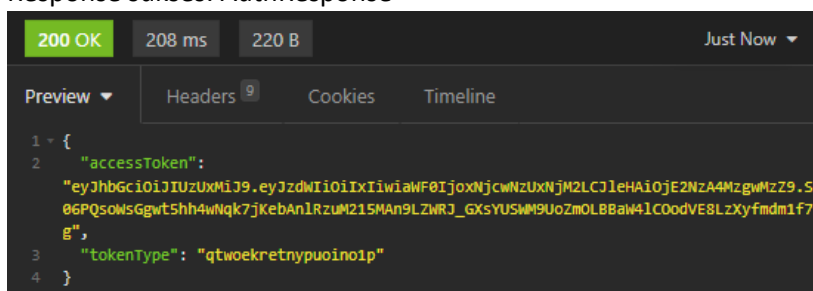
URLs Endpoint : <http://localhost:8080/auth/login>

Autentikasi : none

Request Body :



Response sukses: AuthResponse



Response email / password salah : BadRequestException

```

POST http://localhost:8080/auth/login
{
  "email": "pon_user1@quickat.work",
  "password": "1234"
}
{
  "timestamp": "2022-12-11T09:56:28.588+0000",
  "status": 401,
  "error": "Unauthorized",
  "message": "Bad credentials",
  "path": "/auth/login"
}

```

#### 4. Permintaan lupa atau ganti password pengguna

Metode : GET

URL Endpoint : <http://localhost:8080/auth/forgotpassword?email={email yang akan direset}>

Autentikasi : none

Request Body : none

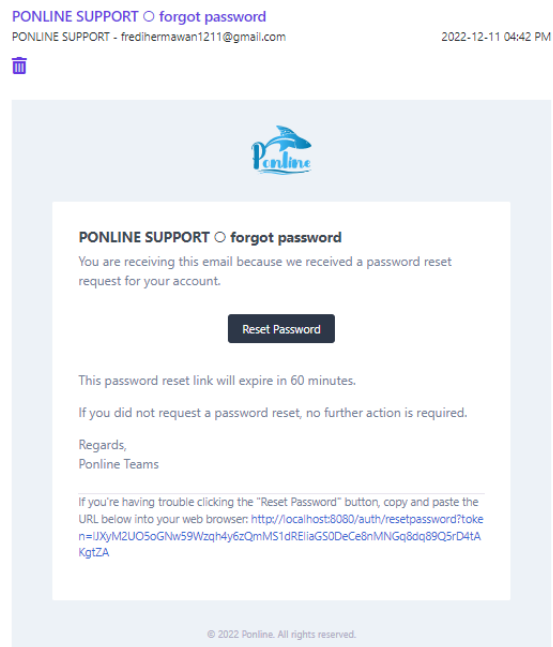
Response : ApiResponse

```

200 OK 4.76 s 75 B
{
  "success": true,
  "message": "Silahkan cek email untuk mereset password anda"
}

```

#### Email Reset Password :



#### Database :

#### Token Reset Password

ofvdsbnjl	tbfxjjerbe	ifomfluvud	knbrfpglm	ggxafwyktd	jtbxhfdmvr	oouombbxll	oebwazzvdl	uwxegxxuud	xvfeumirr	lysyhbwgqm
pon_user1@quickat.work	1	NULL	NULL	PONLINE USER 1	\$2a\$10\$R8XyupJuiM8t3QwIOG6JHD4n1C8RDY\$ymh...	local	NULL	ROLE_USER	NULL	UXyM2UOSoGNw59Wzqh4y6zQmIMS1dREIaGS0DeCe8nMNGq8dQ89Q5rD4TAKgZa

#### 5. Mengganti password

Author : Fredy Hermawan as BackEnd Developer; Tobi Ahmad Fajar as FrontEnd Developer; Tomi Andreansyah as FrontEnd Developer

Metode : POST

URLs Endpoint : [http://localhost:8080/auth/resetpassword?token={token\\_reset\\_password}](http://localhost:8080/auth/resetpassword?token={token_reset_password})

Autentikasi : none

Request Body :

```
JSON ▾ Auth ▾ Query Headers 1 Docs
1 {
2   "password": "123"
3 }
```

Response sukses : ApiResponse

The screenshot shows the Chrome DevTools Network tab with a single request selected. The status bar at the top indicates a 200 OK response, a response time of 117 ms, and a response size of 58 B. The 'Preview' pane is active, displaying the JSON response body: 

```
{  "success": true,  "message": "Berhasil memperbarui password"}
```

Response salah token : ApiResponse

A screenshot of a REST client interface. The top status bar shows a POST request to 'lw59Wzqh4y6zQmMS1dREliAGS0DeCe8nMNGq8dQ89Q5rD4tAKgtZA' with a status of '404 Not Found', a response time of '8.79 ms', and a body size of '51 B'. Below the status bar, the 'JSON' tab is selected, showing the response body: 

```
1 {
2   "password": "123"
3 }
```

 On the right, the 'Preview' tab is selected, showing the response body: 

```
1 {
2   "success": false,
3   "message": "User tidak di temukan"
4 }
```

Uji Coba login dengan password baru :

A screenshot of a REST client interface. The top bar shows a POST request to http://localhost:8080/auth/login with a status of 200 OK, 117 ms, and 220 B. Below the bar, the 'JSON' tab is active, displaying the response body as a JSON object with three fields: 'email' (pon\_user1@quickat.work), 'password' (123), and 'accessToken' (a long alphanumeric string). The 'Preview' tab is also visible, showing the same JSON data in a visual representation.

6. Masuk atau mendaftar dengan provider (google)

Metode : GET

URLs Endpoint :

[http://localhost:8080/oauth2/authorize/google?redirect\\_uri=http://localhost:8080/documentation](http://localhost:8080/oauth2/authorize/google?redirect_uri=http://localhost:8080/documentation)

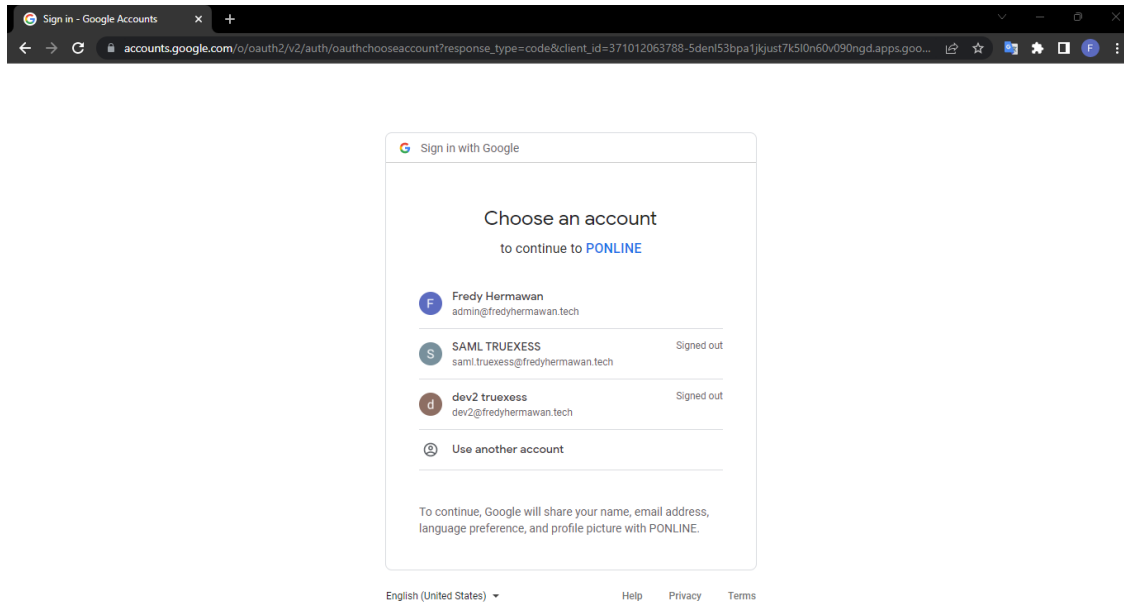
Autentikasi : none

Request Body : none

Data user sebelum :

	cfdvobqnl	tbfewjerbs	lfcmluvud	kwbrtpgltm	ggxafxwykfd	jtixhfcmdv	ccuombxii	oebwazzvdi	uwegexwxuf	xvfeumirr	typyhbwgqm
<input type="checkbox"/>		1 pon_user1@quickat.work	1	NULL	POINLNE USER 1	\$2a\$10\$fvzw7JCG\$2zmQTwH8y4om.scvprKv3lSteNlnP8.Odd...	local	NULL	ROLE_USER	NULL	NULL

Proses Login :



Database sesudah :

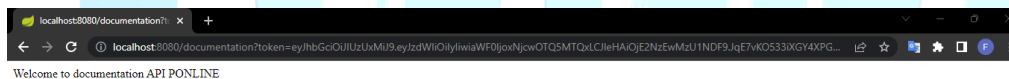
	cfvdcbgj	tbfewjerbs	lfcmfluvud	kwbrtptgtn	ggxafxwykfd	jtxhfdmv	ccuombbxii	oebwazzvdi	uwegexwxuf	xvfeemimr
1	pon_user1@quickat.work	1	NULL		PONLINE USER 1	\$2a\$10\$fwzw7JC9S2zmQTwh8y4om.scvpxkV3/SteNhp9.Odd...	local	NULL	ROLE_USER	NULL
2	admin@fredyhermawan.tech	1	https://lh3.googleusercontent.com/a/AEdFTp0i2Obdd0...	Fredy Hermawan	NULL	google	105983389413348734817	ROLE_USER	NULL	NULL

Response : untuk sekarang di alihkan ke halaman welcome documentation

*Note: nanti akan di arahkan ke halaman dashboard jika frontend / client sudah di mungkinkan untuk mengonsumsi api.*

Untuk informasi token, saat ini di kirimkan di link urlnya :

[http://localhost:8080/documentation?token=eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiIyIiwiaWF0IjoxNjcwOTQ5MTQxLjIleHAiOiJlE2NzEwMzU1NDI0FjJqE7vKO533iXGY4XPGAz8sM\\_1BBVxHtALVd42mlle9bHAa61V9J17grA9gMzi\\_SgdNVxdKwjewupZtpVgXGGZQ](http://localhost:8080/documentation?token=eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiIyIiwiaWF0IjoxNjcwOTQ5MTQxLjIleHAiOiJlE2NzEwMzU1NDI0FjJqE7vKO533iXGY4XPGAz8sM_1BBVxHtALVd42mlle9bHAa61V9J17grA9gMzi_SgdNVxdKwjewupZtpVgXGGZQ)



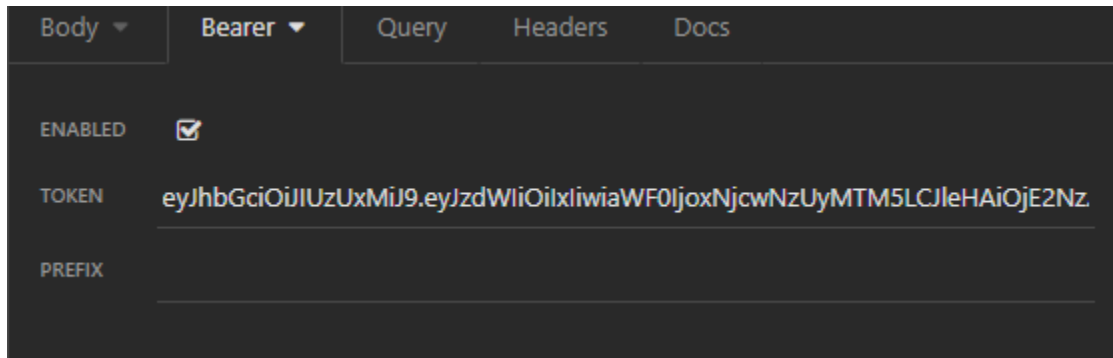


## 7. Mendapatkan Data user (user details/profil)

Metode : GET

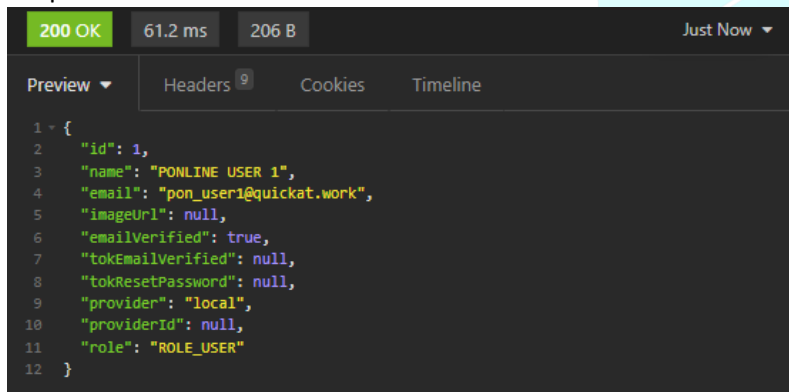
URLs Endpoint : <http://localhost:8080/user/details>

Autentikasi : bearer → accessToken



Request Body : none

Response : Entitas User



## Penjelasan Design UI/UX sesi 1

Dalam mendesain tampilan kami melakukan dua tahapan yang pertama membuat low UI dan High UI menggunakan tool FIGMA

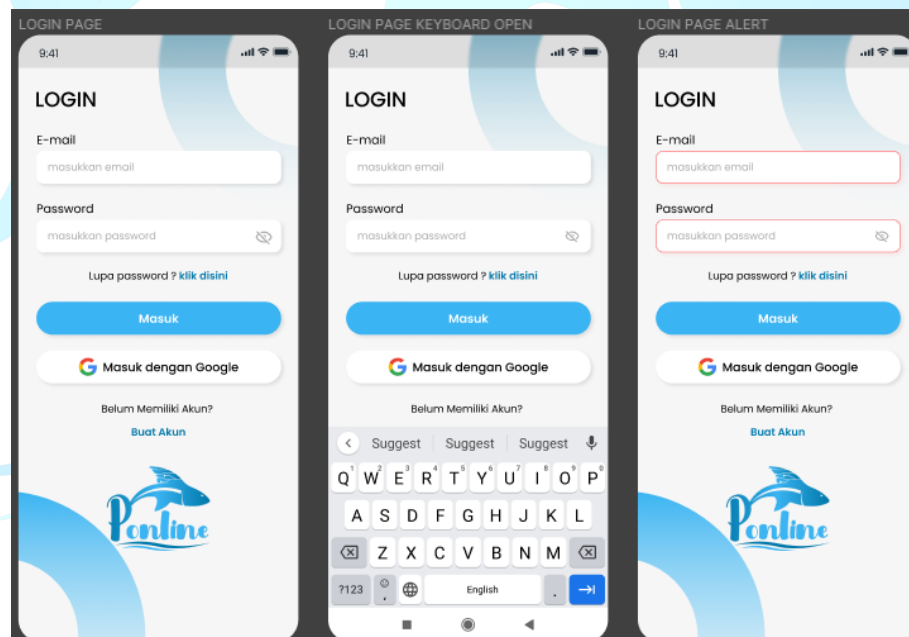
## 1. Splash screen

Halaman yang akan tampil pada awal membuka aplikasi serta saat terjadi loading load data.



## 2. Login page

Halaman login page terdapat form login serta beberapa atributnya



### Form Login

Field Email : menerima input text format email jika input mengalami error maka field akan menunjukkan alert errornya

Field Password : menerima input text yang memiliki fungsi show and hide password

Text button Lupa password: button yang akan mengarahkan ke halaman lupa password

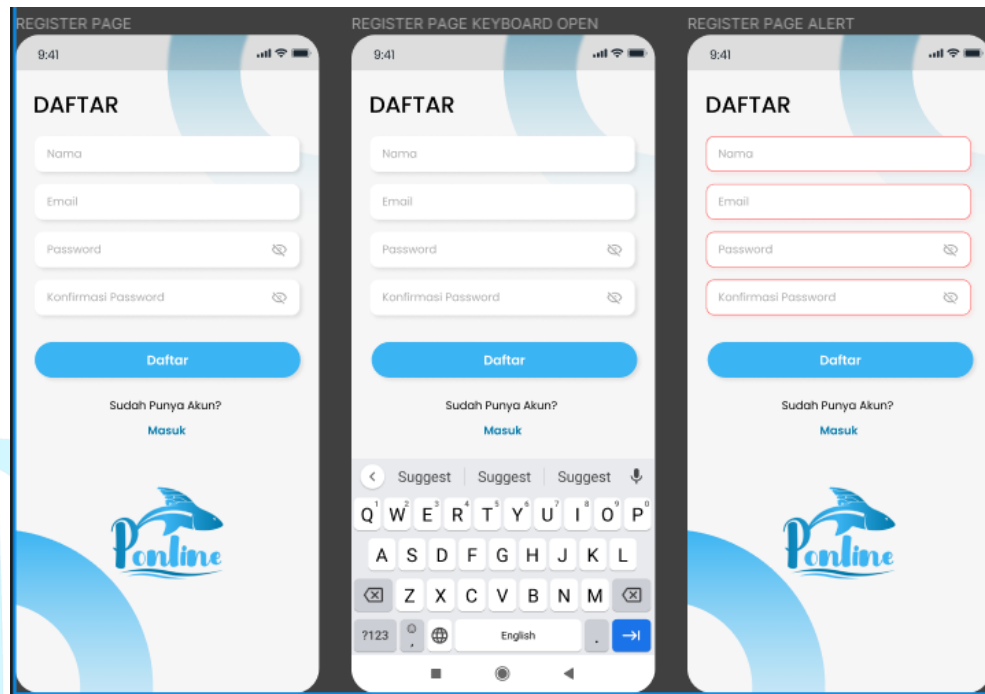
Button Masuk : setelah mengisi email dan password button ini akan mengirim value ke backend untuk melakukan verifikasi data.

Button Masuk Google: melakukan login menggunakan akun google yang sudah ada jika belum pernah mendaftar akun menggunakan google maka akan membuat akun baru menggunakan data dari akun google yang sudah ada.

Text button Buat Akun : button yang akan mengarahkan ke halaman daftar akun

### 3. Register Page

Halaman form pendaftaran akun yang terdapat form serta beberapa atribut lainnya.



#### Form Daftar

Field Nama : menerima input text nama user

Field Email : menerima input text format email

Field Password : menerima input text yang memiliki fungsi show and hide password

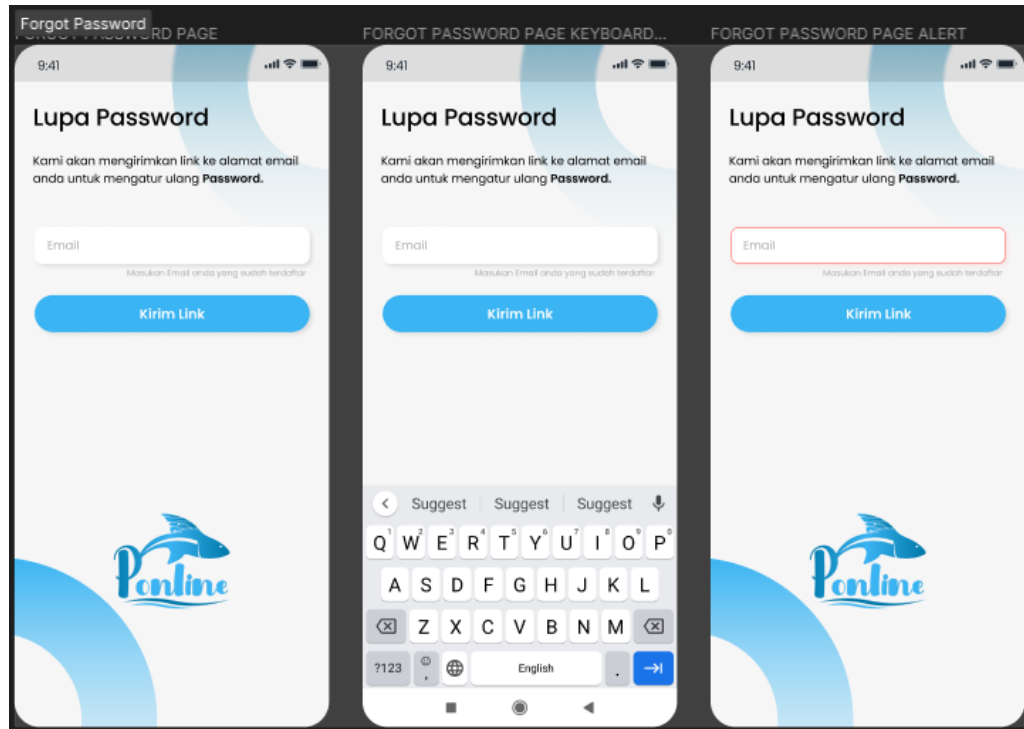
Field Konfirmasi Password : menerima input text yang digunakan untuk konfirmasi password yang sudah di input sebelumnya.

Button Daftar : setelah mengisi data yang sudah di butuhkan button ini akan mengirim value ke backend untuk membuat akun baru.

Text Button Masuk: button yang akan mengarahkan ke halaman login

### 4. Lupa Password Page

Halaman Lupa password yang di gunakan untuk meminta peresetan password menurut email akun terdaftar, Setelah melakukan input email akun yang akan di reset maka system akan mengirimkan link reset pada email.



Form Lupa Password

Field email : menerima input text email

Button Kirim Link : mengirimkan link reset kepada email terdaftar, lalu mengarahkan halaman pada Cek Email Page

##### 5. Cek email page

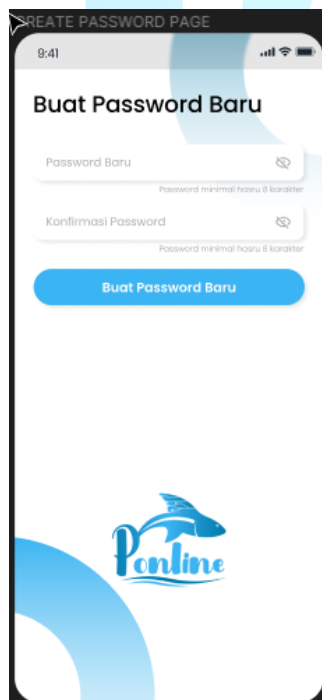
Setelah mengirim link reset password maka akan mengarah pada halaman ini,



Button Masuk : button yang akan mengahkan pada login page

#### 6. Create Password Page

Setelah menerima link resept password melalui email maka akan mengarahkan user pada ham create Password, ham an ini memuat form yang digunakan untuk membuat password baru



#### Form Create Password

Field Password Baru : menerima inut text yang memiliki fungsi show and hide password

Field Konfirmasi Password Baru : menerima inut text yang digunakan untuk konfirmasi password yang sudah di input sebelumnya.

Button Buat Password Baru : button ini berfungsi untuk mengirinkam value form pada system dan membuat password baru pada akun, setelah ini user akan mengarah pada login page.

#### 7. Dashboard Page

Dalam Pengerjaan . . .

