

LAPORAN TUGAS UTS
MATA KULIAH PEMROGRAMAN BACKEND
(PRAKTIKUM)

“Wireframe dan Collection Data”

Dosen Pengampu:
Rachman Sinatriya Marjianto, B.Eng.,M.Sc.



Disusun Oleh :
Azizah Ika Wulandari 434221045

D-IV TEKNIK INFORMATIKA

FAKULTAS VOKASI

UNIVERSITAS AIRLANGGA

2024

Tugas

1. Buatlah wireframe dari SSO unair satu

a. Tampilan di pengguna (Disain berdasarkan tampilan SSO unair)

b. Tampilan di admin. Silahkan anda disain bentuk halaman untuk manajemen usernya.

Garis besar alurnya adalah:

i. Tarik data dari cybercampus berdasarkan jenis usernya (mahasiswa, dosen atau tendik) dan input NIM / NIK

ii. Sistem akan memberi list aplikasi standar

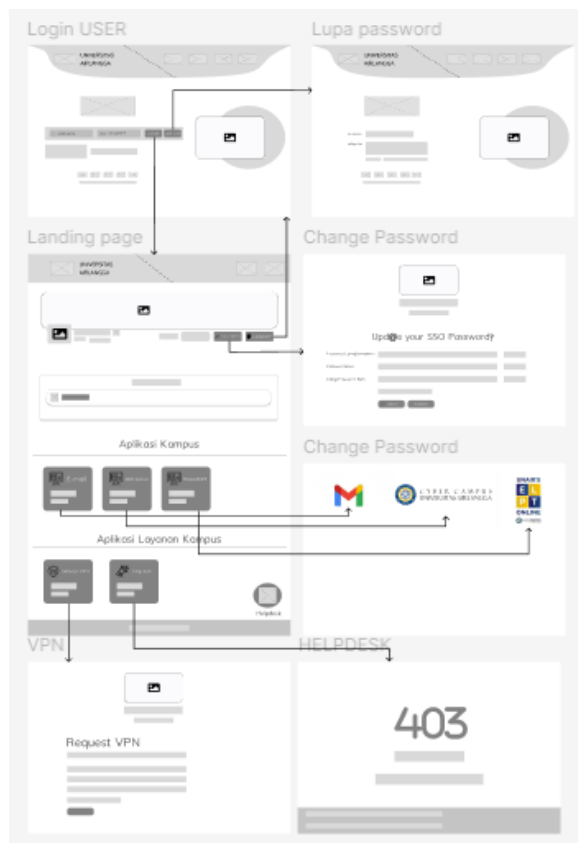
iii. Admin dapat menambahkan aplikasi lain diluar aplikasi standar.

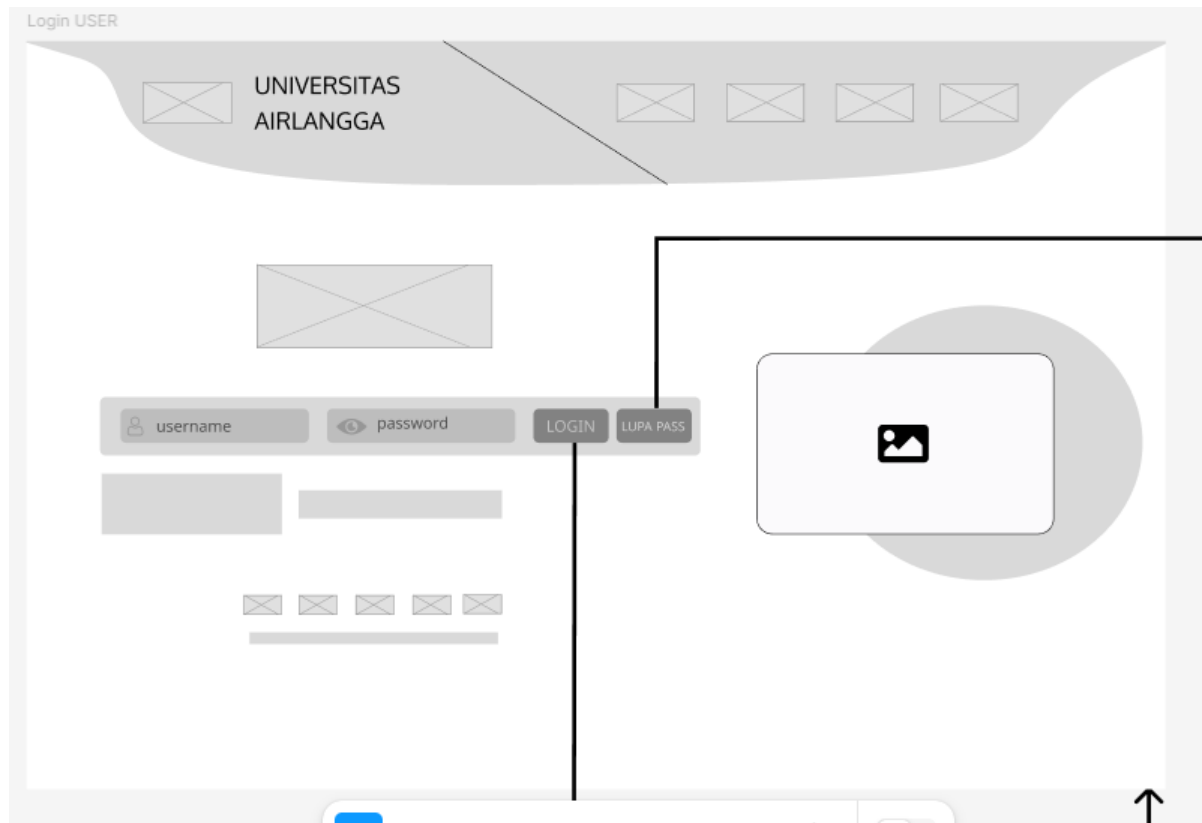
2. Disain bentuk collection yang direkomendasi mongoDB berdasarkan wireframe yang telah anda buat.

JAWABAN

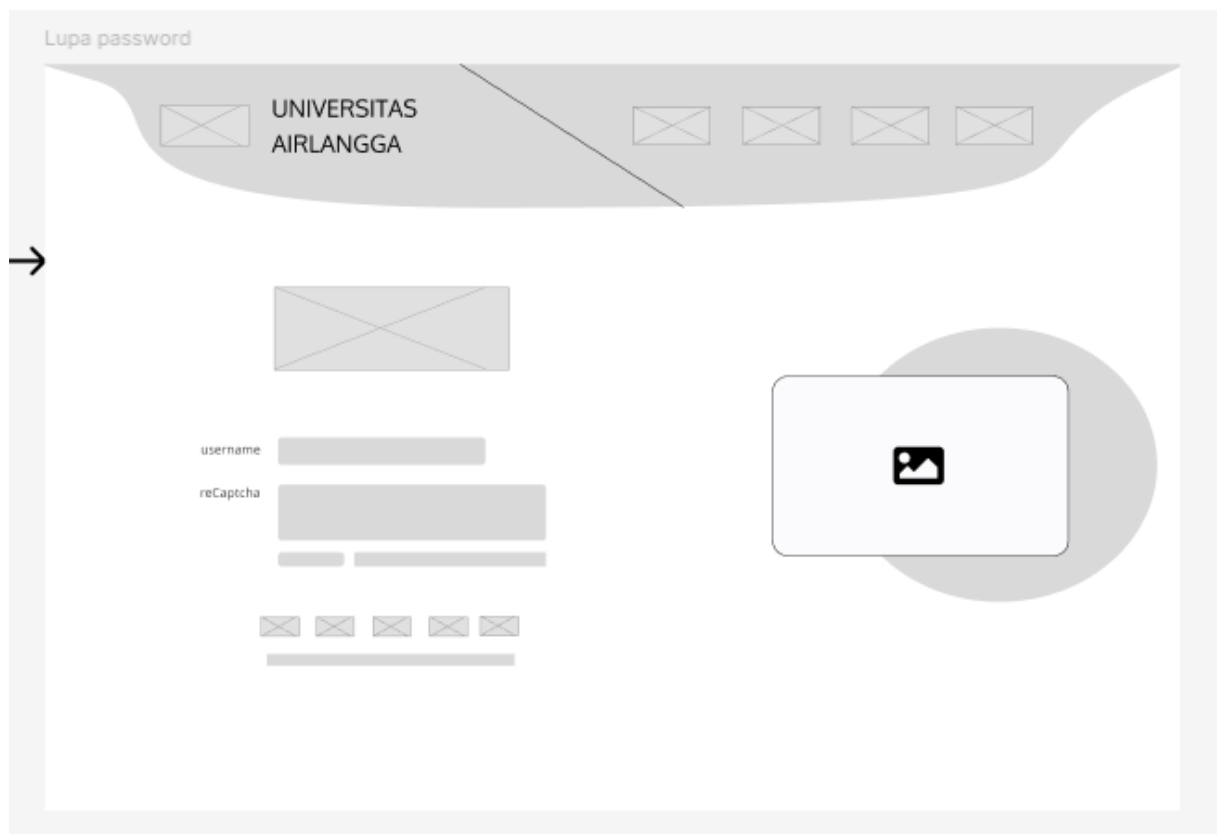
1. WIREFRAME

USER (Keseluruhan DESIGN)

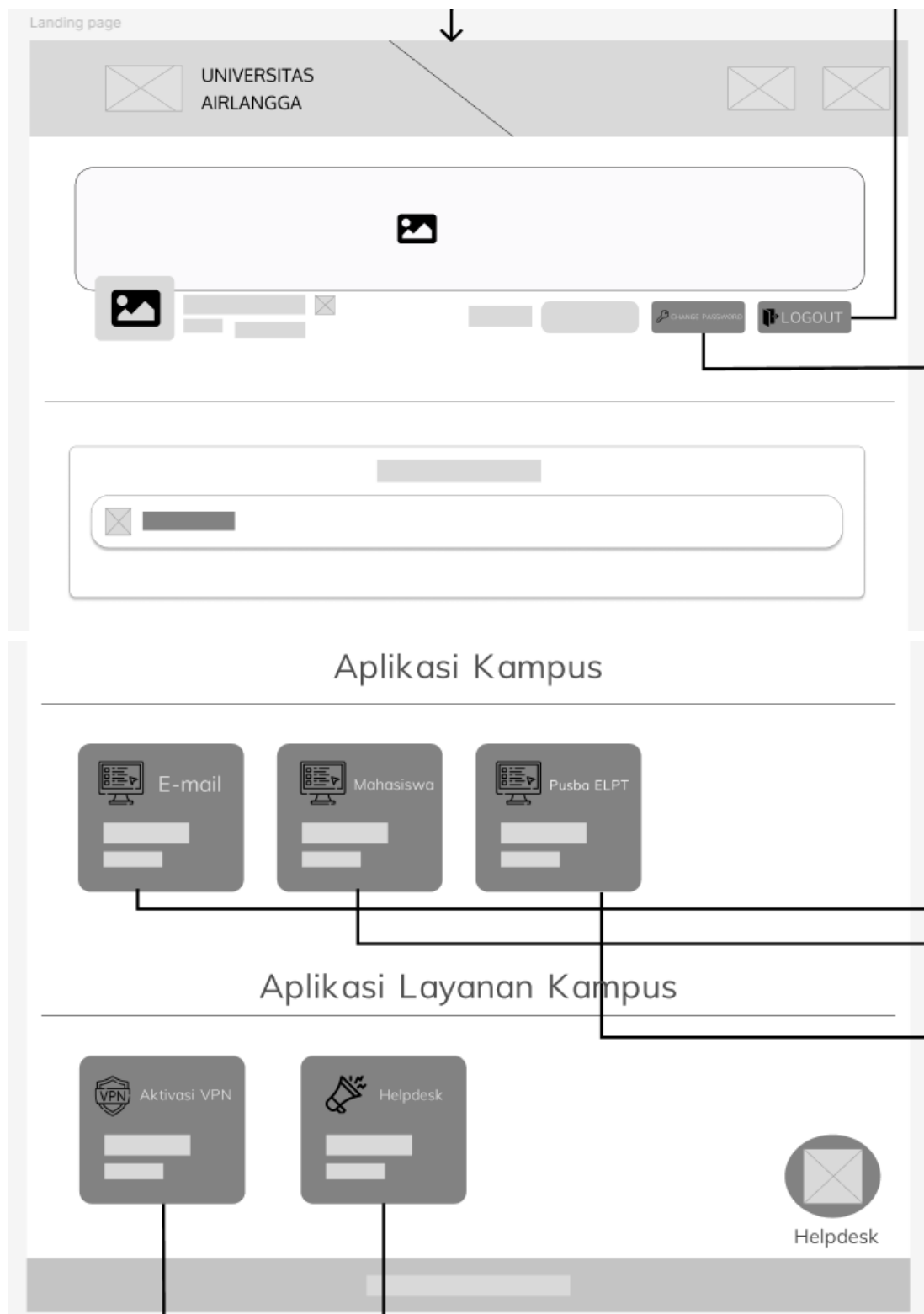




- **Login USER** (user dapat login dengan memasukkan username beserta password)




- **Lupa Password** (user yang lupa dengan passwordnya akan mendapatkan email mengenai password lamanya.)



- **Landing Page** (User dapat melihat profile, mengubah password, logout, dan pencarian aplikasi yang diinginkan. Baik aplikasi kampus maupun aplikasi layanan kampus, beserta ada button melayang untuk helpdesk yang dapat melayani user.)

Change Password



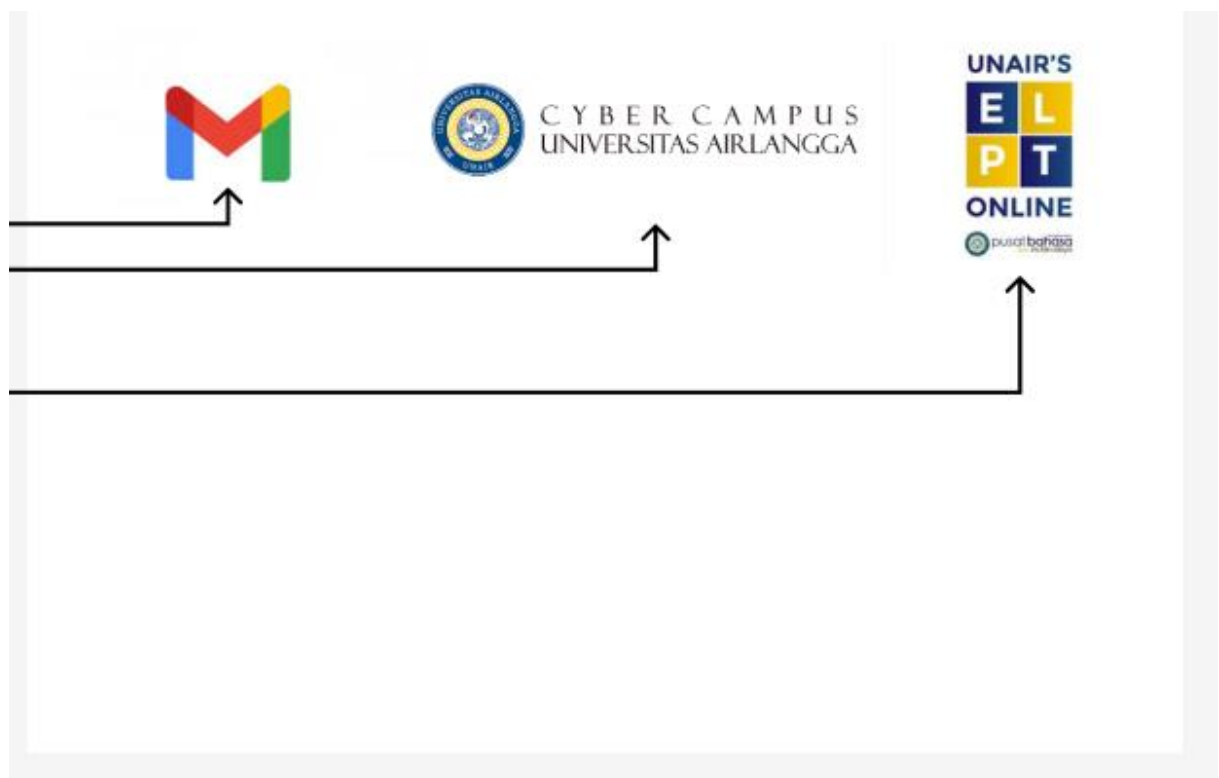
Upd@e your SSO Password?

→ Password Lama/Sementara

Password Baru

Ulangi Password Baru

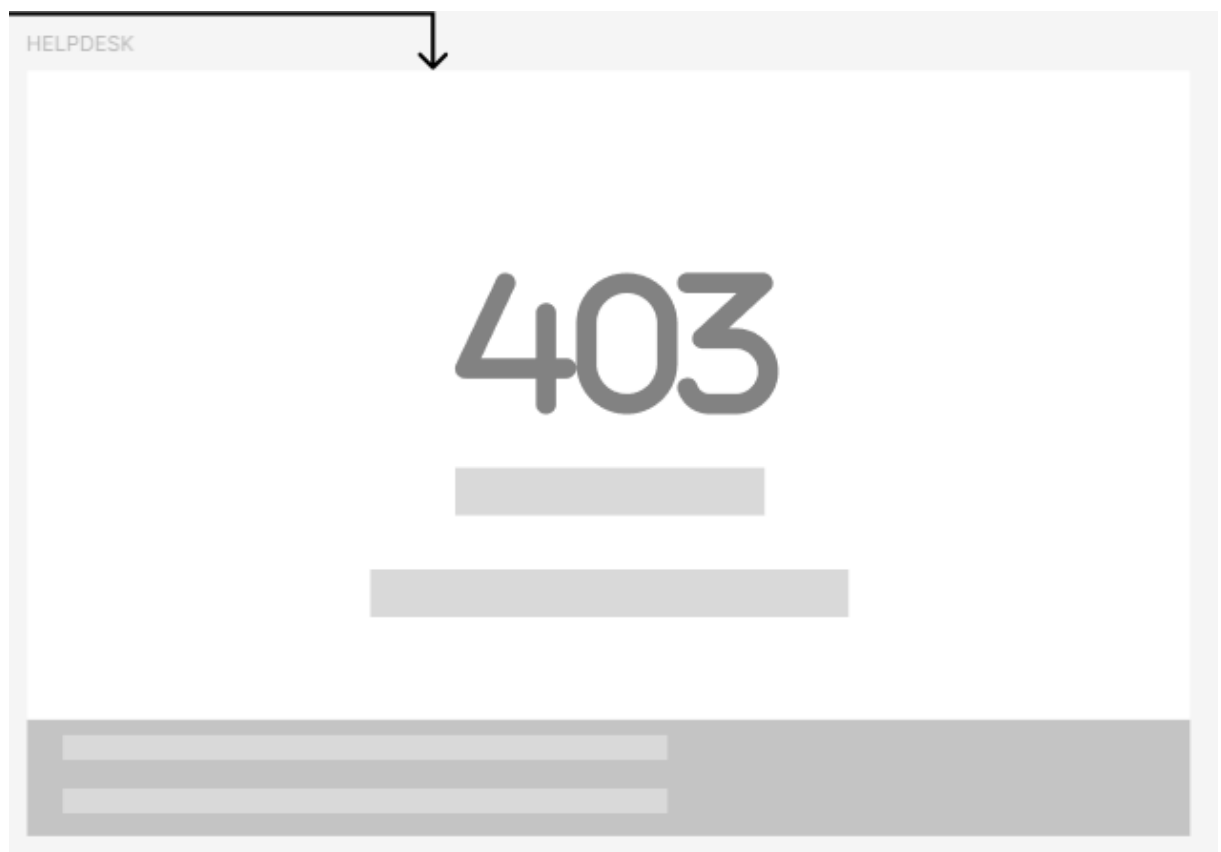
- **Change Password** (User dapat mengubah password lamanya menjadi password yang baru, jika ingin menggantinya.)



- **Ketiga aplikasi yang tersedia di aplikasi kampus** (tergantung user menekan button yang mana.)

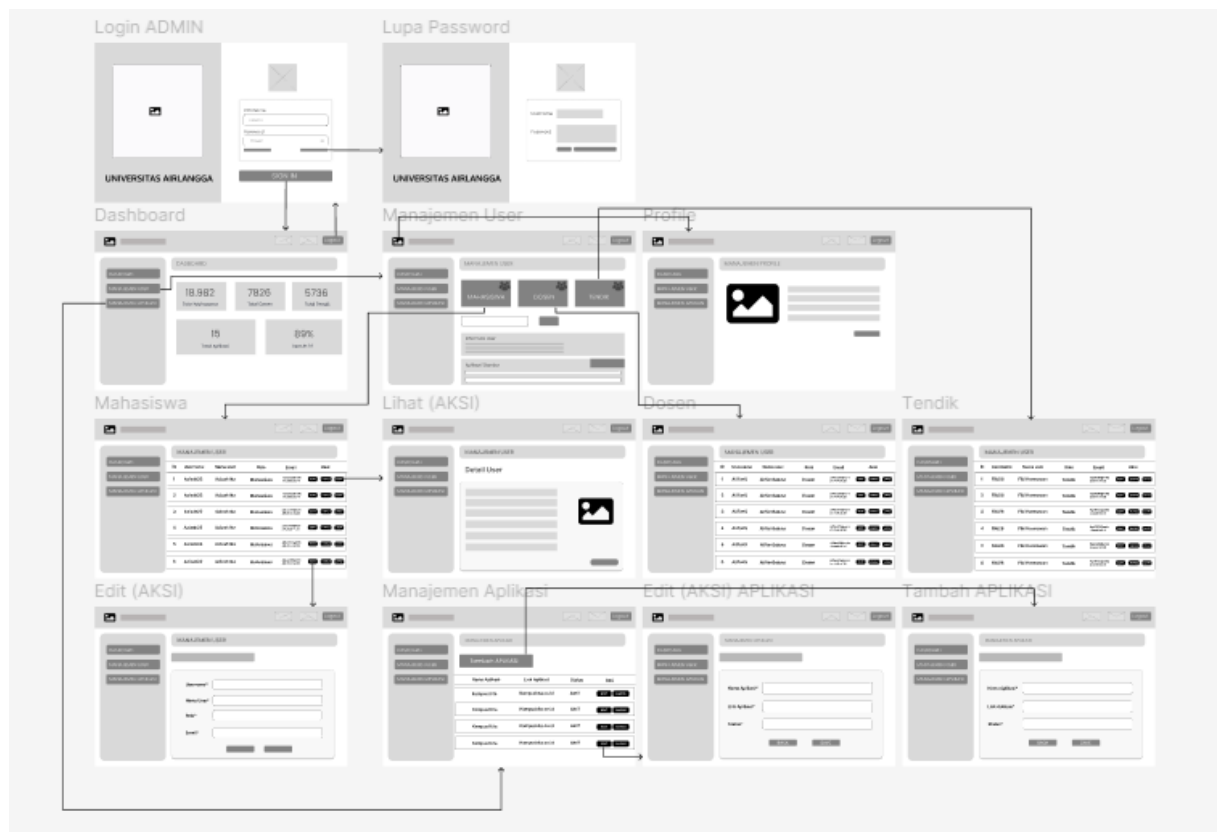


- **VPN** (Jika user menekan Aktivasi VPN)



- **HELPDESK** (Ketika user menekan button HELPDESK).

ADMIN (Keseluruhan DESIGN)



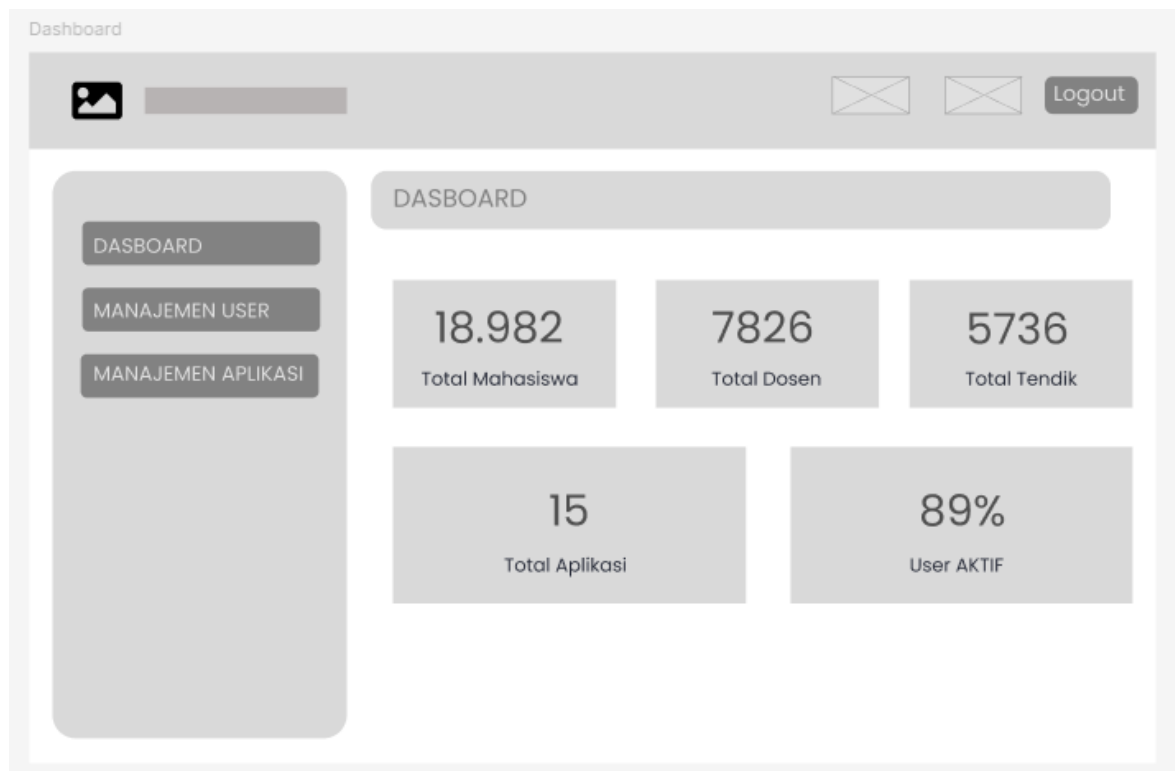
This is a detailed wireframe of the Login ADMIN screen. It includes the following elements:

- Header**: Login ADMIN
- Profile Placeholder**: A large square area with a placeholder image icon.
- University Name**: UNIVERSITAS AIRLANGGA
- Login Form**:
 - Username**: A text input field with the placeholder text 'User Name'.
 - Password**: A text input field with the placeholder text 'Password' and a strength indicator.
- Sign In Button**: A large button labeled SIGN IN.

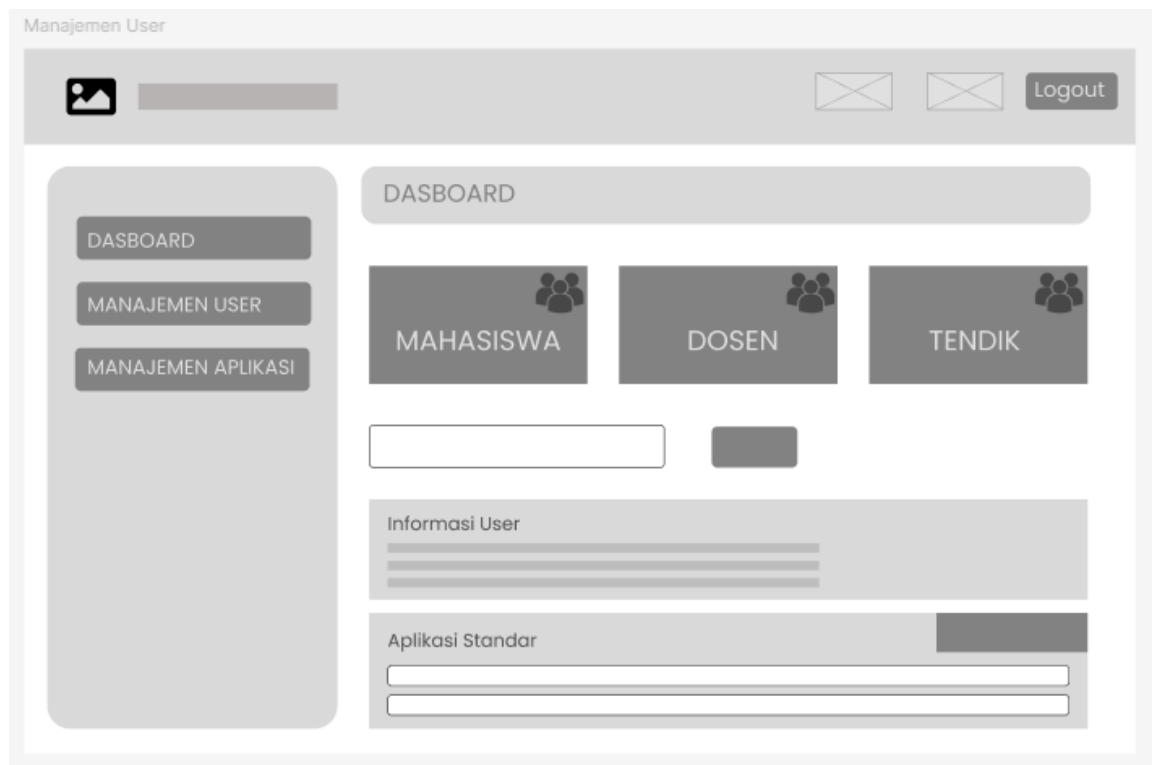
- **Login ADMIN** (Admin memasukkan username dan password untuk login)



- **Lupa password** (Jika Admin lupa dengan passwordnya, dia akan mendapatkan email mengenai password lamanya.)



- **Dashboard** (Terdapat informasi keseluruhan dari jumlah mahasiswa,dosen,tendik,aplikasi (modul),dan jumlah keseluruhan user yang aktif).Dan ada sidebar yang berisi dashboard,manajemen user, dan manajemen aplikasi.






- **Manajemen user** (Berisi keseluruhan informasi lebih lanjut dari mahasiswa, dosen, dan tendik). Dan ada button pencarian dengan memasukkan NIM/NIK untuk mendapatkan informasi dari user yang ingin dicari yang akan memunculkan informasi user dan aplikasi standar apa yang dimiliki.



- **Profile** (Berisi informasi dari ADMIN).

Mahasiswa




[Logout](#)

[DASHBOARD](#)
[MANAJEMEN USER](#)
[MANAJEMEN APLIKASI](#)




MANAJEMEN USER

ID	Username	Nama user	Role	Email	Aksi
1	Azizah22	Azizah Ika	Mahasiswa	azizah22@vokasi.unair.ac.id	EDIT HAPUS LIHAT
2	Azizah22	Azizah Ika	Mahasiswa	azizah22@vokasi.unair.ac.id	EDIT HAPUS LIHAT
3	Azizah22	Azizah Ika	Mahasiswa	azizah22@vokasi.unair.ac.id	EDIT HAPUS LIHAT
4	Azizah22	Azizah Ika	Mahasiswa	azizah22@vokasi.unair.ac.id	EDIT HAPUS LIHAT
5	Azizah22	Azizah Ika	Mahasiswa	azizah22@vokasi.unair.ac.id	EDIT HAPUS LIHAT
6	Azizah22	Azizah Ika	Mahasiswa	azizah22@vokasi.unair.ac.id	EDIT HAPUS LIHAT

- Manajemen user - Mahasiswa

Berisi keseluruhan tabel dari mahasiswa dan menampilkan informasi dari mahasiswa-mahasiswa tersebut). Dan ada button aksi yang dapat diedit, dihapus, dan dilihat (informasi lebih lanjut dari masing-masing mahasiswa).

Dosen




[Logout](#)

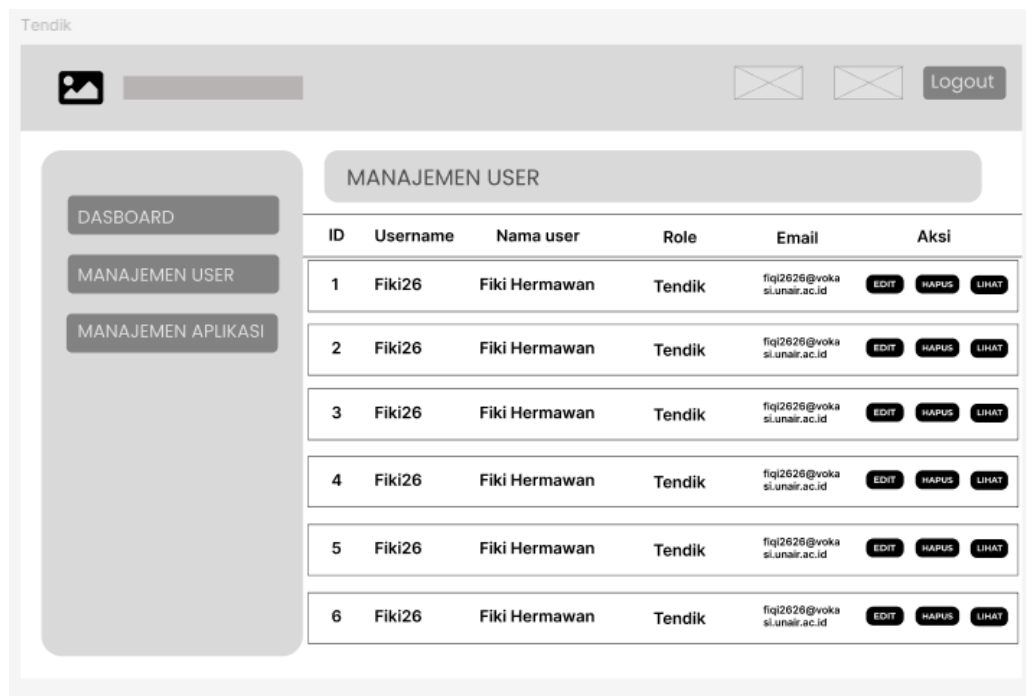
[DASHBOARD](#)
[MANAJEMEN USER](#)
[MANAJEMEN APLIKASI](#)

MANAJEMEN USER

ID	Username	Nama user	Role	Email	Aksi
1	AlifianS	AlifianSukma	Dosen	alifian25@vokasi.unair.ac.id	EDIT HAPUS LIHAT
2	AlifianS	AlifianSukma	Dosen	alifian25@vokasi.unair.ac.id	EDIT HAPUS LIHAT
3	AlifianS	AlifianSukma	Dosen	alifian25@vokasi.unair.ac.id	EDIT HAPUS LIHAT
4	AlifianS	AlifianSukma	Dosen	alifian25@vokasi.unair.ac.id	EDIT HAPUS LIHAT
5	AlifianS	AlifianSukma	Dosen	alifian25@vokasi.unair.ac.id	EDIT HAPUS LIHAT
6	AlifianS	AlifianSukma	Dosen	alifian25@vokasi.unair.ac.id	EDIT HAPUS LIHAT

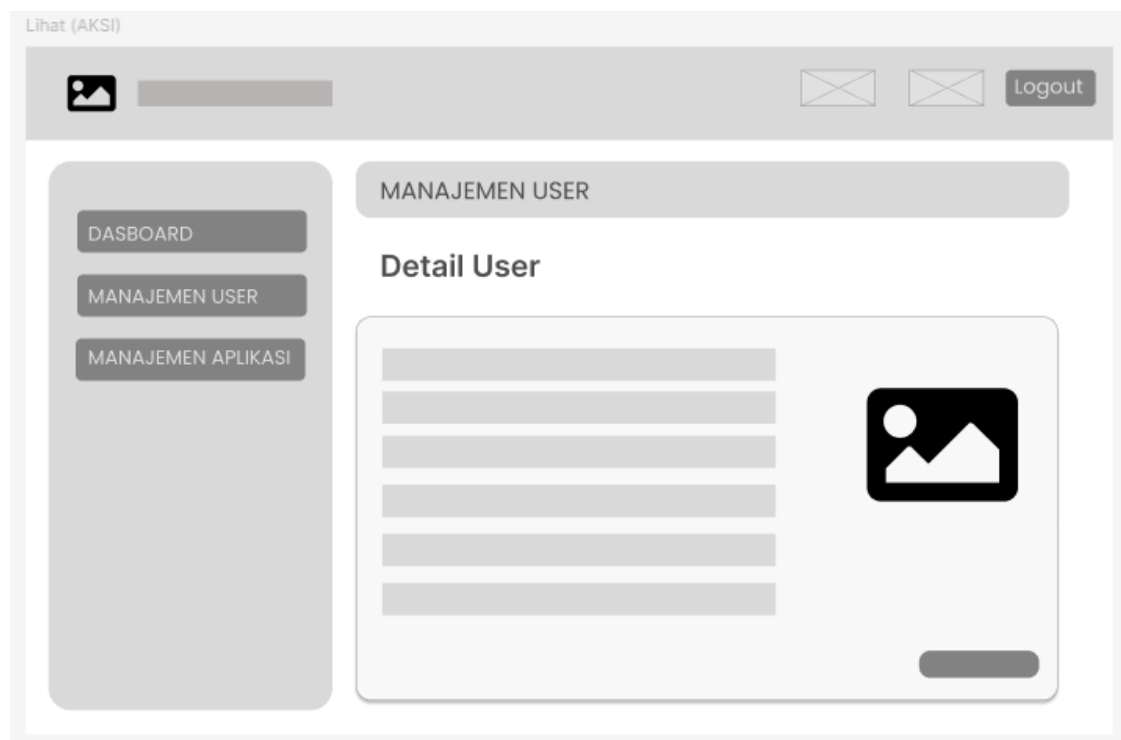
- Manajemen user - Dosen

Berisi keseluruhan tabel dari dosen dan menampilkan informasi dari dosen-dosen tersebut). Dan ada button aksi yang dapat diedit, dihapus, dan dilihat (informasi lebih lanjut dari masing-masing dosen).



- Manajemen user - Tendik

Berisi keseluruhan tabel dari tendik dan menampilkan informasi dari tendik-tendik tersebut). Dan ada button aksi yang dapat diedit, dihapus, dan dilihat (informasi lebih lanjut dari masing-masing tendik).



- Manajemen user – Detail user

Berisi keseluruhan informasi yang dimiliki oleh masing-masing user (mahasiswa, tendik, dosen).

Edit (AKSI)

DASHBOARD

MANAJEMEN USER

MANAJEMEN APLIKASI

MANAJEMEN USER

Username*

Nama User*

Role*

Email*

Save

Cancel

- **Manajemen user – Edit (AKSI)**

Jika admin ingin mengedit beberapa bagian user bisa disini.

Manajemen Aplikasi

DASHBOARD

MANAJEMEN USER

MANAJEMEN APLIKASI

MANAJEMEN APLIKASI

Tambah APLIKASI

Nama Aplikasi	Link Aplikasi	Status	Aksi
Kampus Kita	Kampuskita.co.id	Aktif	<p>EDIT</p> <p>HAPUS</p>
Kampus Kita	Kampuskita.co.id	Aktif	<p>EDIT</p> <p>HAPUS</p>
Kampus Kita	Kampuskita.co.id	Aktif	<p>EDIT</p> <p>HAPUS</p>
Kampus Kita	Kampuskita.co.id	Aktif	<p>EDIT</p> <p>HAPUS</p>

- **Manajemen Aplikasi**

Berisi seluruh aplikasi yang dimiliki oleh ADMIN.

Edit (AKSI) APLIKASI

DASHBOARD

MANAJEMEN USER

MANAJEMEN APLIKASI

MANAJEMEN APLIKASI

Nama Aplikasi*

Link Aplikasi*

Status*

BACK SAVE

- Manajemen Aplikasi – Edit Aplikasi

Jika admin ingin mengedit aplikasi satndar yang dimiliki.

Tambah APLIKASI

DASHBOARD

MANAJEMEN USER

MANAJEMEN APLIKASI

MANAJEMEN APLIKASI

Nama Aplikasi*

Link Aplikasi*

Status*

BACK SAVE

- Manajemen Aplikasi – Tambah Aplikasi

Jika admin ingin menambahkan aplikasi satndar lainnya.

2. COLLECTION DATA

```
> use manajemen_aplikasi
switched to db manajemen_aplikasi
```

Penjelasan: Membuat database dan beralih ke database yang baru.

```
> db.createCollection("roles")
db.roles.insertMany([
  { "name": "admin", "created_at": new Date(), "updated_at": new Date() },
  { "name": "user", "created_at": new Date(), "updated_at": new Date() }
])
{ "acknowledged" : true, "insertedIds" : [ ObjectId("..."), ObjectId("...") ] }
```

Penjelasan: Membuat koleksi 'roles' dan menambahkan dua dokumen (admin dan user). Output menunjukkan operasi berhasil dan memberikan ID untuk setiap dokumen yang dimasukkan.

```
> db.createCollection("jenis_user")
db.jenis_user.insertMany([
  { "name": "Dosen", "created_at": new Date(), "updated_at": new Date() },
  { "name": "Mahasiswa", "created_at": new Date(), "updated_at": new Date() },
  { "name": "Tendik", "created_at": new Date(), "updated_at": new Date() }
])
{ "acknowledged" : true, "insertedIds" : [ ObjectId("..."), ObjectId("..."), ObjectId("...") ] }
```

Penjelasan: Membuat koleksi 'jenis_user' dan menambahkan tiga jenis user (mahasiswa, dosen, dan tendik). Output menunjukkan operasi berhasil dan memberikan ID untuk setiap dokumen.

```
> db.createCollection("modul")
db.modul.insert({
  "name": "Cyber Kampus",
  "kategori": ObjectId("..."), // Masukkan ObjectId dari koleksi kategori jika ada
  "deskripsi": "Aplikasi manajemen perkuliahan untuk mahasiswa, dosen, dan tendik",
  "url_icon": "/path/to/icon",
  "created_at": new Date(),
  "updated_at": new Date()
})
{ "acknowledged" : true, "insertedId" : ObjectId("...") }
```

Penjelasan: Membuat koleksi 'modul' dan menambahkan satu modul 'Cyber Kampus'. Output menunjukkan operasi berhasil dan memberikan ID untuk dokumen yang dimasukkan.

Noted : Modul disini itu aplikasi.

```
> db.createCollection("template_modul")
db.template_modul.insert({
  "jenis_user": ObjectId("..."), // Masukkan ObjectId dari jenis user mahasiswa
  "modul": [ ObjectId("...") ], // Masukkan ObjectId dari modul yang sudah dibuat
  "created_at": new Date(),
  "updated_at": new Date()
})
{ "acknowledged" : true, "insertedId" : ObjectId("...") }
```

Penjelasan: Membuat koleksi 'template_modul' dan menambahkan satu template. Output menunjukkan operasi berhasil dan memberikan ID untuk dokumen yang dimasukkan.

```
> db.createCollection("users")
db.users.insert({
  "username": "AzizahIka",
  "password": "Azizah12345",
  "email": "Azizahika@example.com",
  "role": ObjectId("..."), // Masukkan ObjectId dari role dosen
  "created_at": new Date(),
  "updated_at": new Date(),
  "jenis_user": ObjectId("..."), // Masukkan ObjectId dari jenis user dosen
  "modul": [ ObjectId("...") ], // Modul-modul yang dimiliki user
  "profile": {
    "name": "Azizah22",
    "phone": "08123456789",
    "photo": "/path/to/photo",
    "address": "Jl. Ikan Mujaer No. 1"
  }
})
```

```

    "name": "Azizah22",
    "phone": "08123456789",
    "photo": "/path/to/photo",
    "address": "Jl. Ikan Mujaer No. 1"
  }
})
{ "acknowledged" : true, "insertedId" : ObjectId("...") }

```

Penjelasan: Membuat koleksi 'users' dan menambahkan satu user 'AzizahIka'. Output menunjukkan operasi berhasil dan memberikan ID untuk dokumen yang dimasukkan.

```

> db.createCollection("user_modul")
db.user_modul.insert({
  "user_id": ObjectId("..."), // Masukkan ObjectId dari user AzizahIka
  "modul": [ ObjectId("...") ], // Modul-modul yang dimiliki user
  "created_at": new Date(),
  "updated_at": new Date()
})
{ "acknowledged" : true, "insertedId" : ObjectId("...") }

```

Penjelasan: Membuat koleksi 'user_modul' dan menambahkan relasi antara user dan modul. Output menunjukkan operasi berhasil dan memberikan ID untuk dokumen yang dimasukkan.

```

> db.createCollection("kategori")
db.kategori.insert({
  "name": "Akademik",
  "created_at": new Date(),
  "updated_at": new Date()
})
{ "acknowledged" : true, "insertedId" : ObjectId("...") }

```

Penjelasan: Membuat koleksi 'kategori' dan menambahkan kategori 'Akademik'. Output menunjukkan operasi berhasil dan memberikan ID untuk dokumen yang dimasukkan.


```
> db.users.find().pretty()  
db.roles.find().pretty()  
db.modul.find().pretty()  
db.template_modul.find().pretty()  
db.user_modul.find().pretty()
```

```
{  
  "_id" : ObjectId("..."),  
  "username" : "AzizahIka",  
  "password" : "Azizah12345",  
  "email" : "Azizahika@example.com",  
  "role" : ObjectId("..."),  
  "created_at" : ISODate("2024-10-04T10:00:00Z"),  
  "updated_at" : ISODate("2024-10-04T10:00:00Z"),  
  "jenis_user" : ObjectId("..."),  
  "modul" : [ ObjectId("...") ],  
  "profile" : {  
    "name" : "Azizah22",  
    "phone" : "08123456789",  
    "photo" : "/path/to/photo",  
    "address" : "Jl. Ikan Mujaer No. 1"  
  }  
}
```

Penjelasan: Perintah-perintah ini menampilkan semua dokumen dari koleksi yang berbeda dalam format yang mudah dibaca. Output menunjukkan contoh dari dokumen user yang dimasukkan sebelumnya