

**RANCANGAN SISTEM VALIDASI BAGASI BERBASIS
RADIO FREQUENCY IDENTIFICATION MENGGUNAKAN
WEBSITE**

TUGAS AKHIR



Oleh:

NANA KARTIKA SARI
NIT. 30218018

**PROGRAM STUDI DIPLOMA 3 TEKNIK NAVIGASI UDARA
POLITEKNIK PENERBANGAN SURABAYA
2021**

**RANCANGAN SISTEM VALIDASI BAGASI BERBASIS
RADIO FREQUENCY IDENTIFICATION MENGGUNAKAN
WEBSITE**

TUGAS AKHIR

Diajukan sebagai Syarat Menempuh Mata Kuliah Tugas Akhir Pada Program
Studi Diploma 3 Teknik Navigasi Udara



Oleh:

NANA KARTIKA SARI
NIT. 30218018

**PROGRAM STUDI DIPLOMA 3 TEKNIK PESAWAT UDARA
POLITEKNIK PENERBANGAN SURABAYA
2021**

LEMBAR PERSETUJUAN

RANCANGAN SISTEM VALIDASI BAGASI BERBASIS *RADIO FREQUENCY IDENTIFICATION* DALAM MENGGUNAKAN WEBSITE

Oleh:
NANA KARTIKA SARI
NIT. 30218018

Disetujui untuk diujikan pada:
Surabaya, 13 Agustus 2021

Pembimbing I : Dr. YUYUN SUPRAPTO, S.SiT, MM
NIP. 19820107 200502 2 001



Pembimbing II : TEGUH IMAM SUHARTO, MT
NIP. 19910913 201503 1 003




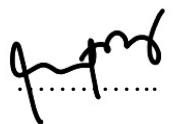

LEMBAR PENGESAHAN

RANCANGAN SISTEM VALIDASI BAGASI BERBASIS *RADIO FREQUENCY IDENTIFICATION* MENGGUNAKAN WEBSITE

Oleh:
NANA KARTIKA SARI
NIT. 30218018

Telah dipertahankan dan dinyatakan lulus pada Ujian Tugas Akhir
Program Pendidikan Diploma III Teknik Navigasi Udara
Politeknik Penerbangan Surabaya
Pada tanggal: 13 Agustus 2021

Panitia Penguji:

1. Ketua : BAMBANG BAGUS H., S.SiT, MM, MT
NIP. 19810915 200502 1 001 
2. Sekretaris : MEITA MAHARANI S., M.Pd
NIP. 19800502 200912 2 001 
3. Anggota : Dr. YUYUN SUPRAPTO, S.SiT, MM
NIP. 19810529 200812 1 001 

Ketua Program Studi
D III Teknik Navigasi Udara



NYARIS PAMBUDIYATNO, S.SiT, M.MTr
NIP. 19820525 200502 1 001

ABSTRAK

RANCANGAN SISTEM VALIDASI BAGASI BERBASIS *RADIO FREQUENCY IDENTIFICATION* MENGGUNAKAN WEBSITE

Oleh :

NANA KARTIKA SARI

NIT : 30218018

Penelitian ini mendeskripsikan perancangan alat pendeteksi bagasi tercatat menggunakan *Radio Frequency Identification* (RFID) yang terintegrasi dengan sistem web. Pemeriksaan bagasi tercatat secara manual memungkinkan terjadinya *human error* dalam proses pemeriksaan bagasi tercatat.

Penelitian ini mengatasi permasalahan tersebut, yaitu dengan merancang alat pendeteksi bagasi tercatat yang mampu mendeteksi data bagasi tercatat secara otomatis berbasis RFID. Perancangan dilakukan dengan menggunakan Tag pasif sebagai data bagasi tercatat, RFID *Reader* RC522, Wemos D1 R1, LED, Buzzer.

Data RFID dimanfaatkan untuk diintegrasikan dengan sistem basis data. Data yang didapat dari RFID diolah oleh sistem dan ditampilkan dalam bentuk *text* melalui *website*, sehingga dapat dijadikan acuan untuk pemeriksaan bagasi tercatat oleh petugas bandara.

Kata kunci : Bagasi tercatat, RFID, Wemos D1 R1, Buzzer, Website.

ABSTRACT

*DESIGN OF BAGGAGE VALIDATION BASED OF RADIO FREQUENCY
IDENTIFICATION USED WEBSITE*

By :

NANA KARTIKA SARI

NIT : 30218018

This study describes the design of a checked baggage detection device using of Radio Frequency Identification (RFID) which is integrated with a web system. Manual checked baggage allows for human error in the process of checking checked baggage.

This research solve the problem, by designing a baggage detection device that which is able to detect checked baggage data automatically based on RFID. The design is done with by using a passive tag as checked baggage data based on RFID, RFID reader RC522.

RFID data is used to be integrated with a database system. The data obtained from RFID is processed by the system and displayed in the text via the website, so that it can be used as a reference for checking checked baggage data by airport officials.

Keywords: Checked baggage, RFID, Wemos D1 R1, LED, Buzzer, Website.

PERNYATAAN KEASLIAN DAN HAK CIPTA

Saya yang bertanda tangan dibawah ini :

Nama : Nana Kartika Sari
NIT : 30218018
Program Studi : Diploma 3 Teknik Navigasi Udara
Judul Tugas Akhir : Rancangan Sistem Validasi Bagasi Berbasis *Radio Frequency Identification* Menggunakan Website

Dengan ini menyatakan bahwa :

1. Karya tulis saya dengan judul “Rancangan Alat Validasi Bagasi Tercatat Bermasis Radio Frequency Identification (RFID) Dalam Optimalisasi Kinerja Petugas Bandara” adalah asli dan belum pernah diajukan untuk mendapatkan gelar Ahli Madya (A. Md) di Politeknik Penerbangan Surabaya.
2. Karya tulis ini adalah murni hasil gagasan serta penelitian saya sendiri dengan arahan dari dosen pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis maupun dipublikasikan oleh orang lain, kecuali secara tertulis dengan jelas dicantumkan, sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya dengan norma yang berlaku di Politeknik Penerbangan Surabaya.

Surabaya, 11 Agustus 2021
Yang membuat pernyataan



MOTTO

“ Jangan katakan pada Allah aku punya masalah besar tetapi katakan pada masalah bahwa aku punya Allah Yang Maha Besar”_ Ali bin Abi Thalib

PERSEMBAHAN

Tugas Akhir ini adalah bagian dari ibadahku kepada Allah SWT, karena kepadaNya kami menyembah dan kepadaNya kami mohon pertolongan.

Sekaligus sebagai ungkapan terima kasihku kepada :
Bapak Bagus Budianto dan Ibu Dewi Yanti yang selalu memberikan motivasi dan do'a dalam hidupku
Semua saudara ku yang selalu memberikan inspirasi dalam hidupku
Semua pihak yang membantu untuk menyusun dan mengerjakan Tugas Akhir ini
Teman – teman seperjuangan ku yang selalu kompak dan penuh dengan sebuah cerita

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa, karena berkat limpahan Rahmat dan Hidayah-Nya yang telah memberikan kesehatan, pengetahuan, ketrampilan, pengalaman yang senantiasa diberikan kepada penulis, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **RANCANGAN SISTEM VALIDASI BAGASI BERBASIS RADIO FREQUENCY IDENTIFICATION MENGGUNAKAN WEBSITE**.

Penyusunan Tugas Akhir ini dimaksudkan sebagai salah satu syarat menyelesaikan pendidikan di Politeknik Penerbangan Surabaya dan memperoleh gelar Ahli Madya (A.Md)

Ucapan terima kasih kami sampaikan kepada segenap pihak yang telah membantu selama proses penyusunan Tugas Akhir ini, terutama kepada :

1. Bapak M. Andra Adityawarman, ST, MT selaku Direktur Politeknik Penerbangan Surabaya.
2. Bapak Nyaris Pambudiyatno, S.SiT, M.Tr selaku Kepala Program Studi Teknik Navigasi Udara di Politeknik Penerbangan Surabaya.
3. Ibu Dr. Yuyun Suprpto, S.SiT, MM. selaku pembimbing I yang selalu memberi semangat, pemahaman, ilmu, dan dukungan moril dalam penyusunan Tugas Akhir.
4. Bapak Teguh Imam Suharto, MT selaku pembimbing II yang senantiasa memberikan arahan dan semangat dalam penyusunan Tugas Akhir.
5. Bapak Bagus Budianto dan Ibu Dewi Yanti selaku orang tua saya, yang tak henti-hentinya memberikan doa, ridho, restu serta bantuan secara materi maupun dukungan moral untuk kelancaran Tugas Akhir ini.
6. Seluruh dosen dan *civitas* akademi Prodi D-III Teknik Navigasi Udara Politeknik Penerbangan Surabaya yang telah membantu baik secara langsung maupun tidak langsung.
7. Seluruh rekan-rekan Taruna Navigasi Udara XI yang selalu memberi semangat, dan mengisi hari-hari saya selama menempuh pendidikan di Politeknik Penerbangan Surabaya.
8. Seluruh Sahabat, senior, junior, mentor, motivator, pelatih, dan penyemangat penulis dalam menempuh pendidikan di Politeknik Penerbangan Surabaya.

Tentunya karya tulis ini masih jauh dari sempurna. Penulis berharap semoga Tugas Akhir ini dapat memberikan manfaat khususnya bagi taruna Politeknik Penerbangan Surabaya. Atas segala kesalahan dan kata-kata yang kurang berkenan, kami memohon maaf. Saran dan kritik membangun kami harapkan demi karya yang lebih baik di masa mendatang.

Surabaya, 11 Agustus 2020

Nana Kartika Sari
NIT.30218018

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
ABSTRAK	iv
<i>ABSTRACT</i>	v
PERNYATAAN KEASLIAN DAN HAK CIPTA	vi
MOTTO	vii
KATA PENGANTAR	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xii
BAB 1 PENDAHULUAN	1
1.1 Latar belakang Masalah	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	2
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan.....	3
BAB 2 LANDASAN TEORI.....	5
2.1 Teori-teori Penunjang	5
2.1.1 Barang dan Bagasi	5
2.1.2 <i>Radio Frequency Identification</i> (RFID)	6
2.1.3 <i>Komponen Radio Frequency Identification</i> (RFID)	6
2.1.4 <i>Kegunaan Radio Frequency Identification</i> (RFID)	8
2.1.5 <i>Cara Kerja Radio Frequency Identification</i> (RFID).....	8
2.1.6 <i>Jenis Radio Frequency Identification</i> (RFID)	10
2.1.7 Wemos D1 R1.....	11
2.1.8 Arduino IDE	13
2.1.9 Buzzer	14
2.1.10Adaptor	15
2.1.11Website	15
2.1.12QoS (<i>Quality of Service</i>).....	16
2.1.12.1 Troughput.....	17
2.1.12.2 Packet Loss.....	18
2.1.12.3 Delay	19
2.1.12.4 Jitter.....	20
2.2 Kajian Penelitian Terdahulu Yang Relevan.....	21
BAB 3 METODE PENELITIAN.....	22
3.1 Desain Penelitian.....	22
3.2 Perancangan Alat	22
3.2.1 Blok Diagram	23
3.2.2 Flowchart Check In.....	24
3.2.3 Flowchart Check Out	24

3.2.4	Komponen Perangkat Keras.....	25
3.2.4.1	RFID Tag.....	25
3.3.4.2	RFID Reader RC522	25
3.3.4.3	Wemos D1 R1	25
3.3.4.4	Buzzer.....	25
3.2.5	Komponen Perangkat Lunak.....	25
3.2.5.1	Internet	25
3.2.5.2	Web Browser.....	26
3.3	Teknik Pengujian Alat	26
3.3.1	Installing Arduino IDE.....	27
3.3.2	Installing Visual Studio Code.....	29
3.3.3	Installing Node JS	32
3.4	Teknik Analisis Data.....	35
3.5	Waktu dan Tempat Penelitian	37
BAB 4 HASIL DAN PEMBAHASAN.....		38
4.1	Pengujian dan Analisis	38
4.1.1	Pengujian Perangkat Keras.....	38
4.1.1.1	Pengujian tegangan input dan output pada catu daya .	39
4.1.1.2	Pengujian Wemos D1 R1	40
4.1.1.3	Pengujian Jarak Baca RFID Reader	41
4.1.1.4	Pengujian Modul Wifi ESP8266 pada Wemos D1 R1.	43
4.1.1.5	Pengujian Buzzer.....	44
4.1.2	Pengujian Perangkat Lunak.....	44
4.1.1.1	Website RFID Baggage Security	44
4.1.1.1.1	Pengujian Proses Login.....	45
4.1.1.1.2	Pengujian User Interface (UI)	45
4.1.1.1.3	Pengujian Proses Mendeteksi Alamat URL	46
4.1.1.1.4	Pengujian Proses Input Data Bagasi saat Check In	47
4.1.1.1.5	Pengujian Proses Check Out	49
4.1.1.1.6	Pengujian Proses Tampilan setelah Check Out.....	50
4.1.3	Pengukuran <i>Quality of Service</i> (QOS)	50
4.1.3.1	Troughput	51
4.1.3.2	Packet Loss.....	51
4.1.3.3	Delay	52
4.1.3.4	Jitter	52
4.2	Pembahasan.....	53
4.3	Kekurangan dan Kelebihan Alat	53
BAB 5 PENUTUP.....		54
5.1	Kesimpulan.....	54
5.2	Saran.....	54
4.3	Kekurangan dan Kelebihan Alat	54

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR RIWAYAT HIDUP

DAFTAR GAMBAR

Gambar 2. 1 Macam – macam RFID Tag	7
Gambar 2. 2 RFID <i>Reader</i> RC522	7
Gambar 2. 3 Cara Kerja RFID	9
Gambar 2. 4 Tampilan Wemos D1 R1	12
Gambar 2. 5 Arduino IDE	13
Gambar 2. 6 Buzzer	14
Gambar 2. 7 Rangkain Adaptor	14
Gambar 3. 1 Skema Desain Alat	23
Gambar 3. 2 Flowchart Check In	24
Gambar 3. 3 Flowchart Check Out	24
Gambar 3. 4 Download Software Arduino	27
Gambar 3. 5 Tampilan Folder Arduino	27
Gambar 3. 6 Tampilan Arduino pada Local Disk C	28
Gambar 3. 7 Tampilan Awal Pada Arduino	28
Gambar 3. 8 Tampilan Pilihan Board WeMos D1 R1	29
Gambar 3. 9 Tampilan Pilihan <i>Serial Port</i>	29
Gambar 3.10 Download VScode	30
Gambar 3.11 Penginstalan VSCode	31
Gambar 3.12 Penginstalan VSCode	31
Gambar 3.13 Penginstalan VSCode	32
Gambar 3.14 Penginstalan VSCode	32
Gambar 3.15 Penginstalan VSCode	33
Gambar 3.16 Penginstalan Node JS	33
Gambar 3.17 Penginstalan Node JS	34
Gambar 3.18 Penginstalan Node JS	34
Gambar 3.19 Penginstalan Node JS	34
Gambar 3.20 Penginstalan Node JS	34
Gambar 3.21 Penginstalan Node JS	35
Gambar 3.22 Proses Compile Arduino	36
Gambar 4.1 Adaptor power supply yang diukur avometer	40
Gambar 4.2 Pengujian Wemos D1 R1	41
Gambar 4.3 Pengujian RFID Scanner RC522	42
Gambar 4.4 Pengujian Modul Wifi pada Wemos D1 R1	43

DAFTAR TABEL

Tabel 2.1 Kategori Frekuensi RFID.....	10
Tabel 2.2 Spesifikasi Wemos D1 R1	12
Tabel 2.3 Data Tiphon Troughput.....	18
Tabel 2.4 Data Tiphon Packet Loss	18
Tabel 2.5 Data Delay.....	19
Tabel 2.6 Data Jitter	20
Tabel 2.7 Tabel Perbandingan.....	21
Tabel 3.1 Waktu Penelitian	37
Tabel 4.1 Tabel Pengujian Tegangan Input Dan Output Adaptor	39
Tabel 4.2 Tabel Pengujian jarak baca RFID TAG Reader.....	42
Tabel 4.3 Tabel Pengujian Akses ke Website RFID Bagagge Security	45
Tabel 4.4 Tabel Pengujian User Interface (UI).....	45
Tabel 4.5 Tabel Pengujian Proses Mendeteksi Alamat URL Scanner.....	46
Tabel 4.6 Tabel Pengujian Input Data Bagasi saat <i>Check-In</i>	48
Tabel 4.7 Tabel Pengujian Proses Check Out.....	49
Tabel 4.8 Tabel Pengujian Tampilan setelah Check Out.....	50
Tabel 4.9 Hasil Perhitungan <i>Throughput</i>	51
Tabel 4.10 Hasil Perhitungan <i>Packet Loss</i>	51
Tabel 4.11 Hasil Perhitungan <i>Dellay</i>	52
Tabel 4.12 Hasil Perhitungan <i>Jitter</i>	52

Daftar Pustaka

- Ahyar Jadid, Zulhemi, Ardiansyah, “Rancang Bangun Sistem Absensi Perkuliahan Auito ID Berbasis RFID yang Terintegrasi dengan Database Berbasis WEB” Jurusan Teknik Elektro, Fakultas Teknik, Universitas Syiah Kuala, 2017
- Barker, D. (2016). *Web content management: Systems, features, and best practices*. “ O’Reilly Media, Inc.”
- Baskas, H. (2018). *The Trip Your Luggage Takes With You*. USA Today. <https://www.usatoday.com/story/travel/flights/2018/04/04/luggage-baggage-handling/482614002/>
- Hidajanto Djamal, “Radio Frequency (RFID) Dan Aplikasinya” Vol 1, No , 2014
- Kukuh, UcuK, Novi, “Perancangan Sistem Presensi untuk Pegawai Dengan RFID Berbasis IoT Menggunakan ESP8266” Program Studi Teknik Informatika, fakultas Komunikasi dan Informatika, Universitas Nasional, 2019
- Muhammad Aiyub,“ Rancang Bangun Sistem Monitoring dan Pengendalian Kinerja Pegawai Berbasis *Radio Fraquency Identifikation* (RFID)”Skripsi, Universitas Syiah Kuala, Fakultas Teknik Elektro, Agustus 2011.
- Nelly Khairani Daulay and M. Nur Alamsyah, “Monitoring Sistem Keamanan Pintu Menggunakan RFID dan Fingerprint Berbasis WEB dan Database”, Program Studi Rekayasa Sistem Komputer, Program Studi Sistem Informasi, Univ. Bina Insan Lubuklinggau, 2019
- V. D. Hunt, A. Puglia, and M. Puglia, RFID: A Guide to Radio Frequency Identification: Wiley Interscience, 2007. Viewed 27 Desember 2015.

MANUAL BOOK



 RFID Baggage Security

PENGUNAAN APLIKASI RFID BAGGAGE SECURITY BERBASIS WEB

DAFTAR ISI

HALAMAN COVER.....	A-1
DAFTAR ISI.....	A-2
PENJELASAN UMUM.....	A-3
PENGOPERASIAN APLIKASI.....	A-4
1. Panduan Mengakses Aplikasi	A-4
2. Panduan Login Kedalam Aplikasi	A-5
3. Panduan Penginputan Alamat URL Alat	A-6
4. Panduan Penginputan Data Bagasi Saat Check In	A-8
5. Panduan Mengcheck Out Bagasi	A-9
6. Panduan Melihat Halaman Daftar Bagasi	A-10
7. Panduan Menghapus Data Bagasi.....	A-12
8. Panduan Memodifikasi Admin	A-14
9. Panduan Memodifikasi Password Admin	A-15
10. Panduan Log Out Dari Aplikasi.....	A-15

PENJELASAN UMUM

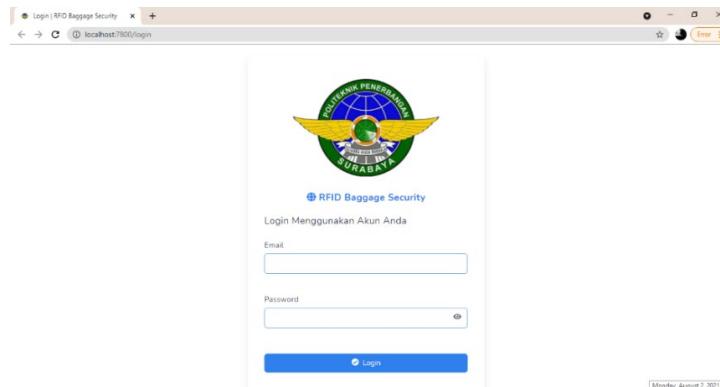
Dalam Manual Book ini akan dijelaskan mengenai tata cara serta prosedur penggunaan untuk menjalankan aplikasi RFID Baggage Security berbasis web. Aplikasi RFID Baggage Security berbasis web adalah sebuah aplikasi yang dapat memudahkan petugas bandara untuk memvalidasi bagasi ketika penumpang telah mengambil barang bagasi bawaan nya dan menuju pintu keluar sehingga petugas bandara dapat memvalidasi menggunakan Rancangan dan aplikasi RFID Baggage Security untuk memvalidasi data bagasi yang telah diinput ketika Check In ke dalam RFID Tag

Di dalam manual book aplikasi ini hanya ada prosedur dari sisi admin, karena aplikasi ini tujuannya mempermudah serta mengfisiensi waktu petugas bandara dalam memvalidasi bagasi. Pada aplikasi ini dilengkapi dengan fitur, penginputan alamat URL alat, panduan penginputan data bagasi saat check in, panduan mengcheck out bagasi, panduan melihat halaman data bagasi setelah check out, panduan menghapus data bagasi, panduan memodifikasi user admin, panduan memodifikasi password admin, panduan log out dari aplikasi.

PENGOPRASIAN APLIKASI

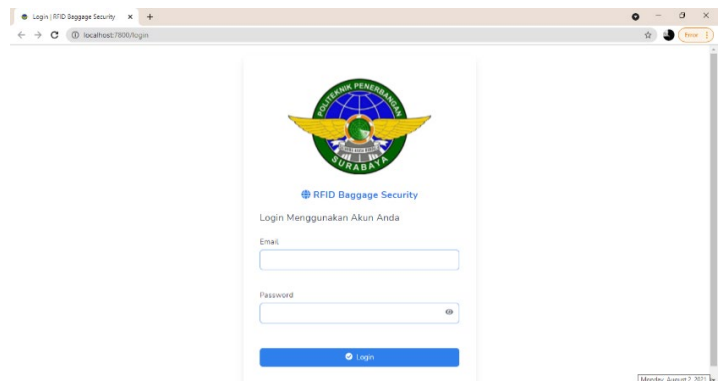
1. Panduan Mengakses Aplikasi

- a) Pastikan Wifi yang digunakan Alat dan PC ataupun smartphone yang digunakan dalam jaringan yang sama.
- b) Masuk ke halaman browser dan ketikkan alamat URL dibawah 192.168.43.11:7800/login
Gambar menetik
- c) Tekan tombol enter atau ikon search
- d) Browser akan menampilkan halaman awal aplikasi pada layar



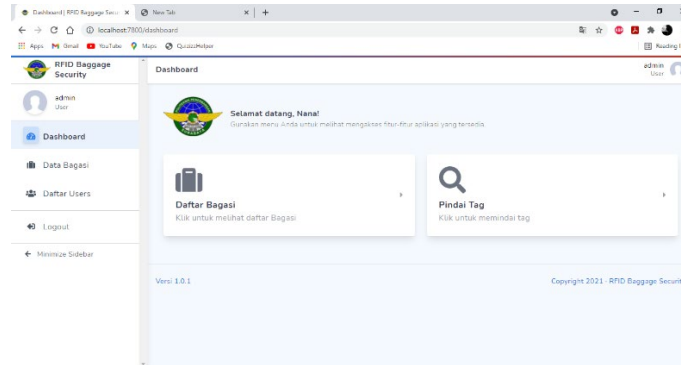
2. Panduan Login Kedalam Aplikasi

- a) Masukkan username serta password dari akun yang sudah didaftarkan disistem pada saat pembuatan. Sebagai factor keamanan aka nada opsi untuk membuat password dapat terlihat atau tidak terlihat.



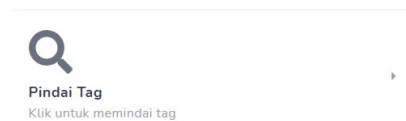
- b) Klik tombol sign in

- c) Browser akan menampilkan halaman Dashboard dari aplikasi RFID Baggage Security

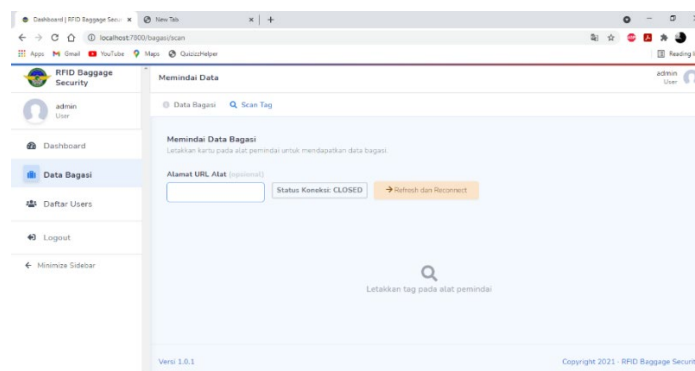


3. Panduan Menginput Alamat URL Alat

- a) Klik pindai tag pada halaman dashboard

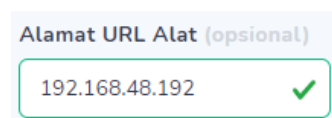


- b) Browser akan menampilkan halaman memindai data bagasi



- c) Masukan IP alat yang telah di ketahui menggunakan Net Analyzer, dengan IP alat 192.168.48.192 dan 192.168.43.106

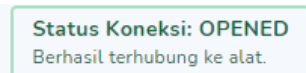
- d) Ketikkan IP alat dalam Alamat URL Alat



- e) Klik refresh dan reconnect untuk mengkoneksikan IP alat dengan server

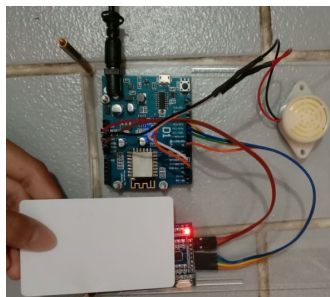


- f) Jika IP alat dan server telah terdeteksi maka status koneksi : OPENED (Berhasil terhubung ke alat)



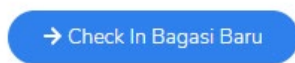
4. Panduan menginput data bagasi

- a) Setelah IP alat dapat terdeteksi dan tersambung dengan server, selanjutnya dekatkan RFID Tag dengan RFID Scanner untuk memindai tag

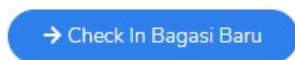


- b) Browser akan menampilkan kode unik Tag

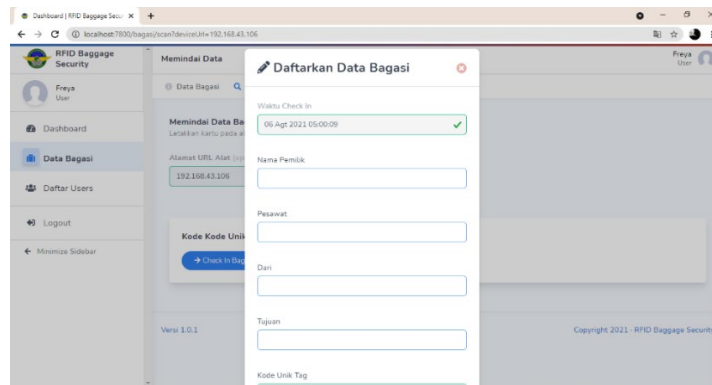
Kode Kode Unik Tag: 80260aa3



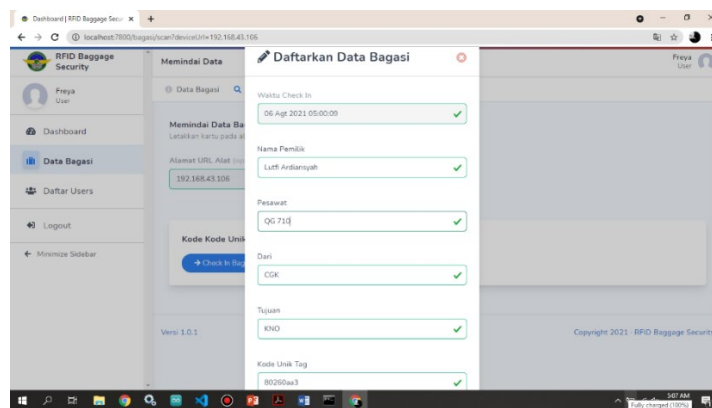
- c) Lalu klik Check In Bagasi Baru untuk mengisi formulir data bagasi



- d) Browser akan menampilkan formulir data bagasi penumpang yang didapatkan dari penumpang pesawat udara. Informasi yang harus diisi adalah nama pemilik, pesawat, dari, tujuan.



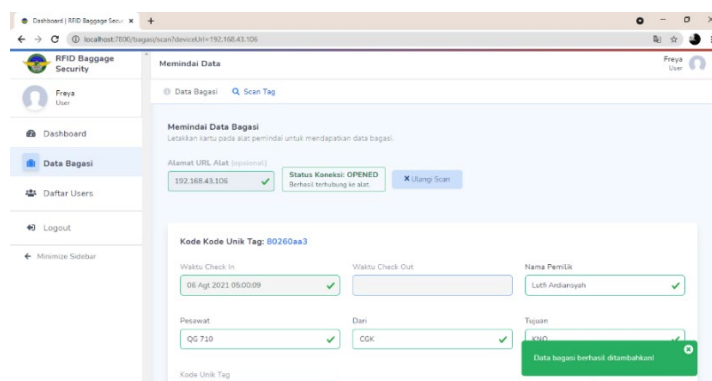
e) Minta dan masukkan informasi yang didapatkan dari penumpang pesawat udara



f) Klik tombol submit jika sudah benar



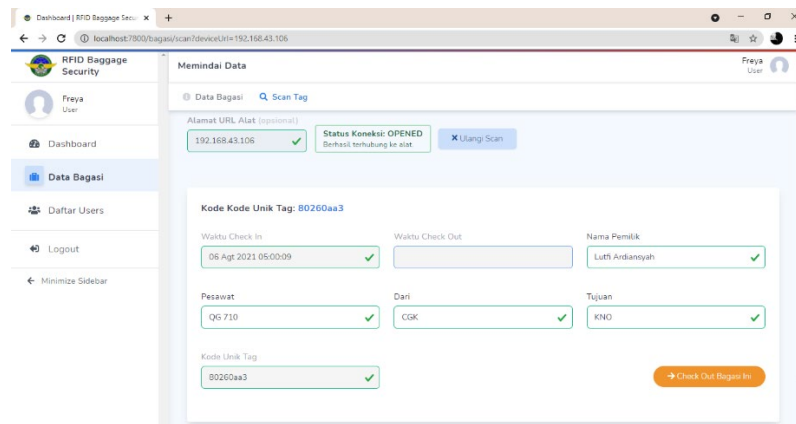
g) Browser akan menampilkan halaman data bagasi berhasil ditambahkan warna hijau pada pojok kanan bawah



h) Untuk menambahkan bagasi baru klik ulangi scan lalu lakukan step yang sama dalam menginput data bagasi penumpang

5. Panduan Mengcheck Out Bagasi

- a) Penumpang mendekatkan RFID Tag dengan RFID Scanner untuk memvalidasi data bagasi yang telah diinput ketika check in
- b) Browser akan menampilkan halaman data bagasi yang telah diinput



- c) Untuk mengcheck out bagasi klik warna orange pada pojok kanan bawah



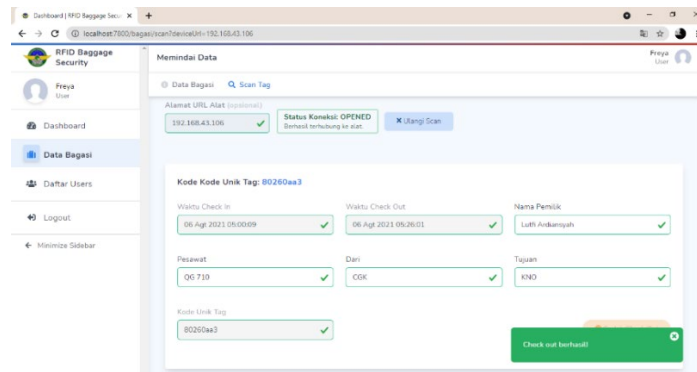
- d) Browser akan menampilkan apakah anda yakin untuk check out bagasi ini? Klik OK jika setuju, Klik Cancel Jika tidak

localhost:7800 says

Apakah Anda yakin untuk check out bagasi ini?

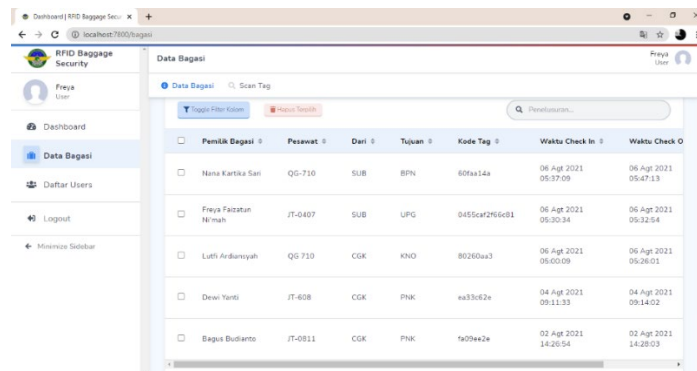


- e) Setelah menekan OK maka browser akan menampilkan waktu check out dan tulisan check out berhasil warna hijau pada pojok kanan bawah



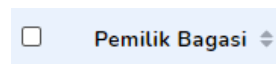
6. Panduan Melihat Halaman Daftar Bagasi

- a) Klik halaman data bagasi maka browser akan menampilkan, tabel data bagasi penumpang yang telah di check out

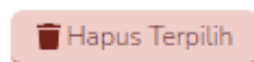


7. Panduan Menghapus Data Bagasi

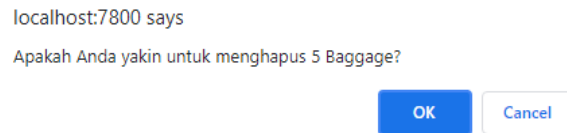
- a) Klik kotak putih disamping tulisan pemilik bagasi untuk menghapus semua data bagasi



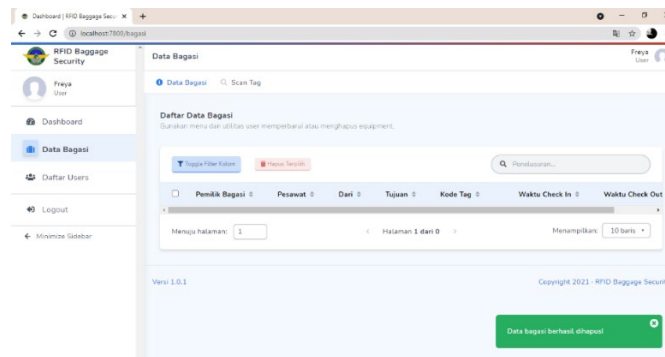
- b) Jika semua data bagasi telah centang warna hijau, klik hapus terpilih untuk menghapus semua data bagasi



- c) Browser akan menampilkan apakah anda yakin untuk menghapus jumlah data bagasi yang ditampilkan, klik warna biru OK jika ingin menghapus, klik warna putih cancel jika tidak ingin menghapus

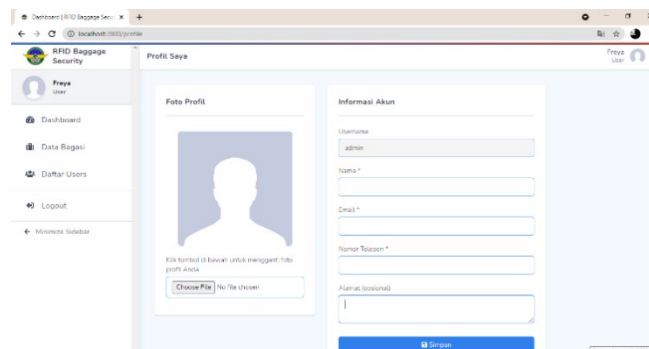


- d) Jika semua telah terhapus browser akan menampilkan data bagasi berhasil dihapus warna hijau pojok kanan bawah

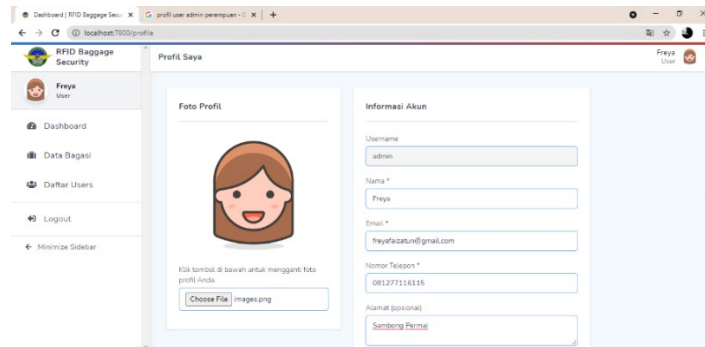


8. Panduan Memodifikasi Admin

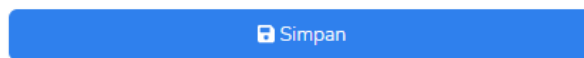
- a) Klik di kolom admin maka browser akan menampilkan informasi akun yang berisi foto profil, nama, email, nomor telepon, alamat.



- b) Isi halaman informasi akun sesuai dengan biodata admin yang bertugas saat itu

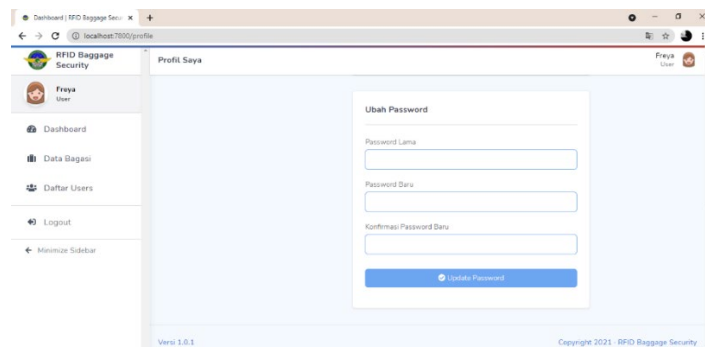


c) Klik simpan untuk menyimpan biodata admin

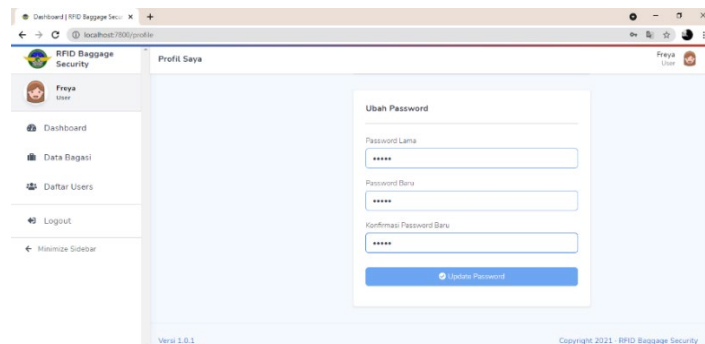


9. Panduan mengubah password

a) Klik di kolom admin maka browser akan menampilkan ubah password dibawah biodata admin




b) Isi password lama dan password baru dan konfirmasi password baru, lalu klik update password untuk merubah password



10. Panduan Log Out Dari Aplikasi

- a) Klik di kolom Log Out

 Logout

- b) Browser akan menampilkan tampilan ke halaman awal aplikasi akses untuk Login

LAMPIRAN

Lampiran A. Coding pada Arduino

```
#include <ArduinoWebsockets.h>
#include <ESP8266WiFi.h>
#include <SoftwareSerial.h>

#include <SPI.h>
#include <MFRC522.h>

#define RST_PIN      D3
#define SS_PIN       D8

MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance

int dummyState = 1;

SoftwareSerial softSerial(D9, D10); // RX, TX

const char* ssid = "wayahe";
const char* password = "wakwaw123";

using namespace websockets;

WebsocketsServer server;
WebsocketsClient client;

void onMessageCallback(WebsocketsMessage message) {
    // Serial.print("--> Got Message: ");
    // Serial.println(message.data());

    String command = message.data();
    if (command.startsWith("")) {
        Serial.print("--> Got Message: ");
        Serial.println(command);
    } else {
        Serial.print("--> Forwarding: ");
        Serial.println(command);
        softSerial.println(command);

        // dummy!
        if (command.charAt(2) == '0') {
            dummyState = 0;
        }
    }
}
```

```

        digitalWrite(14, HIGH);
    } else {
        dummyState = 1;
        digitalWrite(14, LOW);
    }
    Serial.println(dummyState);
    Serial.println(command.charAt(2));
}
}

void onEventsCall back(WebsocketsEvent event, String data) {
    if(event == WebsocketsEvent::ConnectionOpened) {
        Serial.println(" ! Connection Opened");
    } else if(event == WebsocketsEvent::ConnectionClosed) {
        Serial.println(" ! Connection Closed");
    } else if(event == WebsocketsEvent::GotPing) {
        Serial.println(" ! Got a Ping!");
    } else if(event == WebsocketsEvent::GotPong) {
        Serial.println(" ! Got a Pong!");
    }
}

void turnOnBuzzer () {
    digitalWrite(D4, HIGH);
}

void turnOffBuzzer () {
    digitalWrite(D4, LOW);
}

void beepBuzzer () {
    delay(100);
    turnOnBuzzer();
    delay(400);
    turnOffBuzzer();
}

void setup() {
    pinMode(D4, OUTPUT);
    beepBuzzer();

    pinMode(14, OUTPUT);
    digitalWrite(14, LOW);

    pinMode(LED_BUILTIN, OUTPUT);

```

```

digitalWrite(LED_BUILTIN, LOW);

Serial.begin(9600);

// Connect to wifi
WiFi.begin(ssid, password);

// Wait some time to connect to wifi
for(int i = 0; i < 15 && WiFi.status() != WL_CONNECTED; i++) {
    Serial.print(".");
    digitalWrite(LED_BUILTIN, !digitalRead(LED_BUILTIN));
    delay(1000);
}
while (WiFi.waitForConnectResult() != WL_CONNECTED) {
    Serial.println();
    Serial.println("Fail connecting!");
    scanNetworks();
    blink(); blink(); blink(); blink(); blink();
    ESP.restart();
}

Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: "); Serial.println(WiFi.localIP());
Serial.print("Signal Strength (RSSI): "); Serial.print(dBmtoPercentage(WiFi.RSSI())); Serial.println(" %");

server.listen(80);
Serial.print("Is server live? "); Serial.println(server.available(
) ? "Yes" : "No");

Serial.println();
softSerial.begin(4800);

SPI.begin();          // Init SPI bus
mfrc522.PCD_Init();   // Init MFRC522
delay(10);            // Optional delay. Some board do need more t
ime after init to be ready, see Readme
mfrc522.PCD_DumpVersionToSerial(); // Show details of PCD -
MFRC522 Card Reader details
Serial.println();

// Monitor client and start accepting initial connection
digitalWrite(LED_BUILTIN, HIGH);
Serial.println("--> Accepting connection...");

```

```

    client = server.accept();
    beepBuzzer(); beepBuzzer();
    // run callback when messages are received
    client.onMessage(onMessageCallback);
    // run callback when events are occurring
    client.onEvent(onEventsCallback);
    digitalWrite(LED_BUILTIN, LOW);
    Serial.println(" ! Accepted!");
}

long int number = 0;

unsigned long latestUpdateTime = 0;
#define UPDATE_TIME 1000
int newCardAllowed = 1;

String incomingSerialData = "";
String availableSerialData = "0,219.8,0.12,1";

void loop() {
    if (millis() - latestUpdateTime > UPDATE_TIME) {
        latestUpdateTime = millis();
        newCardAllowed = 0;
    }

    if (newCardAllowed == 1 && client.available()) {
        newCardAllowed = 2;
        beepBuzzer();

        Serial.print(F("Card UID:"));
        dump_byte_array(mfrc522.uid.uidByte, mfrc522.uid.size);
        String uid = getUidAsString(mfrc522.uid.uidByte, mfrc522.uid.size) + "\n";
        client.send(uid);
        Serial.println();

        Serial.print(F("PICC type: "));
        MFRC522::PICC_Type piccType = mfrc522.PICC_GetType(mfrc522.uid.sak);
        Serial.println(mfrc522.PICC_GetTypeName(piccType));
    }

    // Monitor client and start accepting connection if possible
    if(client.available()) {
        client.poll();
    }
}

```

```

} else {
    digitalWrite(LED_BUILTIN, HIGH);
    Serial.println(" ! Client unavailable!");

    Serial.println("--> Accepting connection...");
    client = server.accept();
    beepBuzzer(); beepBuzzer();
    // run callback when messages are received
    client.onMessage(onMessageCallback);
    // run callback when events are occurring
    client.onEvent(onEventsCallback);
    digitalWrite(LED_BUILTIN, LOW);
    Serial.println(" ! Accepted!");
}

if (newCardAllowed > 0) {
    return;
}

// Reset the loop if no new card present on the sensor/reader. This
// saves the entire process when idle.
if (!mfrc522.PICC_IsNewCardPresent()) {
    return;
}

// Select one of the cards
if (!mfrc522.PICC_ReadCardSerial()) {
    return;
}

newCardAllowed = 1;
}

```

Lampiran B. Coding pada Dashboard Visual Studio Code

```
import React, { useState, useEffect, useMemo, useCallback } from "react";
import { withApollo } from "../libs/apollo";
import AdminArea, { protectAdminArea } from "../components/AdminArea";
import Header from "../components/Header";
import { useQuery, useMutation } from "@apollo/client";
import gql from "graphql-tag";
import dayjs from "dayjs";
import FadeImage from "../components/FadeImage";
import { FontAwesomeIcon } from "@fortawesome/react-fontawesome";
import appConfig from "../app.json";
import Link from "next/link";
```

```
const QUERY = gql`
```

```
  query A {
    currentUser {
      _id
      username
      Role {
        _id
        name
        privileges
      }
    }
  }
`;
```

```
const Page = (props) => {
  const { error, loading, data, refetch } = useQuery(QUERY, {
    variables: {
```



```

    since: dayjs().subtract(1, "month").format("YYYY-MM-DD"),
  },
});
let currentUser = {};
if (data && data.currentUser) {
  currentUser = data.currentUser;
}

return (
  <AdminArea>
    <Header title="Dashboard" />

    <div className="px-8 py-8">
      <div className="flex justify-start pt-4 pb-8">
        <div className="">
          <div className="block md:hidden">
            <FadeImage
              className="inline-block"
              style={{ maxHeight: 110 }}
              src={appConfig.logo}
            />
          </div>
          <div className="md:block hidden -mt-6">
            <FadeImage
              className="inline-block"
              style={{ maxHeight: 85 }}
              src={appConfig.logo}
            />
          </div>
        </div>
        <div className="font-bold flex flex-col text-left pl-4">

```

Selamat datang!

```
<div className="font-thin text-gray-600 text-sm">
```

```
  Gunakan menu Anda untuk melihat mengakses fitur-fitur aplikasi  
  yang tersedia.
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div className="grid grid-cols-1 sm:grid-cols-2 gap-6">
```

```
  <Card>
```

```
    <Link href="/bagasi">
```

```
      <a className="flex justify-between items-center truncate hover:text-  
blue-500">
```

```
        <div className="font-bold pr-3 truncate text-lg">
```

```
          <FontAwesomeIcon
```

```
            icon="suitcase"
```

```
            size="3x"
```

```
            className="mb-2 text-gray-500"
```

```
          />
```

```
        <br />
```

```
        Daftar Bagasi
```

```
        <div className="text-gray-400 font-normal text-base">
```

```
          Klik untuk melihat daftar Bagasi
```

```
        </div>
```

```
      </div>
```

```
    <div className="text-gray-400">
```

```
      <FontAwesomeIcon icon="caret-right" />
```

```
    </div>
```

```
  </a>
```

```
</Link>
```

```
</Card>
```

```

    <Card>
      <Link href="/bagasi/scan">
        <a className="flex justify-between items-center truncate hover:text-
blue-500">
          <div className="font-bold pr-3 truncate text-lg">
            <FontAwesomeIcon
              icon="search"
              size="3x"
              className="mb-2 text-gray-500"
            />
            <br />
            Pindai Tag
          <div className="text-gray-400 font-normal text-base">
            Klik untuk memindai tag
          </div>
        </div>
        <div className="text-gray-400">
          <FontAwesomeIcon icon="caret-right" />
        </div>
      </a>
    </Link>
  </Card>
</div>
</div>
</AdminArea>
);
};

```

```
export default withApollo({ ssr: true })(protectAdminArea(Page));
```

```
const Card = ({ children, className }) => {
```

```
return (  
  <div className={`bg-white rounded shadow-lg px-6 py-6 mb-  
6 ${className}`} >  
    {children}  
  </div>  
);  
};
```

Lampiran C. Coding pada Login Visual Studio Code

```
import React, { useState, useMemo } from "react";
import Head from "next/head";
import appConfig from "../app.json";
import { withApollo } from "../libs/apollo";
import { gql, useMutation, useApolloClient } from "@apollo/client";
import FadeImage from "../components/FadeImage";
import { ShortText } from "../components/form/ShortText";
import { Password } from "../components/form/Password";
import { Button } from "../components/form/Button";
import { useNotification } from "../components/Notification";
import ms from "ms";
import redirect from "../libs/redirect";
import checkLoggedIn from "../libs/checkLoggedIn";
import cookie from "cookie";
import { FontAwesomeIcon } from "@fortawesome/react-fontawesome";

const LOGIN = gql`
  mutation logIn($username: String!, $password: String!) {
    logIn(username: $username, password: $password) {
      _id
      User {
        _id
        username
      }
      Role {
        _id
        name
        privileges
      }
      status
    }
  }
`
```

```

    }
    token
    expiresIn
  }
}
`;

```

```

const Page = (props) => {
  const notification = useNotification();
  const [internalLoading, setInternalLoading] = useState(false);
  let [credential, setCredential] = useState({
    username: "",
    password: "",
    wontExpired: false,
  });
  // console.log({ credential });

  const apolloClient = useApolloClient();
  const [login] = useMutation(LOGIN);

  return (
    <div>
      <Head>
        <title>Login | {appConfig.name}</title>
      </Head>

      <div className="md:flex w-full h-screen">
        <div className="md:w-1/2 py-5 mx-auto md:py-0 px-2 md:px-0">
          <form
            onSubmit={async (e) => {
              if (e) e.preventDefault();

```

```

setInternalLoading(true);
try {
  const result = await logIn({
    variables: {
      ...credential,
    },
  });
  const { token, expiresIn, User } = result.data.logIn;
  let maxAge = ms(expiresIn) / 1000;
  document.cookie = cookie.serialize("token", token, {
    maxAge,
    path: "/",
  });
  await apolloClient.cache.reset();
  notification.addNotification({
    title: "Login Berhasil!",
    message: "Menuju halaman dashboard...",
    level: "success",
  });
  redirect({}, "/");
} catch (err) {
  setInternalLoading(false);
  notification.handleError(err);
}
}}
className="flex flex-col justify-center items-center h-full"
>
<LoginCard.Container>
  <LoginCard.Title>
    <div className="w-full text-center px-4 md:px-0">
      <FadeImage

```

```

        className="inline-block"
        style={{ maxHeight: 200 }}
        src={appConfig.logo}
    />
</div>
<div className="text-center text-primary-500 font-bold py-4">
    <FontAwesomeIcon icon="globe" /> {appConfig.name}
</div>
Login Menggunakan Akun Anda
</LoginCard.Title>
<LoginCard.Item>
<div>
    <ShortText
        label="Email"
        required
        disabled={internalLoading}
        value={credential.username}
        onChange={(e) => {
            setCredential({
                ...credential,
                username: e.target.value,
            });
        }}
    />
    <Password
        label="Password"
        required
        disabled={internalLoading}
        value={credential.password}
        onChange={(e) => {
            setCredential({

```



```

        ...credential,
        password: e.target.value,
    });
}
/>

<Button
  loading={internalLoading}
  className="btn btn-primary w-full my-5"
  type="submit"
>
  <FontAwesomeIcon icon="check-circle" className="text-
sm" />{" "}
  &nbsp;Login
</Button>
</div>
</LoginCard.Item>
</LoginCard.Container>
{/* <div className="pt-5 w-full md:w-2/3 text-center text-sm px-3">
  Dengan melanjutkan login ke aplikasi ini, maka Anda menyetujui{" "}
  <a
    className="text-blue-400 hover:text-blue-600 hover:underline"
    target="_blank"
    href="/help/terms-and-conditions"
  >
    Syarat dan Ketentuan
  </a>{" "}
  serta{" "}
  <a
    className="text-blue-400 hover:text-blue-600 hover:underline"
    target="_blank"

```

```

        href="/help/privacy-policy"
      >
        Kebijakan Privasi
      </a>{" "}
      aplikasi.
    </div> */}
    {/* <div className="inline text-center md:text-right text-sm text-primary-
500 pt-5 w-full md:w-2/3">
      <FontAwesomeIcon icon="info-circle" /> Versi Aplikasi{" "}
      {appConfig.version}
    </div> */}
  </form>
</div>
</div>
</div>
);
};

```

```
export default withApollo({ ssr: true })(Page);
```

```

Page.getInitialProps = async (context) => {
  const { loggedInUser } = await checkLoggedIn(context.apolloClient);
  if (loggedInUser.currentUser) {
    redirect(context, "/dashboard");
  } else if (typeof loggedInUser.currentUser === "undefined") {
    return { errorCode: 500 };
  }
  return {};
};

```

```
const LoginCard = {
```

```
Container: ({ children }) => {
  return (
    <div className="shadow-lg rounded-md bg-white md:w-2/3 px-5 md:px-
8 py-5 mt-5 md:mt-10">
      {children}
    </div>
  );
},
```

```
Logo: ({ children }) => {
  return (
    <div className="text-center text-xl text-primary-400">{children}</div>
  );
},
```

```
Title: ({ children }) => {
  return (
    <div className="pt-5 pb-3 text-lg text-gunpowder-500">{children}</div>
  );
},
```

```
Item: ({ children }) => {
  return <div className="pb-4">{children}</div>;
},
};
```

Lampiran D. Coding pada LogOut Visual Studio Code

```
import React, { useEffect } from "react";
import Head from "next/head";
import { withApollo } from "../libs/apollo";
import appConfig from "../app.json";
import cookie from "cookie";
import FadeImage from "../components/FadeImage";

const DashboardPage = (props) => {
  useEffect(() => {
    setTimeout(() => {
      document.cookie = cookie.serialize("token", "", {
        maxAge: -1, // Expire the cookie immediately
        path: "/",
      });
      window.location = "/";
    }, 800);
  });

  return (
    <div>
      <div className="container">
        <Head>
          <title>Logout | {appConfig.name}</title>
        </Head>

        <div className="container">
          <div className="page-content">
            <div className="flex flex-row flex-wrap mb-8 px-4">
              <div className="w-full sm:w-1/2"></div>
            </div>
          </div>
        </div>

        <div className="mb-16">
          <div className="block w-full py-16 text-center">
            <FadeImage
              src={appConfig.logo}
              className="inline"
              style={{ maxWidth: 250 }}
            />
            <h3 className="text-lg mt-4">Anda berhasil log out ...</h3>
          </div>
        </div>
      </div>
    )
  );
};
```

```
    </div>  
  );  
};  
  
export default withApollo({ ssr: false })(DashboardPage);
```

Lampiran E. Coding Profil pada Visual Studio Code

```
import React, { useState, useEffect, useMemo, useCallback } from "react";
import { withApollo } from "../libs/apollo";
import {
  showLoadingSpinner,
  hideLoadingSpinner,
  addNotification,
} from "../components/App";
import { handleError } from "../libs/errors";
import AdminArea, { protectAdminArea } from "../components/AdminArea";
import Header from "../components/Header";
import { gql, useQuery, useMutation } from "@apollo/client";
import FadeImage from "../components/FadeImage";
import { FontAwesomeIcon } from "@fortawesome/react-fontawesome";

const QUERY = gql`
  query currentUser {
    currentUser {
      _id
      name
      username
      email
      phone
      status
      address

      roleId
      Role {
        _id
        name
      }
      pictureUrl
    }
  }
`;

const UPDATE = gql`
  mutation updateUser(
    $_id: ID!
    $name: String
    $username: String!
    $email: String
    $phone: String
  ) {
    updateUser($_id: $_id, $name: $name, $username: $username, $email: $email, $phone: $phone)
  }
`;
```

```

    $address: String
  ) {
    updateUser(
      _id: $_id
      name: $name
      username: $username
      email: $email
      phone: $phone
      address: $address
    )
  }
`;

const PROFILE = gql`
  mutation updateUser($_id: ID!, $pictureUrl: String!) {
    updateUser(_id: $_id, pictureUrl: $pictureUrl)
  }
`;

const PASSWORD = gql`
  mutation updateUserPassword(
    $_id: ID!
    $oldPassword: String!
    $newPassword: String!
  ) {
    updateUserPassword(
      _id: $_id
      oldPassword: $oldPassword
      newPassword: $newPassword
    )
  }
`;

const Page = (props) => {
  const [updateUser] = useMutation(UPDATE);
  const [updateUserPassword] = useMutation(PASSWORD);
  const [updateUserProfileImage] = useMutation(PROFILE);
  const { error, loading, data, refetch } = useQuery(QUERY);
  let currentUser = {
    Role: {},
  };
  if (data && data.currentUser) {
    currentUser = data.currentUser;
  }

  let [accountUpdateResult, setAccountUpdateResult] = useState({});

```

```

let [passwordUpdateResult, setPasswordUpdateResult] = useState({});
let [formData, setFormData] = useState({});
useEffect(() => {
  if (currentUser._id) {
    setFormData({ ...currentUser });
  }
}, [currentUser._id]);
// console.log({ formData });

return (
  <AdminArea>
    <Header title="Profil Saya" />

    <div className="px-8 py-8">
      <div className="flex flex-wrap">
        <div className="w-full md:w-1/3 pb-8">
          <div className="px-6 py-6 bg-white rounded shadow">
            <div className="font-bold">Foto Profil</div>
            <hr />
            <div className="text-center p-4">
              <FadeImage
                className="inline rounded-md"
                src={
                  formData.pictureUrl ||
                  currentUser.pictureUrl ||
                  "/images/user-dummy.jpg"
                }
              />
            </div>
          </div>
          <div className="form-group">
            <label>
              Klik tombol di bawah untuk mengganti foto profil Anda.
            </label>
            <input
              className="form-control"
              type="file"
              accept="image/*"
              onChange={(e) => {
                if (e) e.preventDefault();
                const file = e.target.files[0];
                // console.log({ file });
                if (!file.type.startsWith("image")) {
                  return handleError({
                    message: "Only image file is allowed!",
                  });
                }
              }}
            />
          </div>
        </div>
      </div>
    </div>
  </AdminArea>
)

```



```

    if (file.size > 512 * 1024) {
      return handleError({
        message: "Maximum allowed image size is 512KB!",
      });
    }
    let reader = new FileReader();

    reader.onloadend = async () => {
      showLoadingSpinner();
      try {
        // console.log("READER", reader);
        await updateUserProfileImage({
          variables: {
            _id: formData._id,
            pictureUrl: reader.result,
          },
        });
        await refetch();
        setFormData({
          ...formData,
          pictureUrl: reader.result,
        });
      } catch (err) {
        handleError(err);
      }
      hideLoadingSpinner();
    };

    reader.readAsDataURL(file);
  }}
/>
</div>
</div>
</div>

<div className="w-full md:w-1/2 md:px-6">
  <form
    className="px-6 py-6 bg-white rounded shadow"
    onSubmit={async (e) => {
      if (e) e.preventDefault();
      showLoadingSpinner();
      try {
        await updateUser({
          variables: {
            ...formData,
          },
        });
      }
    }}
  >

```

```

    });
    await refetch();
    addNotification({
      message: `Informasi akun berhasil diperbarui`,
      level: "success",
    });
    setAccountUpdateResult({
      message: `Informasi akun berhasil diperbarui`,
      level: "success",
    });
  } catch (err) {
    handleError(err);
    setAccountUpdateResult({
      message: err.message,
      level: "error",
    });
  }
  hideLoadingSpinner();
}}
>
<div className="font-bold">Informasi Akun</div>
<hr />
<div className="form-group">
  <label>Username</label>
  <input
    className="form-control"
    disabled
    value={formData.username}
    onChange={(e) => {
      if (e) e.preventDefault();
    }}
  />
</div>
<div className="form-group">
  <label>Nama *</label>
  <input
    className="form-control"
    required
    value={formData.name}
    onChange={(e) => {
      if (e) e.preventDefault();
      setFormData({
        ...formData,
        name: e.target.value,
      });
    }}
  />
</div>

```

```

/>
</div>
<div className="form-group">
  <label>Email * </label>
  <input
    className="form-control"
    type="email"
    required
    value={formData.email}
    onChange={(e) => {
      if (e) e.preventDefault();
      setFormData({
        ...formData,
        email: e.target.value,
      });
    }}
  />
</div>
<div className="form-group">
  <label>Nomor Telepon *</label>
  <input
    className="form-control"
    required
    value={formData.phone}
    onChange={(e) => {
      if (e) e.preventDefault();
      setFormData({
        ...formData,
        phone: e.target.value,
      });
    }}
  />
</div>
<div className="form-group">
  <label>Alamat (opsional)</label>
  <textarea
    className="form-control"
    value={formData.address}
    onChange={(e) => {
      if (e) e.preventDefault();
      setFormData({
        ...formData,
        address: e.target.value,
      });
    }}
  />

```

```

</div>
{accountUpdateResult?.level ? (
  <div
    className={`rounded text-sm px-4 py-3 ${
      accountUpdateResult?.level === "success"
        ? "text-green-600 bg-green-200"
        : "text-red-600 bg-red-200"
    }}
  >
    {accountUpdateResult?.message}
  </div>
) : null}
<div className="py-4">
  <button className="btn btn-primary btn-block" type="submit">
    <FontAwesomeIcon icon="save" /> Simpan
  </button>
</div>
</form>

<div className="py-4" />

<form
  className="px-6 py-6 bg-white rounded shadow"
  onSubmit={async (e) => {
    if (e) e.preventDefault();
    showLoadingSpinner();
    try {
      if (
        formData.newPassword !== formData.newPasswordConfirmation
      ) {
        throw {
          message: `Konfirmasi password tidak cocok!`,
        };
      }
      await updateUserPassword({
        variables: {
          ...formData,
        },
      });
    };
    await refetch();
    addNotification({
      message: `Password berhasil diperbarui`,
      level: "success",
    });
    setPasswordUpdateResult({
      message: `Password berhasil diperbarui`,

```

```

        level: "success",
    });
    setFormData({
        ...formData,
        oldPassword: "",
        newPassword: "",
        newPasswordConfirmation: "",
    });
} catch (err) {
    handleError(err);
    setPasswordUpdateResult({
        message: err.message,
        level: "error",
    });
}
}
hideLoadingSpinner();
}}
>
<div className="font-bold">Ubah Password</div>
<hr />
<div className="form-group">
    <label>Password Lama</label>
    <input
        className="form-control"
        required
        type="password"
        value={formData.oldPassword}
        onChange={(e) => {
            if (e) e.preventDefault();
            setFormData({
                ...formData,
                oldPassword: e.target.value,
            });
        }}
    />
</div>
<div className="form-group">
    <label>Password Baru</label>
    <input
        className="form-control"
        required
        type="password"
        value={formData.newPassword}
        onChange={(e) => {
            if (e) e.preventDefault();
            setFormData({

```

```

        ...formData,
        newPassword: e.target.value,
    });
    }}
  />
</div>
<div className="form-group">
  <label>Konfirmasi Password Baru</label>
  <input
    className="form-control"
    required
    type="password"
    value={formData.newPasswordConfirmation}
    onChange={(e) => {
      if (e) e.preventDefault();
      setFormData({
        ...formData,
        newPasswordConfirmation: e.target.value,
      });
    }}
  />
</div>
{passwordUpdateResult?.level ? (
  <div
    className={`rounded text-sm px-4 py-3 ${
      passwordUpdateResult?.level === "success"
        ? "text-green-600 bg-green-200"
        : "text-red-600 bg-red-200"
    }`}
  >
    {passwordUpdateResult?.message}
  </div>
) : null}
<div className="py-4">
  <button className="btn btn-info btn-block" type="submit">
    <FontAwesomeIcon icon="check-circle" /> Update Password
  </button>
</div>
</form>
</div>
</div>
</AdminArea>
);
};

```

```
export default withApollo({ ssr: true })(protectAdminArea(Page));
```

Lampiran F. Coding User Pada Visual Studio Code

```
import React, { useState, useEffect, useMemo, useCallback } from "react";
import { withApollo } from "../libs/apollo";
import AdminArea, { protectAdminArea } from "../components/AdminArea";
import Header from "../components/Header";
import { FontAwesomeIcon } from "@fortawesome/react-fontawesome";
import Table from "../components/Table";
import { useRouter } from "next/router";
import { useNotification } from "../components/Notification";
import { gql, useQuery, useMutation, useApolloClient } from "@apollo/client";
import { FormModal } from "../components/Modal";
import { showLoadingSpinner, hideLoadingSpinner } from "../components/App";
import FadeImage from "../components/FadeImage";
import appConfig from "../app.json";
import { ShortText } from "../components/form/ShortText";
import { Email } from "../components/form/Email";
import { PhoneNumber } from "../components/form/PhoneNumber";
import { LongText } from "../components/form/LongText";
import { SingleSelect } from "../components/form/SingleSelect";
```

```
const ALL_USERS = gql`
  query A {
    allUsers {
      _id
      username
      email
      phone
      roleId
      Role {
        _id
        name
      }
      status
      name
      address
      pictureUrl
    }
  }
`;
```

```
const ACTIVATE = gql`
  mutation B($_id: ID!) {
    activateUser(_id: $_id)
  }
`;
```



```

`;

const DEACTIVATE = gql`
  mutation C($_id: ID!) {
    deactivateUser(_id: $_id)
  }
`;

const Page = (props) => {
  const notification = useNotification();
  const router = useRouter();
  const { loading, error, data, refetch } = useQuery(ALL_USERS, {});

  let allUsers = [];
  if (data && data.allUsers) {
    allUsers = data.allUsers;
  }

  const [formModalVisible, setFormModalVisible] = useState(false);
  const [resetPasswordModalVisible, setResetPasswordModalVisible] =
    useState(false);
  const [userData, setUserData] = useState({});

  const handleInput = (key) => (e) => {
    setUserData({
      ...userData,
      [key]: e.target.value,
    });
  };

  const [activateUser] = useMutation(ACTIVATE);
  const [deactivateUser] = useMutation(DEACTIVATE);
  const apolloClient = useApolloClient();

  // console.log(allUsers);

  const columns = useMemo(() => {
    let columnDefinitions = [
      {
        Header: "Username",
        accessor: "username",
      },
      {
        Header: "Nama",
        accessor: "name",
        Cell: (props) => (

```

```

<div
  className={
    "w-full " +
    (props.row.original._deletedAt ? "text-gray-200" : "text-black")
  }
>
  {props.cell.value || props.row.original.username}
</div>
),
},
{
  Header: "Email",
  accessor: "email",
},
{
  Header: "Kontak",
  accessor: "phone",
},
{
  Header: "Role",
  accessor: "_id",
  Cell: (props) => <div>{props.row.original.Role.name}</div>,
},
{
  Header: "Status",
  accessor: "status",
  Cell: (props) => (
    <div>
      <div
        className={
          "pb-2 font-bold " +
          (props.cell.value === "Aktif"
            ? "text-green-500"
            : "text-red-500")
        }
      >
        {props.cell.value}
      </div>
      {props.cell.value === "Aktif" ? (
        <button
          className="btn btn-sm text-xs btn-danger"
          onClick={async (e) => {
            if (e) e.preventDefault();
            showLoadingSpinner();
            try {
              await deactivateUser({

```

```

        variables: {
          _id: props.row.original._id,
        },
      });
      await refetch();

      notification.addNotification({
        title: "Sukses!",
        message: `User ${props.row.original.username} saat ini telah berstatus No
n Aktif`,
        level: "success",
      });
    } catch (err) {
      notification.handleError(err);
    }
    hideLoadingSpinner();
  }}
  >
  <FontAwesomeIcon icon="ban" /> Set Tidak Aktif
</button>
): (
<button
  className="btn btn-sm text-xs btn-success"
  onClick={async (e) => {
    if (e) e.preventDefault();
    showLoadingSpinner();
    try {
      await activateUser({
        variables: {
          _id: props.row.original._id,
        },
      });
      await refetch();

      notification.addNotification({
        title: "Sukses!",
        message: `User ${props.row.original.username} saat ini telah berstatus Akt
if`,
        level: "success",
      });
    } catch (err) {
      notification.handleError(err);
    }
    hideLoadingSpinner();
  }}
  >

```

```

        <FontAwesomeIcon icon="check-circle" /> Set Aktif
      </button>
    )}
  </div>
),
},
{
  Header: "Utilitas",
  accessor: "address",
  Cell: (props) => (
    <div>
      <div className="py-1">
        <button
          type="button"
          className="btn btn-primary btn-sm"
          onClick={(e) => {
            setUserData({
              ...props.row.original,
            });
            setFormModalVisible(true);
          }}
        >
          <FontAwesomeIcon icon="pencil-alt" /> Update User
        </button>
      </div>
      <div className="py-1">
        <button
          type="button"
          className="btn btn-purple btn-sm"
          onClick={(e) => {
            setUserData({
              ...props.row.original,
              newPassword: "",
              passwordConfirmation: "",
            });
            setResetPasswordModalVisible(true);
          }}
        >
          <FontAwesomeIcon icon="key" /> Reset Password
        </button>
      </div>
    </div>
  ),
},
];

```

```

return columnDefinitions;
});

return (
  <AdminArea>
    <Header title="Daftar Users" />

    <FormModal
      title={
        <span>
          <FontAwesomeIcon icon="user" /> User
        </span>
      }
      visible={formModalVisible}
      onClose={(e) => {
        if (e) e.preventDefault();
        setFormModalVisible(false);
      }}
      onSubmit={async (e) => {
        if (e) e.preventDefault();
        showLoadingSpinner();
        try {
          if (userData._id) {
            await apolloClient.mutate({
              mutation: gql`
                mutation updateUser(
                  $_id: ID!
                  $username: String
                  $email: String
                  $phone: String
                  $name: String
                  $address: String
                  $roleId: String
                ) {
                  updateUser(
                    _id: $_id
                    name: $name
                    username: $username
                    email: $email
                    phone: $phone
                    address: $address
                    roleId: $roleId
                  )
                }
              `
            ) {
              ,
            }
          }
          variables: {

```

```

        ...userData,
    },
  });
  await refetch();
  notification.addNotification({
    title: "Sukses",
    message: `User ${userData.username} berhasil diperbarui!`,
    level: "success",
  });
} else {
  const { password, passwordConfirmation } = userData;
  if (password !== passwordConfirmation) {
    throw {
      message: "Konfirmasi password tidak cocok!",
    };
  }
}

await apolloClient.mutate({
  mutation: gql`
    mutation registerUser(
      $username: String!
      $password: String!
      $roleId: String!
      $name: String!
      $status: String!
      $email: String
      $phone: String
      $address: String
    ) {
      registerUser(
        username: $username
        password: $password
        roleId: $roleId
        name: $name
        status: $status
        email: $email
        phone: $phone
        address: $address
      ) {
        _id
      }
    }
  `,
  variables: {
    ...userData,
    status: "Aktif",
  },
});

```

```

    },
  });
  await refetch();
  notification.addNotification({
    title: "Sukses",
    message: `User ${userData.username} berhasil ditambahkan!`,
    level: "success",
  });
}
setFormModalVisible(false);
} catch (err) {
  console.log({ err });
  notification.handleError(err);
}
hideLoadingSpinner();
}}
size="lg"
>
<div className="grid grid-cols-1 sm:grid-cols-2 gap-6">
  <div>
    <ShortText
      label="Username"
      disabled={!userData._id}
      required
      value={userData.username}
      onChange={(e) => {
        if (e) e.preventDefault();
        if (!userData._id) return;
        setUserData({
          ...userData,
          username: e.target.value,
        });
      }}
    />
    <SingleSelect
      label="Role"
      value={userData.roleId}
      options={[
        {
          label: "Admin",
          value: "__ADMIN__",
        },
        {
          label: "User",
          value: "__USER__",
        },
      ]},
    />
  </div>
</div>

```

```

    ]}
    renderValue={(value) => {
      const selectedItem = [
        {
          label: "Admin",
          value: "__ADMIN__",
        },
        {
          label: "User",
          value: "__USER__",
        },
      ].find((item) => item.value === value);
      return selectedItem.label;
    }}
    onChange={(e) => {
      if (e) e.preventDefault();
      setUserData({
        ...userData,
        roleId: e.target.value,
      });
    }}
  />

```

```

    {!userData._id && (
      <div>
        <div className="form-group pb-6">
          <label>Password *</label>
          <input
            id="new-password"
            autoComplete="new-password"
            type="password"
            className="form-control"
            value={userData.password}
            onChange={(e) => {
              if (e) e.preventDefault();
              setUserData({
                ...userData,
                password: e.target.value,
              });
            }}
          />
        </div>
        <div className="form-group pb-6">
          <label>Konfirmasi Password *</label>
          <input
            id="confirm-password"

```



```

        autoComplete="confirm-password"
        type="password"
        className="form-control"
        value={userData.passwordConfirmation}
        onChange={(e) => {
            if (e) e.preventDefault();
            setUserData({
                ...userData,
                passwordConfirmation: e.target.value,
            });
        }}
    />
</div>
</div>
)}
</div>
<div>
    <ShortText
        label="Nama"
        required
        value={userData.name}
        onChange={(e) => {
            if (e) e.preventDefault();
            setUserData({
                ...userData,
                name: e.target.value,
            });
        }}
    />
    <Email
        label="Email"
        required
        value={userData.email}
        onChange={(e) => {
            if (e) e.preventDefault();
            setUserData({
                ...userData,
                email: e.target.value,
            });
        }}
    />
    <PhoneNumber
        label="Kontak / HP"
        required
        value={userData.phone}
        onChange={(e) => {

```

```

        if (e) e.preventDefault();
        setUserData({
            ...userData,
            phone: e.target.value,
        });
    }}
/>
<LongText
    label="Alamat"
    value={userData.address}
    onChange={(e) => {
        if (e) e.preventDefault();
        setUserData({
            ...userData,
            address: e.target.value,
        });
    }}
/>
</div>
</div>
</FormModal>

<FormModal
    title={
        <span>
            <FontAwesomeIcon icon="key" /> Reset Password
        </span>
    }
    size="sm"
    visible={resetPasswordModalVisible}
    onClose={(e) => {
        if (e) e.preventDefault();
        setResetPasswordModalVisible(false);
    }}
    onSubmit={async (e) => {
        if (e) e.preventDefault();
        showLoadingSpinner();
        try {
            const { newPassword, passwordConfirmation } = userData;
            if (newPassword !== passwordConfirmation) {
                throw {
                    message: "Konfirmasi password tidak cocok!",
                };
            }
        }
        await apolloClient.mutate({
            mutation: gql`

```

```

        mutation resetUserPassword($_id: ID!, $newPassword: String!) {
          resetUserPassword(_id: $_id, newPassword: $newPassword)
        }
      },
      variables: {
        _id: userData._id,
        newPassword: userData.newPassword,
      },
    });
    await refetch();
    notification.addNotification({
      title: "Sukses",
      message: `Password untuk user ${userData.username} berhasil diperbarui!`,
      level: "success",
    });
    setResetPasswordModalVisible(false);
  } catch (err) {
    console.log({ err });
    notification.handleError(err);
  }
  hideLoadingSpinner();
}}
>
<UserPasswordForm formData={userData} handleInput={handleInput} />
</FormModal>

<div className="px-8 py-8">
  <div className="flex justify-between items-end pb-8">
    <div className="flex items-center">
      <FadeImage
        className="inline-block"
        style={{ maxHeight: 60 }}
        src={appConfig.logo}
      />
    <div className="px-3 font-bold flex flex-col text-left">
      Daftar User
      <div className="font-thin text-gray-600 text-sm">
        Gunakan menu dan utilitas user memperbarui atau menghapus users.
      </div>
    </div>
  </div>
</div>
<div>
  <button
    type="button"
    className="btn btn-primary"
    onClick={(e) => {

```

```

        setUserData({});
        setFormModalVisible(true);
    }}
    >
    <FontAwesomeIcon icon="plus-circle" /> User Baru
  </button>
</div>
</div>

  <Table loading={loading} columns={columns} data={allUsers} />
</div>
</AdminArea>
);
};

```

```
export default withApollo({ ssr: true })(protectAdminArea(Page));
```

```

const UserPasswordForm = ({ formData, handleInput }) => (
  <div>
    <div className="form-group">
      <label>Username (digunakan untuk login)</label>
      <input
        className="form-control"
        required
        disabled
        value={formData.username}
        onChange={handleInput("username")}
      />
    </div>
    <div className="form-group">
      <label>Password *</label>
      <input
        type="password"
        className="form-control"
        value={formData.newPassword}
        onChange={handleInput("newPassword")}
      />
    </div>
    <div className="form-group">
      <label>Konfirmasi Password *</label>
      <input
        type="password"
        className="form-control"
        value={formData.passwordConfirmation}
        onChange={handleInput("passwordConfirmation")}
      />
    </div>
  </div>
);

```

```
</div>  
</div>  

```

DAFTAR RIWAYAT HIDUP



NANA KARTIKA SARI, Lahir di Larantuka pada tanggal 08 November 2000. Merupakan anak pertama dari empat bersaudara dari pasangan Bapak Bagus Budianto dan Ibu Dewi Yanti. Bertempat tinggal di Sambong Permai Blok H No.1 RT 02, RW 5, Kec. Jombang Kab. Jombang Jawa Timur 61414. Memulai pendidikan Sekolah Dasar di SD Jombang 2, lulus tahun 2012. Melanjutkan pendidikan Sekolah Menengah Pertama di SMPN 1 Tembelang, lulus tahun 2015. Melanjutkan Sekolah Menengah Atas di SMA Negeri Ploso, lulus tahun 2018. Selanjutnya pada bulan September 2018 diterima sebagai Taruna Politeknik Penerbangan Surabaya pada Program Studi Diploma 3 Teknik Navigasi Udara Angkatan XI sampai dengan saat ini. Selama masa pendidikan di Politeknik Penerbangan Surabaya telah mengikuti *On The Job Training (OJT)* di Perum LPPNPI Cabang Jakarta Air Traffic Service Center (JATSC). Aktif sebagai salah satu anggota Gita Swara Buana XI.