

**RANCANG BANGUN SISTEM INFORMASI
PEMINJAMAN INVENTARIS LABORATORIUM
PRODI TEKNOLOGI INFORMASI**

TUGAS AKHIR

Diajukan Oleh :

Ghazanul Fikri Khaironi

NIM. 200705018

Mahasiswa Fakultas Sains dan Teknologi

Program Studi Teknologi Informasi



**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI AR-RANIRY**

BANDA ACEH

2024/1446

**RANCANG BANGUN SISTEM INFORMASI PEMINJAMAN
INVENTARIS LABORATORIUM PRODI TEKNOLOGI
INFORMASI**

TUGAS AKHIR

Diajukan Kepada Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh
Sebagai Salah Satu Beban Studi Memperoleh Gelar Sarjana (S1)
dalam Ilmu/Prodi Teknologi Informasi

Oleh:

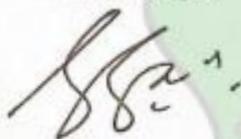
HAZANUL FIKRI KHAIRONI

NIM. 200705018

**Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi**

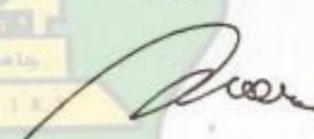
Disetujui Untuk di Munaqasyahkan Oleh:

Pembimbing I,



Khairan AR, M. Kom
NIP.198607042014031001

Pembimbing II,



Hendri Ahmadian, S.Si., M.I.M
NIP.198301042014031002

Mengetahui,

Ketua Program Studi Teknologi Informasi



Matahayati, M.T
NIP.198301272015032003

LEMBAR PENGESAHAN

**RANCANG BANGUN SISTEM INFORMASI PEMINJAMAN
INVENTARIS LABORATORIUM PRODI TEKNOLOGI
INFORMASI**

TUGAS AKHIR

Telah Diuji Oleh Panitia Ujian Munaqasah Tugas Akhir
Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh Dan Dinyatakan Lulus
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S-1)
Pada Prodi Teknologi Informasi

Pada Hari/Tanggal : Jumat, 19 Juli 2024
13 Muharram 1446 H

Panitia Ujian Munaqasah Tugas Akhir:

Ketua,



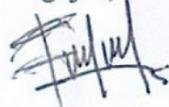
Khairan AR, M.Kom
NIP. 198607042014031001

Sekretaris,



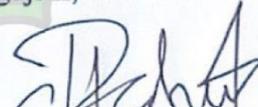
Hendri Ahmadian, S.Si., M.I.M
NIP. 198301042014031002

Penguji I,



Fathiah, M.Eng
NIP. 198606152019032010

Penguji II,



Rahmat Musfekar, M.Kom
NIP. 198909132020121015

Mengetahui:

Dekan Fakultas Sains dan Teknologi
UIN Ar-Raniry Banda Aceh,



Dr. Ir. M. Dirhamsyah, M.T., M.P.U
NIP. 196210021988111001

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang Bertanda tanga dibawah ini :

Nama : Ghazanul Fikri Khaironi
NIM : 200705018
Program Studi : Teknologi Informasi
Fakultas : Sains dan Teknologi
Judul Tugas akhir : Rancang Bangun Sistem Informasi Peminjaman Inventaris Laboratorium Prodi Teknologi Informasi

Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya :

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah tugas akhir orang lain;
3. Tidak menggunakan tugas akhir orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik tugas akhir;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri tugas akhir ini dan mampu bertanggung jawab atas tugas akhir ini.

Bila dikemudian hari ada tuntutan dari pihak lain atas tugas akhir saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memang dibuktikan bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.

Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 30 Juli 2024
Yang Menyatakan



(Ghazanul Fikri Khaironi)

ABSTRAK

Nama : Ghazanul Fikri Khaironi
NIM : 200705018
Program Studi : Teknologi Informasi
Judul Tugas akhir : Rancang Bangun Sistem Informasi Peminjaman Inventaris Laboratorium Prodi Teknologi Informasi
Tanggal Sidang : 19 Juli 2024
Jumlah Halaman : 153 Halaman
Pembimbing I : Khairan AR, M. Kom
Pembimbing II : Hendri Ahmadian, S.Si., M.I.M

Penelitian ini bertujuan untuk merancang dan mengembangkan sebuah aplikasi berbasis web untuk peminjaman inventaris. Aplikasi ini dilengkapi dengan beberapa fitur utama seperti *authentication*, *add*, *delete*, *edit*, *monitoring*, *download*, serta fitur peminjaman dan pengembalian inventaris. Aplikasi ini telah diuji menggunakan metode pengujian *black box* untuk memastikan bahwa setiap fitur berfungsi sesuai dengan yang diharapkan. Pengujian dilakukan oleh operator program studi Teknologi Informasi, dan hasilnya menunjukkan bahwa semua fitur aplikasi berjalan dengan baik tanpa ada kendal. Dengan aplikasi ini, diharapkan dapat membantu dalam mengelola inventaris laboratorium secara lebih baik, serta dapat mengurangi kesalahan dan kehilangan data yang mungkin terjadi dalam proses peminjaman dan pengembalian barang. Selain itu, aplikasi ini juga dapat memberikan kemudahan dalam pemantauan status inventaris secara real-time, sehingga memudahkan pihak pengelola dalam melakukan pengawasan dan pengelolaan inventaris. Hasil penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam pengelolaan inventaris, khususnya di lingkungan akademik dan laboratorium.

Kata Kunci :Inventaris Laboratorium, Aplikasi Web, Black Box.

ABSTRACT

Name : Ghazanul Fikri Khaironi
NIM : 200705018
Department : Information Technology
Title : Rancang Bangun Sistem Informasi Peminjaman Inventaris Laboratorium Prodi Teknologi Informasi
Date : 19 Juli 2024
Number of Pages : 153 Pages
Supervisor I : Khairan AR, M. Kom
Supervisor II : Hendri Ahmadian, S.Si., M.I.M

This research aims to design and develop a web-based application for inventory lending. This application is equipped with several main features such as authentication, add, delete, edit, monitoring, and download, as well as inventory borrowing and returning features. This application has been tested using the black box testing method to ensure each feature functions as expected. The Information Technology study program operator carried out testing, and the results showed that all application features ran well without any problems. With this application, it is hoped that it can help in managing laboratory inventory better, and can reduce errors and data loss that may occur in the process of borrowing and returning goods. Apart from that, this application can also provide convenience in monitoring inventory status in real-time, making it easier for managers to monitor and manage inventory. It is hoped that the results of this research can make a significant contribution to inventory management, especially in academic and laboratory environments.

Keywords: Laboratory Inventory, Web Application, Black Box.

KATA PENGANTAR

Bismillahirrahmanirrahim

Segala puji bagi Allah, Tuhan dari semesta alam. Shalawat dan salam semoga selalu tercurah kepada junjungan kita Nabi Muhammad Shallallahu ‘Alaihi Wa sallaam, keluarga dan sahabatnya. Alhamdulillah dengan rahmat Allah yang Maha Rahman dan yang Maha Rahim, sehingga penulis dapat menyelesaikan tugas akhir dengan judul “Rancang Bangun Sistem Informasi Peminjaman Inventaris Laboratorium Prodi Teknologi Informasi” ini. Tugas akhir ini merupakan salah satu syarat untuk menyelesaikan program studi Strata satu Teknologi Informasi pada Fakultas Sains dan Teknologi di Universitas Islam Negeri Ar-Raniry Banda Aceh.

Ucapan terima kasih penulis sampaikan kepada berbagai pihak yang menjadi sebab dari mereka penulis belajar, mendapatkan ilmu, mendapatkan dukungan, serta mendapatkan hal yang bermanfaat lainnya sehingga penulis sampai pada titik menyelesaikan tugas akhir ini. Terutama dalam konteks ini penulis sampaikan kepada :

1. Orang tua terhebat yang penulis cintai Bapak Khaidir, Ibu Rohani dan keluarga besar penulis yang selalu memberikan dukungan, dan doa yang tak ternilai harganya selama ini. Hanya Allah yang mampu membalas kasih sayang mereka yang tak terhingga, semoga selalu Allah limpahkan rahmat kepada mereka dan mendapatkan ridha serta cinta dari-Nya.
2. Bapak Khairan Ar, M.Kom dan Bapak Hedri Ahmadian, S.Si., M.I.M selaku pembimbing yang selalu bersedia meluangkan waktu dan fikirannya untuk membimbing penulis demi kesempurnaan tugas akhir ini. Ketua dan Sekretaris Program Studi Teknologi Informasi, Malahayati, M.T. Dan Khairan Ar, M.Kom, serta Bapak dan Ibu dosen Program Studi Teknologi Informasi yang telah memberikan ilmu pengetahuan dalam bidang Teknologi Informasi kepada penulis sehingga penulis mampu menyelesaikan tugas akhir ini.
3. Penasehat Akademik, Dr. Ir. M. Dirhamsyah, M.T. yang telah membimbing dan memberikan saran selama masa perkuliahan.

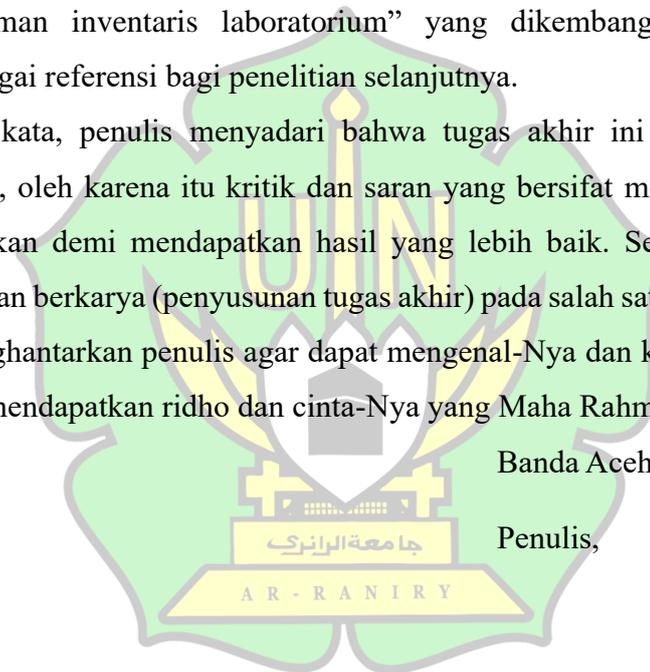
4. Staf Prodi Ibu Cut Ida Rahmadiana S,Si. yang telah membantu membantu penulis dalam hal pengurusan administrasi dan surat-surat untuk keperluan penyelesaian tugas akhir.
5. Teman-teman mahasiswa Prodi Teknologi Informasi dan teman teman penulis di luar perkuliahan yang selalu memberikan dukungan dan perspektif pemikiran yang membantu peneliti dalam membuat tugas akhir ini.

Penulis berharap hasil tugas akhir ini dapat berguna dalam mengelola inventaris laboratorium pada Prodi Teknologi Infomasi dengan adanya “aplikasi web peminjaman inventaris laboratorium” yang dikembangkan. Dan dapat dijadikan sebagai referensi bagi penelitian selanjutnya.

Akhir kata, penulis menyadari bahwa tugas akhir ini masih jauh dari kesempurnaan, oleh karena itu kritik dan saran yang bersifat membangun sangat penulis harapkan demi mendapatkan hasil yang lebih baik. Semoga perjalanan mempelajari dan berkarya (penyusunan tugas akhir) pada salah satu ilmu milik-Nya ini dapat menghantarkan penulis agar dapat mengenal-Nya dan kekasih-Nya lebih banyak serta mendapatkan ridho dan cinta-Nya yang Maha Rahman dan Rahim.

Banda Aceh, 30 Juli 2024

Penulis,



Ghazanul Fikri Khaironi

DAFTAR ISI

LEMBAR PERSETUJUAN TUGAS AKHIR	i
LEMBAR PENGESAHAN.....	ii
LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR	iii
ABSTRAK.....	iv
ABSTRACK.....	v
KATA PENGANTAR	vi
DAFTAR ISI	vi
DAFTAR GAMBAR	viii
DAFTAR TABEL	xi
DAFTAR LAMPIRAN.....	xii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan Penelitian.....	3
1.4. Batasan Masalah.....	3
1.5. Manfaat Penelitian	3
BAB II LANDASAN TEORI.....	4
2.1. Personal Extreme Programming (PXP)	4
2.2. Unified Modeling Language (UML).....	7
2.3. Sistem Informasi.....	8
2.4. Inventaris.....	8
2.5. Figma	9
2.6. Aplikasi	9
2.7. Web	9
2.8. Aplikasi Web	10
2.9. Flutter	11
2.10. Dart.....	11
2.11. Firebase	12
2.12. Visual Studio Code.....	12
2.13. Penelitian Terdahulu.....	13
BAB III METODE PENELITIAN	16

3.1.	Pengumpulan Data	17
3.2.	<i>Requirements</i>	17
3.3.	<i>Planning</i>	21
3.4.	<i>Iteration Inialization</i>	22
3.5.	<i>Design</i>	22
3.2.	Implementasi	41
3.3.	<i>System Testing</i>	42
3.4.	Tempat penelitian	52
BAB IV HASIL DAN PEMBAHASAN		53
4.1	Design	53
4.3.1	Use Case Diagram.....	53
4.3.2	Database Schema	54
4.2	Activity Diagram.....	57
4.3	Implementasi	74
4.3.1	Fitur Authentication	74
4.3.2	Fitur Add	77
4.3.3	Fitur Delete	82
4.3.4	Fitur Edit	87
4.3.5	Fitur Monitoring.....	88
4.3.6	Fitur Download	90
4.3.7	Fitur Peminjaman.....	93
4.3.8	Fitur Pengembalian	102
4.4	System Testing	114
BAB V KESIMPULAN DAN SARAN		125
5.1	Kesimpulan	125
5.2	Saran.....	125
DAFTAR PUSTAKA.....		126
LAMPIRAN.....		129

DAFTAR GAMBAR

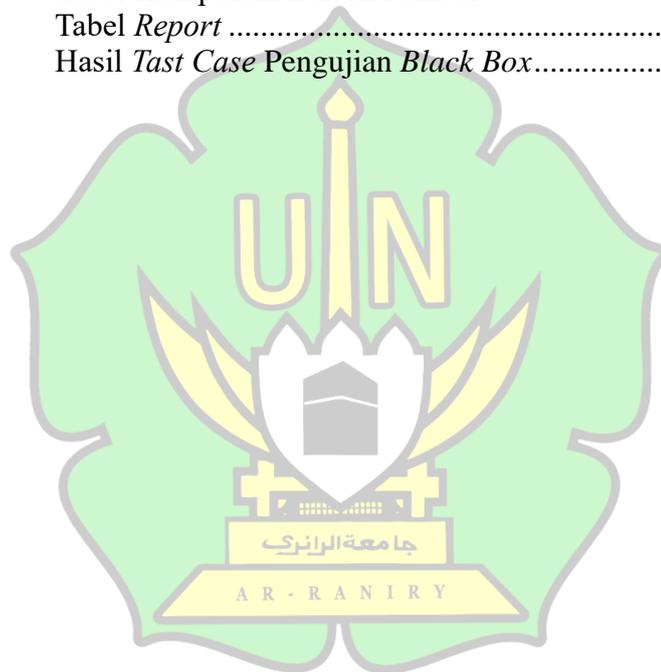
Gambar 2. 1	Tahapan Proses PXP (Nuzul Firdaus, 2023)	7
Gambar 3. 1	Tahapan Penelitian.....	16
Gambar 3. 2	Desain <i>Sign In Screen</i>	23
Gambar 3. 3	Desain <i>Logout</i>	23
Gambar 3. 4	Desain Tambah Kategori Barang.....	24
Gambar 3. 5	Desain Tambah Barang Berkode	25
Gambar 3. 6	Desain Tambah Barang Tidak Berkode	25
Gambar 3. 7	Desain Hapus Kategori Barang	26
Gambar 3. 8	Desain Hapus Barang Berkode.....	27
Gambar 3. 9	Desain Hapus Barang Tidak Berkode	27
Gambar 3. 10	Desain Edit Kategori	28
Gambar 3. 11	Desain <i>Monitoring Screen</i>	28
Gambar 3. 12	Desain <i>Download Screen</i>	29
Gambar 3. 13	Desain Peminjaman <i>Screen Input Data Web</i>	30
Gambar 3. 14	Desain Peminjaman <i>Screen Input Data Mobile</i>	30
Gambar 3. 15	Desain Peminjaman <i>Screen Memilih Kategori Barang Web</i>	31
Gambar 3. 16	Desain Peminjaman <i>Screen Memilih Barang Mobile</i>	31
Gambar 3. 17	Desain Peminjaman <i>Screen Memilih Barang Berkode Web</i>	32
Gambar 3. 18	Desain Peminjaman <i>Screen Memilih Barang Berkode Mobile</i>	32
Gambar 3. 19	Desain Peminjaman <i>Screen Memilih Barang Tidak Berkode Web</i>	33
Gambar 3. 20	Desain Peminjaman <i>Screen Memilih Barang Tidak Berkode Mobile</i>	33
Gambar 3. 21	Desain Peminjaman <i>Screen Konfirmasi Pinjaman Web</i>	34
Gambar 3. 22	Desain Peminjaman <i>Screen Konfirmasi Pinjaman Mobile</i>	34
Gambar 3. 23	Desain Pengembalian <i>Screen Memilih Barang Web</i>	35
Gambar 3. 24	Desain Pengembalian <i>Screen Memilih Barang Mobile</i>	35
Gambar 3. 25	Desain Pengembalian <i>Screen List Kategori Barang Web</i>	36
Gambar 3. 26	Desain Pengembalian <i>Screen List Kategori Barang Mobile</i>	36
Gambar 3. 27	Desain Pengembalian <i>Screen Barang Berkode Web</i>	37
Gambar 3. 28	Desain Pengembalian <i>Screen Barang Berkode Mobile</i>	37
Gambar 3. 29	Desain Pengembalian <i>Screen Barang Tidak Berkode Web</i>	38
Gambar 3. 30	Desain Pengembalian <i>Screen Barang Tidak Berkode Mobile</i>	38

Gambar 3. 31	Desain Pengembalian <i>Screen</i> Memilih Barang Berkode <i>Web</i>	39
Gambar 3. 32	Desain Pengembalian <i>Screen</i> Memilih Barang Berkode <i>Mobile</i>	39
Gambar 3. 33	Desain Pengembalian <i>Screen</i> Memilih Barang Tidak Berkode <i>Web</i>	40
Gambar 3. 34	Desain Pengembalian <i>Screen</i> Memilih Barang Tidak Berkode <i>Mobile</i>	40
Gambar 3. 35	Desain Pengembalian <i>Screen</i> Konfirmasi Pengembalian <i>Web</i>	41
Gambar 3. 36	Desain Pengembalian <i>Screen</i> Konfirmasi Pengembalian <i>Mobile</i>	41
Gambar 4. 1	<i>Use Case Diagram</i>	53
Gambar 4. 2	<i>Database Schema</i>	54
Gambar 4. 3	<i>Activity Diagram Sign Up</i>	58
Gambar 4. 4	<i>Activity Diagram logout</i>	59
Gambar 4. 5	<i>Activity Diagram</i> Tambah Kategori Barang	60
Gambar 4. 6	<i>Activity Diagram</i> Tambah Barang	62
Gambar 4. 7	<i>Activity Diagram</i> Hapus Kategori Barang.....	64
Gambar 4. 8	<i>Activity Diagram</i> Hapus Barang.....	66
Gambar 4. 9	<i>Activity Diagram</i> Edit Barang	67
Gambar 4. 10	<i>Activity Diagram Monitoring</i>	68
Gambar 4. 11	<i>Activity Diagram Download Database</i>	69
Gambar 4. 12	<i>Activity Diagram</i> Peminjaman	71
Gambar 4. 13	<i>Activity Diagram</i> peminjaman.....	73
Gambar 4. 14	Hasil <i>Sign In Screen</i>	74
Gambar 4. 15	Hasil <i>Logout</i> Fitur	75
Gambar 4. 16	<i>Backend</i> Fitur <i>Sign In</i>	76
Gambar 4. 17	<i>Backend</i> Fitur <i>Logout</i>	76
Gambar 4. 18	Hasil Tambah kategori Barang <i>Screen</i>	77
Gambar 4. 19	Hasil Tambah Barang Berkode <i>Screen</i>	78
Gambar 4. 20	Hasil Tambah Barang Tidak Berkode <i>Screen</i>	79
Gambar 4. 21	<i>Backend</i> Fitur Tambah Kategori Barang	80
Gambar 4. 22	<i>Backend</i> Fitur Tambah Barang Berkode.....	81
Gambar 4. 23	<i>Backend</i> Fitur Tambah Barang Tidak Berkode	81
Gambar 4. 24	Hasil Fitur Hapus Kategori Barang	82
Gambar 4. 25	Hasil Hapus Barang Berkode <i>Screen</i>	83
Gambar 4. 26	Hasil Hapus Barang Tidak Berkode <i>Screen</i>	84
Gambar 4. 27	<i>Backend</i> Fitur Hapus Kategori Barang.....	85
Gambar 4. 28	<i>Backend</i> Fitur Hapus Barang Berkode	86
Gambar 4. 29	<i>Backend</i> Fitur Hapus Barang Tidak Berkode	86
Gambar 4. 30	Hasil Edit kategori Barang <i>Screen</i>	87
Gambar 4. 31	<i>Backend</i> Fitur Edit.....	88
Gambar 4. 32	Hasil <i>Monitoring Screen</i>	89
Gambar 4. 33	<i>Backend</i> Fitur <i>Monitoring</i>	90
Gambar 4. 34	Hasil Fitur <i>Download</i>	91

Gambar 4. 35	<i>Backend</i> Fitur <i>Download</i>	92
Gambar 4. 36	<i>Google Sheet</i> Pengambilan Data Dari <i>Database</i>	93
Gambar 4. 37	Hasil Menginput Data Peminjaman <i>Screen Web</i>	93
Gambar 4. 38	Hasil Menginput Data Peminjaman <i>Screen Mobile</i>	94
Gambar 4. 39	Hasil Memilih Kategori Barang Peminjaman <i>Screen Web</i>	95
Gambar 4. 40	Hasil Memilih Kategori Barang Peminjaman <i>Screen Mobile</i>	95
Gambar 4. 41	Hasil Memilih Barang Berkode Ketika Peminjaman <i>Screen Web</i>	96
Gambar 4. 42	Hasil Memilih Barang Berkode Ketika Peminjaman <i>Screen Mobile</i>	97
Gambar 4. 43	Hasil Memilih Barang Tidak Berkode Ketika Peminjaman <i>Screen Web</i>	98
Gambar 4. 44	Hasil Memilih Barang Tidak Berkode Ketika Peminjaman <i>Screen Mobile</i>	98
Gambar 4. 45	Hasil Konfirmasi Peminjaman <i>Screen Web</i>	99
Gambar 4. 46	Hasil Konfirmasi Peminjaman <i>Screen Mobile</i>	100
Gambar 4. 47	<i>Backend</i> Fitur Pinjam Barang Berkode	101
Gambar 4. 48	<i>Backend</i> Fitur Pinjam Barang Tidak Berkode	102
Gambar 4. 49	Hasil Memilih Barang Pengembalian <i>Screen Web</i>	103
Gambar 4. 50	Hasil Memilih Barang Pengembalian <i>Screen Mobile</i>	103
Gambar 4. 51	List kategori Barang Pengembalian <i>Screen Web</i>	104
Gambar 4. 52	List kategori Barang Pengembalian <i>Screen Mobile</i>	105
Gambar 4. 53	Hasil Memilih Barang Berkode Ketika Pengembalian <i>Screen Web</i>	106
Gambar 4. 54	Hasil Memilih Barang Berkode Ketika Pengembalian <i>Screen Mobile</i>	106
Gambar 4. 55	Hasil Memilih Barang Tidak Berkode Ketika Pengembalian <i>Screen Web</i>	107
Gambar 4. 56	Hasil Memilih Barang Tidak Berkode Ketika Pengembalian <i>Screen Mobile</i>	108
Gambar 4. 57	Hasil Pengembalian <i>Screen Web</i> Ketika Telah Memilih Barang Berkode	109
Gambar 4. 58	Hasil Pengembalian <i>Screen Mobile</i> Ketika Telah Memilih Barang Berkode	109
Gambar 4. 59	Hasil Pengembalian <i>Screen Web</i> Ketika Telah Memilih Barang Tidak Berkode	110
Gambar 4. 60	Hasil Pengembalian <i>Screen Mobile</i> Ketika Telah Memilih Barang Tidak Berkode	111
Gambar 4. 61	Hasil Konfirmasi Pengembalian <i>Screen Web</i>	112
Gambar 4. 62	Hasil Konfirmasi Pengembalian <i>Screen Mobile</i>	112
Gambar 4. 63	<i>Backend</i> Penggalan Kode Fitur Kembalikan Barang Berkode	113
Gambar 4. 64	<i>Backend</i> Penggalan Kode Fitur Kembalikan Barang Tidak Berkode	114

DAFTAR TABEL

Tabel 3. 1	Fitur Admin	17
Tabel 3. 2	Fitur Mahasiswa	18
Tabel 3. 3	<i>User Story</i> Admin.....	19
Tabel 3. 4	<i>User Story</i> Mahasiswa.....	20
Tabel 3. 5	Perangkat Lunak.....	20
Tabel 3. 6	Perangkat Keras.....	21
Tabel 3. 7	Ranking Prioritas	21
Tabel 3. 8	<i>Test Case</i> Pengujian Black Box	43
Tabel 4. 1	Tabel Barang.....	54
Tabel 4. 2	Tabel <i>User</i>	55
Tabel 4. 3	Tabel BarangPinjaman.....	55
Tabel 4. 4	Tabel KumpulanFileIconGambar	56
Tabel 4. 5	Tabel <i>Report</i>	56
Tabel 4. 6	Hasil <i>Tast Case</i> Pengujian <i>Black Box</i>	115



DAFTAR LAMPIRAN

Lampiran 1	<i>Back End</i>	129
Lampiran 2	<i>Front End Admin File Barang</i>	130
Lampiran 3	<i>Front End Admin File Riwayat</i>	131
Lampiran 4	<i>Front End Admin File tambah barang</i>	132
Lampiran 5	<i>Front End Mahasiswa File Barang</i>	133
Lampiran 6	<i>Front End Mahasiswa File Home</i>	134
Lampiran 7	<i>Front End Mahasiswa File Peminjaman</i>	135
Lampiran 8	<i>Front End Mahasiswa File Pengembalian</i>	136
Lampiran 9	<i>Front End Mobile Mahasiswa File Home</i>	137
Lampiran 10	<i>Front End Mobile Mahasiswa File Peminjaman</i>	138
Lampiran 11	<i>Front End Mobile Mahasiswa File Pengembalian</i>	139



BAB I

PENDAHULUAN

1.1. Latar Belakang

Inventaris adalah penyimpanan berbagai jenis barang, baik dalam bentuk bahan mentah maupun barang jadi, yang ditempatkan untuk penggunaan di masa mendatang atau dalam periode waktu yang spesifik (Rangga Bakti et al., 2023). Inventaris juga diterapkan pada pengelolaan laboratorium, laboratorium sendiri merupakan fasilitas yang mendukung proses pembelajaran untuk meningkatkan keterampilan dan pengetahuan mahasiswa. Kualitas laboratorium dapat dinilai dari kelengkapan sumber daya pembelajaran yang disediakan. Oleh karena itu, pengelolaan inventaris merupakan elemen kunci dalam mengontrol peralatan, peminjaman dan komponen yang ada di laboratorium (Aminah et al., 2019). Prodi Teknologi Informasi (TI) juga memiliki laboratorium yang sudah mengelola inventarisnya, namun masih menggunakan cara manual dan belum terkelola dengan baik.

Pencatatan data peminjaman inventaris laboratorium pada Prodi TI dilakukan dengan cara manual, yaitu dengan mencatat semua data yang diperlukan pada saat peminjaman dan pengembalian pada sebuah buku yang telah disediakan oleh prodi. Sehingga proses pendataan alat-alat inventaris laboratorium prodi teknologi informasi belum terkelola dengan baik. Hal ini berdampak pada mudahnya kehilangan inventaris Prodi TI.

Sehingga Prodi TI membutuhkan peningkatan dalam pengolahan data terintegrasi untuk mengelola peminjaman inventaris laboratorium. Selain itu, perlu juga meningkatkan sistem penyimpanan data yang dapat disimpan dengan aman dan terorganisir dengan baik. Dengan menciptakan aplikasi *web* peminjaman inventaris laboratorium, diharapkan dapat membantu staf prodi TI dalam meningkatkan kinerjanya, sehingga dapat bekerja lebih efektif dan efisien dalam proses pendataannya.

Aplikasi *web* peminjaman inventaris laboratorium yang akan dibangun dalam penelitian ini akan menggunakan metode *Personal Extreme Programming* (PXP). Metode PXP adalah pilihan yang sesuai dalam konteks penelitian ini, karena penelitian ini hanya melibatkan satu programmer tanpa ketergantungan pada sebuah tim. Kecepatan perubahan dalam kebutuhan klien terhadap sistem yang sedang dibangun juga membuat metode PXP menjadi pilihan yang lebih berharga, karena PXP selalu siap untuk mengakomodasi perubahan yang mungkin terjadi selama proses pengembangan (Evanansy, 2022).

Pembuatan aplikasi *web* peminjaman inventaris laboratorium ini memiliki tujuan yaitu merekap data peminjaman inventaris alat laboratorium yang telah terdata, yang nantinya akan di arsipkan. Meskipun pendataan ini sudah dilakukan namun pendataan ini masih dilakukan dengan cara manual. Sehingga perlu dilakukan pembangunan aplikasi *web* peminjaman inventaris laboratorium untuk memfasilitasi pendataan secara otomatis.

Berdasarkan paparan diatas, peneliti pada penelitian ini akan membangun sebuah sistem berbasis aplikasi *web* yang akan diterapkan pada Prodi TI. Sistem peminjaman alat inventaris laboratorium ini akan dapat diakses oleh Mahasiswa dan Admin. Implementasi sistem ini pada Prodi TI akan memberikan manfaat besar dengan memudahkan staf prodi dalam pengelolaan data-data peminjaman alat inventaris laboratorium. Dengan sistem ini, diharapkan Prodi TI dapat meningkatkan efisiensi dalam manajemen inventori dan mengatasi keterbatasan yang ada pada pendekatan manual sebelumnya.

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dijabarkan diatas, maka dapat dirumuskan masalahnya sebagai berikut :

1. Bagaimana merancang aplikasi peminjaman alat inventaris laboratorium berbasis *web* menggunakan metode *Personal Extreme Programming* (PXP).
2. Bagaimana melakukan pengujian menggunakan *black box testing* pada aplikasi *web* peminjaman inventaris laboratorium?

1.3. Tujuan Penelitian

Tujuan pada penelitian ini meliputi

1. Merancang aplikasi peminjaman alat inventaris laboratorium berbasis *web* pada Prodi Teknologi Informasi UIN Ar-Raniry.
2. Melakukan pengujian aplikasi *web* peminjaman inventaris laboratorium menggunakan *black box testing*.

1.4. Batasan Masalah

Batasan masalah pada penelitian ini meliputi

1. Aplikasi berbasis *web* ini berfokus pada pendataan dan pencatatan, serta peminjaman alat inventaris laboratorium.
2. Aplikasi berbasis *web* ini dapat diakses oleh admin dan mahasiswa Prodi Teknologi Informasi UIN Ar-Raniry.

1.5. Manfaat Penelitian

1. Manfaat Teoritis

Hasil dari penelitian ini bisa menjadi bahan rujukan untuk pengembangan *web* aplikasi lainnya

2. Manfaat Praktis

Dapat digunakan oleh Prodi Teknologi Informasi untuk pendataan dan pencatatan peminjaman alat inventaris laboratorium

BAB II

LANDASAN TEORI

2.1. Personal Extreme Programming (PXP)

Personal Extreme Programming (PXP) adalah suatu pendekatan pengembangan perangkat lunak yang dirancang khusus untuk pengembang tunggal. PXP adalah hasil dari penggabungan antara dua pendekatan yang berbeda, yaitu *Personal Software Process* (PSP) dan *Extreme Programming* (XP), yang menghasilkan metode yang sangat berguna untuk pengembangan individu. PXP mengadopsi sejumlah prinsip dasar dari XP, yang telah terbukti efektif dalam pengembangan perangkat lunak, dan mengintegrasikannya ke dalam kerangka kerja PSP. Salah satu keuntungan utama PXP adalah pengurangan jumlah dokumen dan data yang harus diisi dalam formulir, sehingga mengurangi beban administratif dan memungkinkan pengembang untuk lebih fokus pada aspek teknis dan kreatif dari pengembangan perangkat lunak. (Wibowo et al., 2019a)

Dalam prakteknya, PXP menggunakan iterasi berbasis sprint untuk mengatur pengembangan. Sprint-sprint ini telah direncanakan sebelumnya dan memungkinkan pengembang untuk lebih responsif terhadap perubahan kebutuhan. Karena seringkali sulit untuk memprediksi semua persyaratan proyek sejak awal, PXP memungkinkan pengambilan keputusan yang lebih cepat terkait perubahan yang mungkin muncul sepanjang perjalanan pengembangan. Melalui pendekatan ini, PXP menciptakan lingkungan yang mendukung pengembang tunggal dalam menghadapi perubahan kebutuhan dengan lebih baik, memastikan bahwa perangkat lunak yang dihasilkan tetap relevan dan berkualitas tinggi. Dengan menggabungkan aspek-aspek terbaik dari PSP dan XP, PXP membantu pengembang individu dalam mencapai produktivitas dan hasil yang lebih baik dalam pengembangan perangkat lunak. (Wibowo et al., 2019b)

Pada skripsi (Nuzul Firdaus, 2023) PXP memiliki tahapan proses yang terdiri dari *requirements, planning, iteration inialization, design, implementation, system testing* dan *retrospective* berikut merupakan penjelasannya.

1. *Requirements*

Tahap requirements merupakan fase dimana informasi terkait kebutuhan aplikasi ditentukan, seperti kebutuhan fungsional dan non-fungsional dari aplikasi yang dikembangkan.

2. *Planning*

Dalam fase perencanaan, pengembang merencanakan serangkaian cerita pengguna (*user story*) beserta detail-detail tugas yang harus diselesaikan dalam setiap iterasi, dengan merujuk pada kebutuhan yang telah ditentukan sebelumnya. Selanjutnya, menentukan tingkat prioritas dan memperkirakan waktu yang diperlukan untuk menyelesaikan *user story* dan tugas-tugas yang terkait.

3. *Iteration Inialization*

Inisialisasi iterasi merupakan langkah awal dari suatu iterasi atau sprint. Pada awal iterasi, *user story* dipilih untuk dikerjakan berdasarkan prioritas yang telah ditentukan pada tahap perencanaan, dan menjadi fokus utama selama iterasi tersebut. Durasi iterasi dapat bervariasi, biasanya antara satu hingga tiga minggu, tergantung pada kompleksitas *user story* yang sedang dikerjakan. Di akhir setiap iterasi, hasilnya dapat berupa rilis versi dari *user story* atau fitur yang telah berhasil dibuat.

4. *Design*

Dalam langkah ini, pengembang mengambil langkah untuk merancang desain dan pemodelan aplikasi yang sesuai dengan *user story* yang sedang dikerjakan dalam suatu iterasi. Desain dan pemodelan ini dilakukan dengan fokus memenuhi kebutuhan yang terkait dengan *user story* yang sedang dikerjakan, tanpa berusaha mengantisipasi kebutuhan yang mungkin muncul di masa mendatang.

5. *Implementation*

Ini adalah fase di mana pengembang membuat kode program untuk mengembangkan aplikasi sesuai dengan desain yang telah dipersiapkan, dan kemudian melakukan pengujian terhadap kode tersebut. Tahap ini terdiri dari tiga langkah, yaitu pengujian unit, penulisan kode, dan perbaikan kode. Tahap implementasi diselesaikan ketika kode program berjalan tanpa kesalahan (error) dan pengujian unit berhasil.

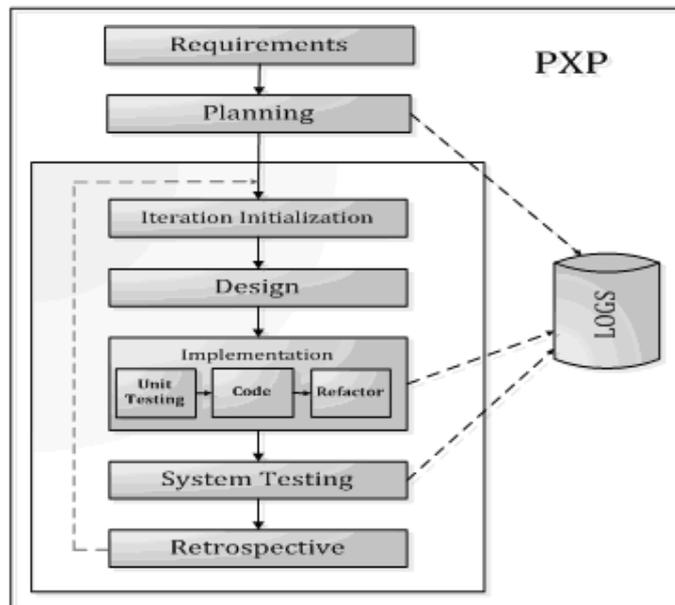
6. *System Testing*

Dalam tahapan ini, perancang harus melakukan verifikasi terhadap kesesuaian solusi yang telah diterapkan dengan persyaratan awal proyek. Semua kekurangan atau cacat yang ditemukan harus dicatat dengan cermat, dan langkah-langkah perbaikan yang sesuai harus diambil untuk memastikan bahwa solusi tersebut memenuhi standar dan persyaratan yang telah ditetapkan (Rinaldy, 2019).

7. *Retrospective*

Tahap *retrospektif* menandai penutupan sebuah iterasi dalam proses pengembangan. Pada tahap ini, data dan analisis yang telah terkumpul selama fase-fase sebelumnya digunakan untuk evaluasi. Perancang harus memeriksa apakah estimasi waktu yang digunakan untuk tugas-tugas tersebut sesuai dengan kenyataan, dan jika tidak, maka harus mengidentifikasi potensi penyebab keterlambatan untuk memastikan bahwa perkiraan di masa depan lebih akurat.

Selama tahap *retrospektif*, perancang juga melakukan analisis mendalam terhadap semua usulan perbaikan proses yang mungkin. Fase ini bisa berakhir dengan merilis produk yang telah dikerjakan, atau, dalam konteks pengembangan proyek yang lebih besar, menjadi landasan untuk memulai iterasi baru dengan kembali ke fase "*Inisialisasi Iterasi*". Tahap *retrospektif* juga dapat menjadi penanda akhir pengembangan proyek saat semua persyaratan pengguna untuk perangkat lunak telah terpenuhi (Rinaldy, 2019).



Gambar 2. 1 Tahapan Proses PXP (Nuzul Firdaus, 2023)

2.2. Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah sebuah alat yang digunakan untuk memvisualisasikan dan mendokumentasikan hasil analisis dan desain sistem secara visual. Ini juga merupakan seperangkat konvensi pemodelan yang digunakan untuk menggambarkan sistem perangkat lunak yang terkait dengan objek. UML adalah bahasa pemodelan yang digunakan untuk sistem atau perangkat lunak yang berdasarkan paradigma berorientasi objek.

Konsep dasar dalam UML mencakup klasifikasi struktural, perilaku dinamis, dan manajemen model, yang dapat dianggap sebagai istilah utama yang muncul saat membuat diagram, dan view merupakan kategori dari diagram tersebut. UML mendefinisikan berbagai jenis diagram, seperti diagram use case, diagram kelas, diagram statechart, diagram aktivitas, diagram urutan, diagram kolaborasi, diagram komponen, dan diagram deployment (Ronal et al., 2022).

2.3. Sistem Informasi

Menurut Laudon pada Skripsi(Angga Iswara, 2019) sistem informasi adalah suatu rangkaian komponen yang beroperasi secara terpadu, saling berinteraksi, dan bertujuan untuk mengumpulkan, mengolah, menyimpan, serta menyebarkan data dan informasi yang krusial dalam konteks organisasi. Tujuan utama dari sistem informasi adalah untuk memberikan dukungan yang diperlukan dalam pengambilan keputusan yang cerdas, koordinasi yang efisien, pengendalian yang tepat, analisis yang mendalam terhadap permasalahan yang ada, dan visualisasi yang memudahkan pemahaman dalam organisasi tersebut. Dengan demikian, sistem informasi menjadi pilar utama dalam mengoptimalkan efisiensi dan efektivitas operasional suatu organisasi, serta memberikan wawasan yang berharga untuk mencapai tujuan organisasi tersebut.

Sedangkan menurut Tata Sutabri pada skripsi (karmila, 2019) sistem informasi dalam suatu organisasi adalah suatu sistem yang dirancang untuk menyatukan kebutuhan pengolahan transaksi harian yang mendukung fungsi manajerial organisasi, dalam rangka mendukung pelaksanaan strategi organisasi. Sistem ini bertujuan untuk menghasilkan laporan-laporan yang diperlukan oleh pihak eksternal tertentu.

2.4. Inventaris

Inventaris barang adalah suatu proses komprehensif yang melibatkan pencatatan, pendataan, serta pengelolaan data persediaan barang yang dimiliki oleh suatu instansi atau organisasi. Kegiatan ini tidak hanya mencakup pengidentifikasian dan dokumentasi barang, tetapi juga melibatkan perencanaan, pengawasan, dan pengendalian terhadap persediaan tersebut untuk memastikan ketersediaan yang optimal, efisiensi operasional, dan pemenuhan kebutuhan organisasi. Dengan demikian, inventaris barang merupakan aspek penting dalam manajemen aset dan persediaan yang berperan dalam menjaga kelancaran operasi dan efektivitas suatu instansi (Wirna et al., 2022).

2.5. Figma

Figma merupakan sebuah perangkat lunak desain berbasis *web* yang dapat diakses melalui sistem operasi Windows, Linux, dan Mac. Selain berfungsi untuk membuat desain antarmuka (UI) dan prototipe, Figma juga menyediakan fitur kolaborasi tim dan komentar (Vallendito, 2020).

2.6. Aplikasi

Menurut Hartono pada skripsi (Juliasnyah, 2020) aplikasi adalah penggunaan dalam suatu *computer*, instruksi (*instruction*) atau pernyataan (*statement*) yang di susun sedemikian rupa sehingga *computer* dapat memproses *input* menjadi *output*.

Sedangkan menurut Hasan Abdurrahman dan Asep Ririh Riswayan pada jurnal (Widarma & Rahayu, 2018) aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut.

2.7. Web

Menurut Hamzah Hartono pada skripsi (Pangestu, 2022) *website* adalah kumpulan halaman situs dan dokumen pendukungnya, seperti gambar, rekaman, dan dokumen terkomputerisasi lainnya, yang disimpan di server *web*. Sebagian besar dari konten ini dapat diakses melalui *web*. Dalam konteks yang lebih luas, *website* adalah kumpulan pengelolaan dan dokumen yang mengandung berbagai jenis data dan memiliki beragam kapasitas, termasuk kemampuan untuk menampilkan informasi dan layanan tertentu.

Sementara itu, menurut Rajak & Muharto pada skripsi (Pangestu, 2022), *website* adalah sebuah sistem yang terdiri dari berbagai halaman yang saling terhubung, dan aksesnya tidak terbatas pada jenis komputer, *smartphone*, atau sistem operasi tertentu. Sebuah *website* juga dapat dianggap sebagai alamat atau *uniform resource locator* (URL) yang digunakan untuk menyimpan data dan informasi yang dapat diakses oleh berbagai pengguna dari berbagai lokasi.

Berdasarkan penjelasan di atas dapat disimpulkan, *website* adalah suatu entitas online yang terdiri dari halaman-halaman yang saling terhubung dan dapat diakses melalui berbagai perangkat dan sistem operasi, yang menyediakan akses kepada berbagai jenis informasi dan data.

2.8. Aplikasi Web

Aplikasi *web* adalah sistem perangkat lunak yang didasarkan pada teknologi dan standar yang ditetapkan oleh *World Wide Web Consortium* (W3C). Aplikasi ini memiliki peran penting dalam menyediakan sumber daya, seperti konten dan layanan, kepada pengguna melalui antarmuka yang dapat diakses melalui peramban *web*. Aplikasi *web* memberikan akses terhadap berbagai informasi dan fungsi melalui tampilan yang terintegrasi dengan peramban *web* pengguna.

Menurut Lowe dan Pressman (Muhammad, 2018) aplikasi *web* sebenarnya adalah perangkat lunak. Hal ini disebabkan karena aplikasi *web* berisi kumpulan instruksi dan data yang bekerja bersama untuk menyediakan informasi dan fungsionalitas kepada pengguna. Oleh karena itu, aplikasi *web* adalah bagian integral dari dunia perangkat lunak, dan disiplin yang digunakan dalam pengembangan aplikasi *web* sejalan dengan disiplin yang digunakan dalam pengembangan perangkat lunak pada umumnya.

Berdasarkan penjelasan di atas dapat disimpulkan, Aplikasi *web* adalah jenis perangkat lunak yang dirancang untuk dapat diakses oleh pengguna melalui browser ketika terhubung ke jaringan internet. Aplikasi ini memanfaatkan koneksi jaringan untuk memberikan layanan dan fungsionalitas kepada pengguna, yang dapat diakses dengan mudah melalui browser. Aplikasi berbasis *web* didasarkan pada berbagai bahasa pemrograman, seperti HTML, JavaScript, CSS, dan bahasa pemrograman lainnya.

2.9. Flutter

Flutter merupakan *Software Development Kit* (SDK) aplikasi *mobile* yang berfungsi untuk membuat aplikasi baik Android dan iOS dari satu *codebase* dengan performa yang tinggi. Tujuannya adalah untuk memungkinkan pengembang menyediakan aplikasi berkinerja tinggi di seluruh platform. Flutter menggunakan Bahasa pemrograman C, C++, Skia dan Drat. Flutter memiliki dua hal penting di antaranya:

1. Perangkat Pengembang Perangkat Lunak (SDK) set perangkat yang akan membantu membuat aplikasi. Ini menggabungkan peralatan untuk kumpulan kode menjadi kode native (kode untuk iOS dan Android).
2. Perpustakaan antar muka pengguna berdasarkan widget (Framework) Mengumpulkan Komponen UI (tombol, input teks, penggeser, dan sebagainya.) dapat disesuaikan dengan kebutuhan aplikasi.

Widget adalah blok struktur penting dari UI aplikasi flutter. Setiap widget adalah bagian UI yang bukan dapat diubah. Berbeda dengan kerangka kerja lainnya yang memisahkan tampilan, pengontrol tampilan, tata letak, dan properti lainnya (Juliasnyah, 2020).

2.10. Dart

Dart adalah bahasa pemrograman yang direncanakan oleh Lars Bak dan Kasper Lund. Bahasa pemrograman ini dibuat sebagai Bahasa pemrograman aplikasi yang dapat dengan mudah dipelajari dan menyebar. Bahasa pemrograman besutan Google ini bisa dimanfaatkan untuk bermacam-macam platform termasuk *web*, aplikasi *smartphone*, server.

Dart juga bisa digunakan untuk membuat aplikasi dari basis kode tunggal menjadi aplikasi Android atau iOS. Dart dapat digunakan tanpa syarat oleh para Developer, karena bahasa ini diluncurkan secara open-source oleh Google di bawah izin BSD (Juliasnyah, 2020).

2.11. Firebase

Firebase adalah organisasi spesialis *cloud* dengan *back-end* sebagai *servis* yang berada di San Fransisco, California. Firebase menciptakan berbagai produk untuk pengembangan aplikasi *mobile* dan *web*. Pada tahun 2011, Andrew Lee dan James Tamplin mendirikan Firebase, yang memulai debutnya pada tahun 2012 dengan database cloud real-time dan gratis. Produk utama Firebase adalah database dengan API yang memungkinkan developer menyimpan dan menyinkronkan data di multiple client. Organisasi ini diakuisisi oleh Google pada bulan Oktober 2014.

Firebase adalah *backend* sebagai database penyedia layanan dan layanan real-time. Aplikasi yang memungkinkan developer membuat API untuk disinkronkan dengan klien yang berbeda dan disimpan di cloud. Selain itu, developer dapat melindungi database dengan memanfaatkan server Firebase dan mengikuti aturan Firebase yang ada. Firebase menyediakan hosting file statis dengan fasilitas untuk hosting file CDN dan SNL (Sonita & Fardianitama, 2018).

2.12. Visual Studio Code

Visual Studio Code (VS Code) adalah sebuah teks editor yang sangat populer, yang dikembangkan oleh Microsoft. Salah satu keunggulan utama dari VS Code adalah keberadaannya sebagai teks editor ringan dan handal yang tersedia di berbagai sistem operasi, termasuk Linux, Mac, dan Windows. Dengan dukungan untuk sistem operasi multiplatform, VS Code memungkinkan pengembang dari berbagai latar belakang untuk mengakses dan menggunakan alat ini.

VS Code secara langsung mendukung beberapa bahasa pemrograman, seperti JavaScript, Typescript, dan Node.js. Namun, daya tarik utamanya adalah kemampuan untuk mendukung bahasa pemrograman lainnya melalui penggunaan plugin yang dapat diunduh dan diinstal melalui marketplace Visual Studio Code. Hal ini memungkinkan pengguna untuk menyesuaikan pengalaman pengkodean mereka sesuai dengan kebutuhan dan preferensi mereka.

Selain itu, VS Code memiliki beragam fitur yang membuatnya sangat berguna dalam pengembangan perangkat lunak, seperti Intellisense untuk memberikan saran kode, integrasi Git untuk manajemen kode sumber, dan

kemampuan debugging. Kelebihan lainnya adalah fleksibilitas dalam menambahkan fitur melalui ekstensi, yang memungkinkan pengguna untuk menyesuaikan alat ini sesuai dengan tugas dan kebutuhan proyek mereka.

Yang membuat VS Code semakin menonjol adalah komitmen Microsoft untuk melakukan pembaruan versi secara berkala setiap bulan. Ini berarti bahwa alat ini selalu diperbarui dengan fitur-fitur terbaru, perbaikan, dan peningkatan kinerja. Hal ini membedakan VS Code dari teks editor lain yang mungkin tidak mendapatkan pembaruan sejauh ini.

Lebih dari itu, VS Code adalah perangkat *open source*, sehingga pengembang dari seluruh dunia dapat melihat dan berkontribusi pada pengembangannya. Kode sumbernya dapat ditemukan di GitHub, yang memungkinkan komunitas pengembang untuk berpartisipasi dalam pengembangan dan perbaikan VS Code ke depannya. Keterbukaan ini menjadikan VS Code favorit di kalangan para pengembang aplikasi, yang dapat berkontribusi pada perbaikan dan pengembangan alat ini, memastikan bahwa itu tetap relevan dan memenuhi kebutuhan pengguna (Costa Pinaria et al., n.d.).

2.13. Penelitian Terdahulu

Dibutuhkannya referensi atau penelitian terkait bagi peneliti yang ingin menulis guna untuk menghindari duplikasi dan pelanggaran hak cipta. Penelitian terdahulu berfungsi untuk bahan tolak ukur dan acuan. Tentunya untuk melihat perbandingan dengan penelitian ini, maka dalam landasan teori ini penulis mencantumkan hasil-hasil penelitian terdahulu sebagai berikut:

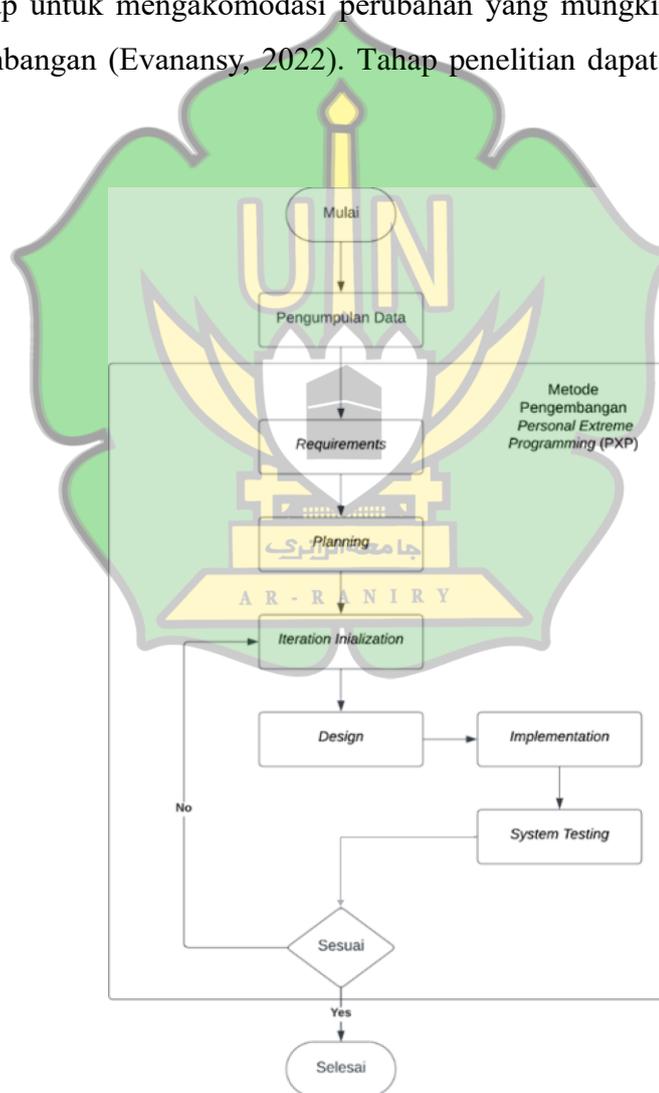
No	Nama Peneliti	Judul	Tahun	Deskripsi
1.	(Rezandy, 2018)	Rancang Bangun Aplikasi Peminjaman Alat Berbasis <i>Web</i> Pada Program Studi Pspt Di Smkn 1 Surabaya	2018	Penelitian ini memungkinkan Tersedianya Aplikasi yang mampu melakukan transaksi peminjaman dan pengembalian alat dan aplikasi mampu menghasilkan laporan ketersediaan alat dan laporan riwayat peminjaman alat. Dengan menggunakan PHP sebagai <i>Backend</i> , HTMP sebagai <i>Frontend</i> dan menggunakan <i>System Development Life Cycle</i> (SDLC) sebagai metode penelitiannya.
2.	(Widhiatmojo, 2018)	Sistem Informasi Peminjaman Di Biro Aset Dan Rumah Tangga Universitas Muhammadiyah Magelang	2018	Penelitian ini memungkinkan Tersedianya sistem informasi peminjaman di Biro ART yang dikembangkan dapat mengelola peminjaman asset berdasarkan <i>priority</i> yang telah ditentukan dan peminjaman dapat dilakukan sesuai dengan <i>privilege</i> masing-masing user peminjam. Dengan menggunakan yii framework dengan bahasa pemograman PHP dan menggunakan MySQL sebagai <i>databasenya</i> .

No	Nama Peneliti	Judul	Tahun	Deskripsi
3.	(Maloky, 2022)	Rancang Bangun Sistem Peminjaman Peralatan Di Laboratorium Telekomunikasi Berbasis Iot Menggunakan Aplikasi Android	2022	Penelitian ini memungkinkan Tersedianya Aplikasi android Lab Asisstant dibuat menggunakan aplikasi android studio yang terdiri dari 10 halaman yaitu halaman login, halaman register, halaman utama, halaman semester, halaman mata kuliah, halaman jobsheet, halaman barang, halaman QR Code, halaman notifikasi peminjaman, dan halaman riwayat peminjaman. Semua halaman menggunakan file berformat xml dan java untuk mengatur design dan activity-nya pada masing-masing halaman. Dan menggunakan Firebase untuk mengupdate data secara real time.

BAB III

METODE PENELITIAN

Metode yang digunakan pada rancang bangun aplikasi *web* peminjaman inventaris laboratorium adalah *Personal Extreme Programming* (PXP). Metode PXP ini merupakan pilihan yang sesuai dalam konteks penelitian ini, karena penelitian ini hanya melibatkan satu programmer tanpa ketergantungan pada sebuah tim. Kecepatan perubahan dalam kebutuhan klien terhadap sistem yang sedang dibangun juga membuat metode PXP menjadi pilihan yang lebih berharga, karena PXP selalu siap untuk mengakomodasi perubahan yang mungkin terjadi selama proses pengembangan (Evanansy, 2022). Tahap penelitian dapat di lihat sebagai berikut:



Gambar 3. 1 Tahapan Penelitian

3.1. Pengumpulan Data

Di lakukannya pengumpulan data untuk mendapatkan Informasi yang diperlukan untuk proses pengembangan. Pengumpulan data dilakukan dengan cara wawancara informal kepada staf Prodi Teknologi Informasi. Pengambilan data ini bertujuan untuk mengetahui inventaris laboratorium apa saja yang ada dan fitur-fitur yang di perlukan dalam aplikasi *web* peminjaman inventaris laboratorium.

3.2. Requirements

Tahapan pertama dalam Personal Extreame Programming (PXP) adalah menentukan kebutuhan atau persyaratan dalam pengembangan aplikasi seperti kebutuhan fungsional dan non-fungsional.

1. Kebutuhan Fungsional

Kebutuhan fungsional menggambarkan fungsi-fungsi dan fitur-fitur yang ada pada *web* aplikasi yang ingin dibangun. Kebutuhan fitur tersebut ditulis dalam bentuk *user story* dengan format “Sebagai <jenis pengguna>, saya ingin <melakukan tindakan tertentu> sehingga<mendapatkan manfaat dari tindakan tersebut>”. Pada tabel di bawah ini dapat dilihat penjelasan fitur-fitur yang akan dibuat. Fitur-fitur yang terdapat pada aplikasi *web* peminjaman inventaris laboratorium ini dibagi menjadi dua, yaitu user admin dan user mahasiswa. Admin tidak bisa mengakses fitur yang ada pada mahasiswa begitupun sebaliknya. Berikut merupakan fitur-fitur *user* admin yang dapat dilihat pada Tabel 3.1.

Tabel 3. 1 Fitur Admin

No	Fitur	Deskripsi
1.	<i>Authentication</i>	Fitur ini akan menangani kebutuhan proses otentikasi <i>admin</i> seperti <i>login</i> dan <i>logout</i> .

No	Fitur	Deskripsi
2.	<i>Add</i>	Fitur ini akan menangani proses penambahan kategori barang dan penambahan barang yang ada di dalam kategori.
3.	<i>Delete</i>	Fitur ini akan menangani proses penghapusan kategori barang dan penghapusan barang yang ada dalam kategori barang
4.	Edit	Fitur ini akan menangani tentang proses pengeditan kategori barang seperti nama barang, <i>icon</i> barang, dan tipe barang
5.	<i>Monitoring</i>	Fitur ini akan menangani tentang riwayat peminjaman dan pengembalian dari mahasiswa.
6.	<i>Download</i>	Fitur ini akan menangani tentang <i>Download database</i> peminjaman dan pengembalian barang dari mahasiswa.

Sumber: Data diolah (2023)

Berikut merupakan fitur-fitur *user* mahasiswa yang dapat dilihat pada Tabel 3.2.

Tabel 3. 2 Fitur Mahasiswa

No	Fitur	Deskripsi
1.	Peminjaman	Fitur ini akan menangani proses peminjaman barang yang di lakukan oleh <i>user</i> mahasiswa
2.	Pengembalian	Fitur ini akan menangani proses pengembalian barang yang di lakukan oleh <i>user</i> mahasiswa

Sumber: Data diolah (2023)

Berikut *user stories* dari kebutuhan fitur *user* admin yang telah di kategorikan dapat dilihat pada Tabel 3.3.

Tabel 3. 3 *User Story* Admin

No	Fitur	Judul User Story	User Story
1.	<i>Authentication</i>	<i>Sign In</i>	Sebagai <i>admin</i> yang terdaftar, saya ingin <i>login</i> ke aplikasi <i>web</i> dengan akun yang terdaftar, sehingga saya bisa akses fitur admin.
		<i>Logout</i>	Sebagai <i>admin</i> yang sudah <i>login</i> , saya ingin <i>logout</i> dari aplikasi <i>web</i> .
2.	<i>Add</i>	Tambah Kategori Barang	Sebagai admin saya ingin menambahkan kategori barang baru.
		Tambah Barang	Sebagai admin saya ingin menambahkan barang ke dalam kategori barang yang telah saya buat.
3.	<i>Delete</i>	Hapus Kategori Barang	Sebagai admin saya ingin menghapus kategori barang yang telah saya buat sebelumnya.
		Hapus Barang	Sebagai admin saya ingin menghapus barang yang ada dalam kategori barang.
4.	<i>Edit</i>	Edit Kategori Barang	Sebagai admin saya ingin mengedit kategori barang seperti mengedit nama, <i>icon</i> barang dan tipe barang dari kategori tersebut.
5.	<i>Monitoring</i>	Riwayat peminjaman dan pengembalian	Sebagai admin saya ingin mengetahui riwayat peminjaman dan pengembalian barang dari mahasiswa.
6.	<i>Download</i>	<i>Download database</i> peminjaman dan pengembalian	Sebagai admin saya ingin mengetahui mengunduh riwayat peminjaman dan pengembalian barang dari mahasiswa.

Sumber: Data diolah (2023)

Berikut *user stories* dari kebutuhan fitur *user* mahasiswa yang telah di kategorikan dapat dilihat pada Tabel 3.4.

Tabel 3. 4 *User Story* Mahasiswa

No	Fitur	Judul User Story	User Story
1.	peminjaman	Peminjaman barang	Sebagai <i>user</i> saya ingin meminjam barang.
2.	pengembalian	Pengembalian barang	Sebagai <i>user</i> saya ingin mengembalikan barang yang telah saya pinjam

Sumber: Data diolah (2023)

2. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah perangkat lunak dan perangkat keras yang digunakan. Perangkat lunak yang digunakan dalam membangun bangun *Web* aplikasi pada penelitian ini dapat dilihat pada Tabel 3.5.

Tabel 3. 5 Perangkat Lunak

No	Nama	Versi
1	Visual Studio Code	1.84
2	Flutter	3.13.9
3	Dart	3.1.5
4	Figma	-
5	Firebase	10.17.0

Sumber: Data diolah (2023)

Ada pun perangkat keras yang digunakan dalam membangun bangun *Web* aplikasi pada penelitian ini dapat dilihat pada Tabel 3.6.

Tabel 3. 6 Perangkat Keras

No	Nama	Versi
1	Laptop	Acer, Nitro AN515-58
2	Windows	11
3	Ram 8 GB	3.13.9

Sumber: Data diolah (2023)

3.3. *Planning*

Tahapan *planning* mencakup penentuan prioritas dalam melaksanakan setiap task yang ada. Proses penentuan prioritas ini dikenal dengan istilah *story point*. *Story point* diperoleh dengan memberi ranking pada setiap fitur yang ada. Semakin penting fitur yang akan diterapkan, semakin tinggi pula *story point* yang diberikan, dengan rentang nilai 1-5 (Putra et al., 2024). Berdasarkan analisis fitur pada tabel 3.3 dan tabel 3.4, nilai untuk *story points* dapat dilihat pada Tabel 3.7.

Tabel 3. 7 Ranking Prioritas

Fitur	Story Point	Ranking
Peminjaman	5	1
Pengembalian	5	2
Add	4	3
Delete	4	4
Edit	4	5
Monitoring	3	6
Authentication	2	7
Download	2	8

Sumber: Data diolah (2024)

3.4. *Iteration Inialization*

Langkah ini menandai permulaan setiap siklus iterasi yang akan berlangsung selama pengembangan sistem. Proses iterasi ini dilakukan sesuai dengan tabel 3.7 ranking prioritas.

3.5. *Design*

Desain adalah fase dalam metode PXP yang mencakup proses perancangan sistem, termasuk perancangan *database*, antarmuka pengguna atau *user interface* dan pengalaman pengguna atau *user experience*. Pada perancangan *database* akan dibuat dengan pendekatan *database schema* yang akan menjelaskan alur dari *database* yang ada pada sistem ini, rincian mengenai *database schema* akan dijelaskan pada bab berikutnya. Kemudian untuk pembuatan desain *user interface* akan disajikan dalam bentuk desain *high fidelity* yang didesain menggunakan figma. Dan untuk desain *user experience* akan disajikan dalam bentuk UML yang di dalamnya terdapat *use case diagram* dan *activity diagram* yang akan dijelaskan pada bab selanjutnya.

1. *High Fidelity*

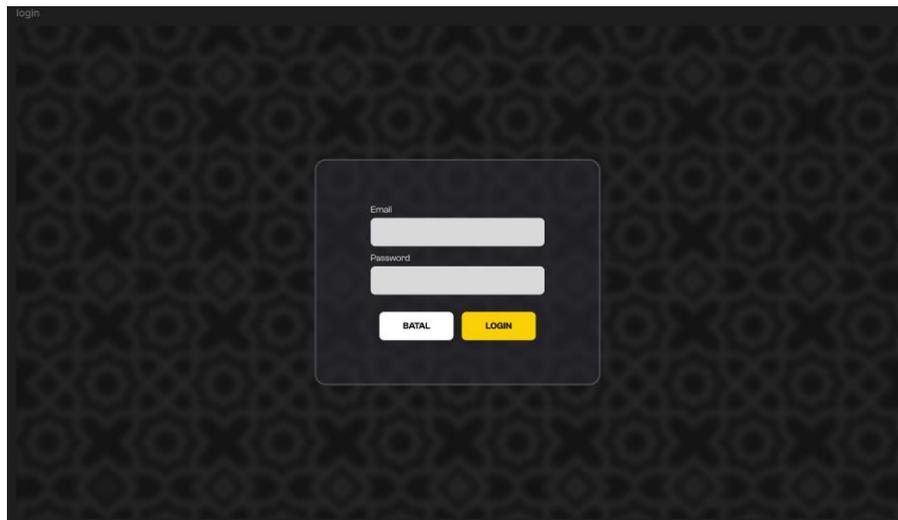
High fidelity adalah desain yang telah memiliki tingkat presisi dan akurasi yang detail, sehingga elemen-elemen seperti warna, ukuran, jarak, dan bentuknya telah dibuat dengan tingkat presisi yang sangat akurasi (Prasetyo, 2022). Berikut desain *high fidelity* dari aplikasi *web* peminjaman inventaris laboratorium.

A. *High fidelity Admin*

Berikut merupakan desain *High fidelity* dari admin

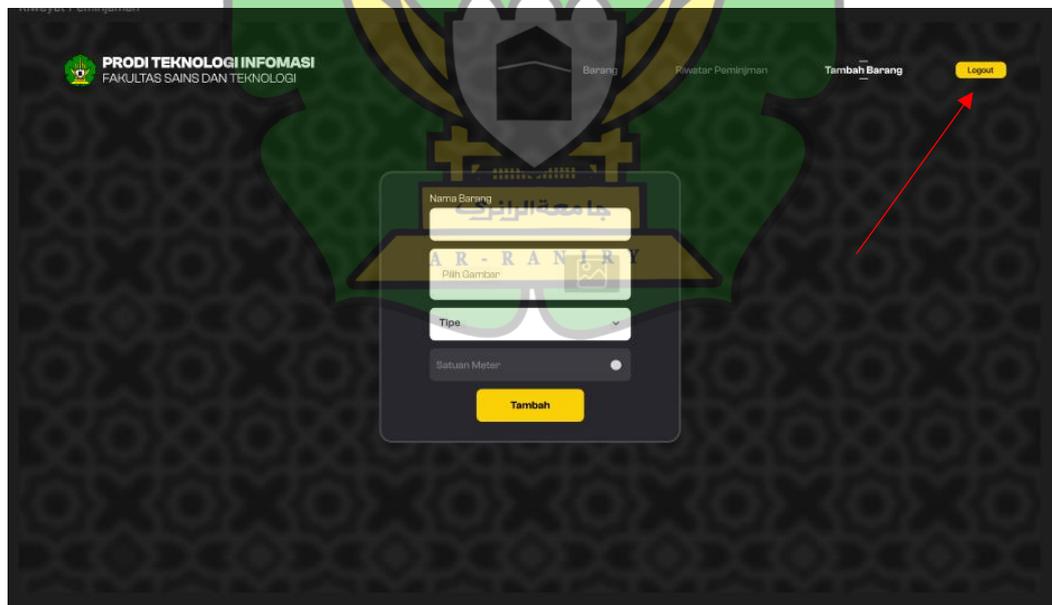
a. *Authentication Screen*

Authentication di bagi menjadi dua yaitu *Sign In* dan *logout*. *Sign In* adalah halaman untuk masuk ke dalam menu admin . Pada halaman ini tersedia *form* yang harus diisi oleh admin seperti *email*, dan *password*. Berikut merupakan desain dari *Sign In Screen*.



Gambar 3. 2 Desain *Sign In Screen*

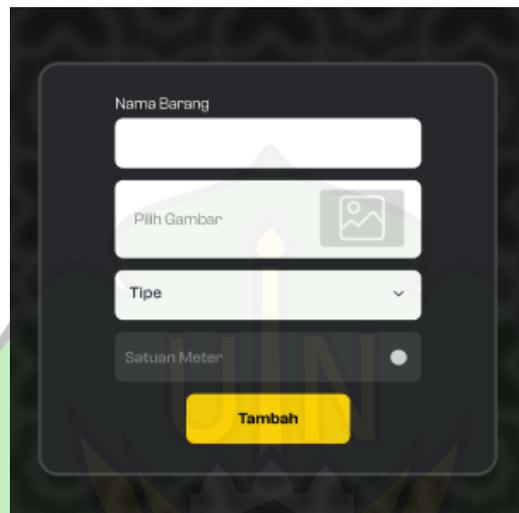
Logout adalah fitur yang hanya dapat diakses oleh admin. Fitur ini berfungsi untuk mengganti status *authentication user* dari yang sebelumnya *login* menjadi mahasiswa (tidak *login*). Berikut merupakan desain dari fitur *logout*.



Gambar 3. 3 Desain *Logout*

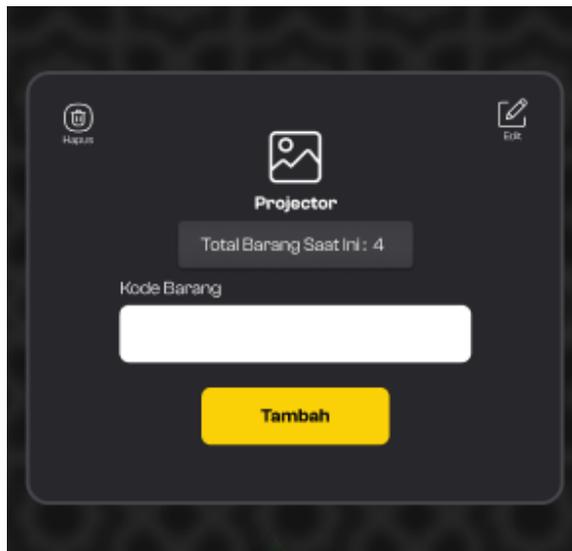
b. *Add Screen*

Add screen di bagi menjadi dua yaitu tambah kategori barang dan tambah barang. Tambah kategori barang adalah halaman untuk menambahkan kategori barang. Pada halaman ini tersedia form yang harus diisi oleh admin seperti nama barang, mengupload *icon* barang dan memilih tipe barang. Berikut merupakan desain dari tambah kategori barang.

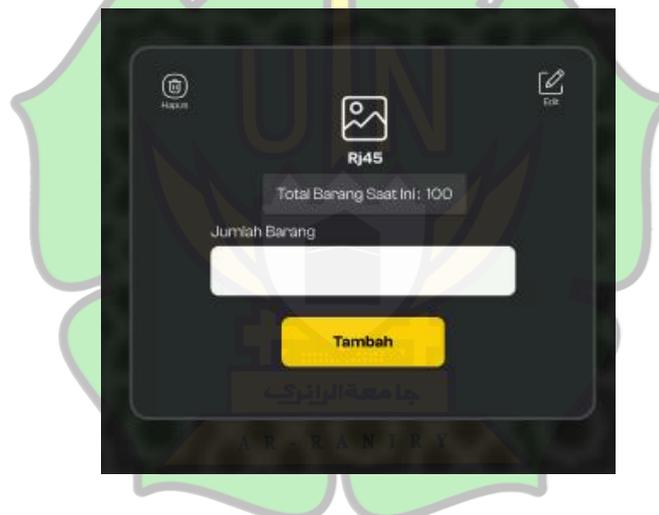


Gambar 3. 4 Desain Tambah Kategori Barang

Tambah barang adalah halaman untuk menambah barang ke dalam kategori barang yang telah dibuat sebelumnya. Pada halaman ini terdapat 2 tampilan berbeda yaitu barang berkode dan barang tidak berkode. Ketika kategori barang yang di pilih admin barang berkode maka admin dapat mengimputkan kode dari barang yang ingin di tambahkan, Ketika kategori barang yang di pilih admin barang tidak berkode maka admin dapat mengisi jumlah barang yang ingin di tambahkan. Berikut merupakan tampilan dari tambah barang berkode dan tampilan tambah barang tidak berkode.



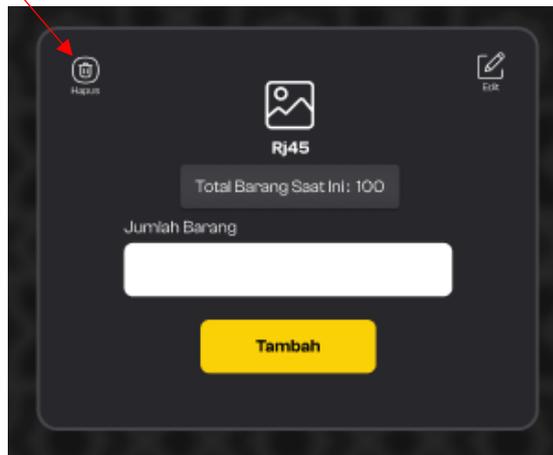
Gambar 3. 5 Desain Tambah Barang Berkode



Gambar 3. 6 Desain Tambah Barang Tidak Berkode

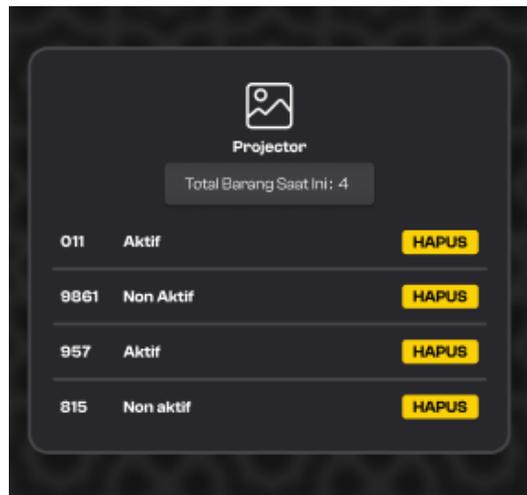
c. *Delete Screen*

Delete screen di bagi menjadi dua yaitu hapus kategori barang dan hapus barang. hapus kategori barang adalah halaman untuk menghapus kategori barang, disini admin bisa mengklik icon hapus untuk menghapus kategori barang. Berikut merupakan tampilan dari hapus kategori barang

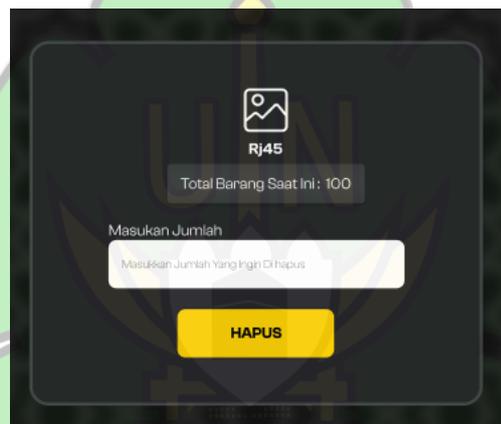


Gambar 3. 7 Desain Hapus Kategori Barang

Hapus barang adalah halaman untuk menghapus barang yang ada dalam kategori barang yang telah dibuat sebelumnya. Pada halaman ini terdapat 2 tampilan berbeda, yaitu barang berkode dan barang tidak berkode. Ketika kategori barang yang dipilih oleh admin adalah barang yang berkode, maka akan di tampilkan *list* barang yang tersedia pada kategori tersebut, kemudian admin dapat memilih barang yang ingin dihapus. Ketika kategori barang yang dipilih oleh admin adalah barang yang tidak berkode, maka admin akan diminta untuk menginputkan jumlah barang yang ingin dihapus dari kategori barang tersebut. Berikut merupakan tampilan dari penghapusan barang berkode dan tidak berkode.



Gambar 3. 8 Desain Hapus Barang Berkode



Gambar 3. 9 Desain Hapus Barang Tidak Berkode

d. Edit Screen

Edit *screen* adalah halaman untuk mengedit kategori yang telah dibuat sebelumnya. Pada halaman ini admin bisa mengedit kategori seperti nama barang, *icon* barang dan tipe barang. Berikut merupakan desain dari edit *screen*.

Frame 166

Nama Barang Baru

Pilih Gambar

Tipe

Satuan Meter

*HANYA UBAH BAGIAN YANG INGIN DIUBAH

Ubah

Gambar 3. 10 Desain Edit Kategori

e. *Monitoring Screen*

Monitoring screen adalah halaman untuk melihat riwayat dari peminjaman dan pengembalian barang oleh mahasiswa. Pada halaman ini admin bisa melihat riwayat seperti Nama, No Telpon, Barang Pinjaman, Jumlah, Kode Barang, Tanggal Pinjaman, Jam Pinjaman, Jumlah Pengembalian, Tanggal Pengembalian dan Jam Pengembalian. Berikut merupakan desain dari *monitoring screen*.

Riwayat Peminjaman

PRODI TEKNOLOGI INFOMASI
FAKULTAS SAINS DAN TEKNOLOGI

Barang Riwayat Peminjaman Tambah Barang Logout

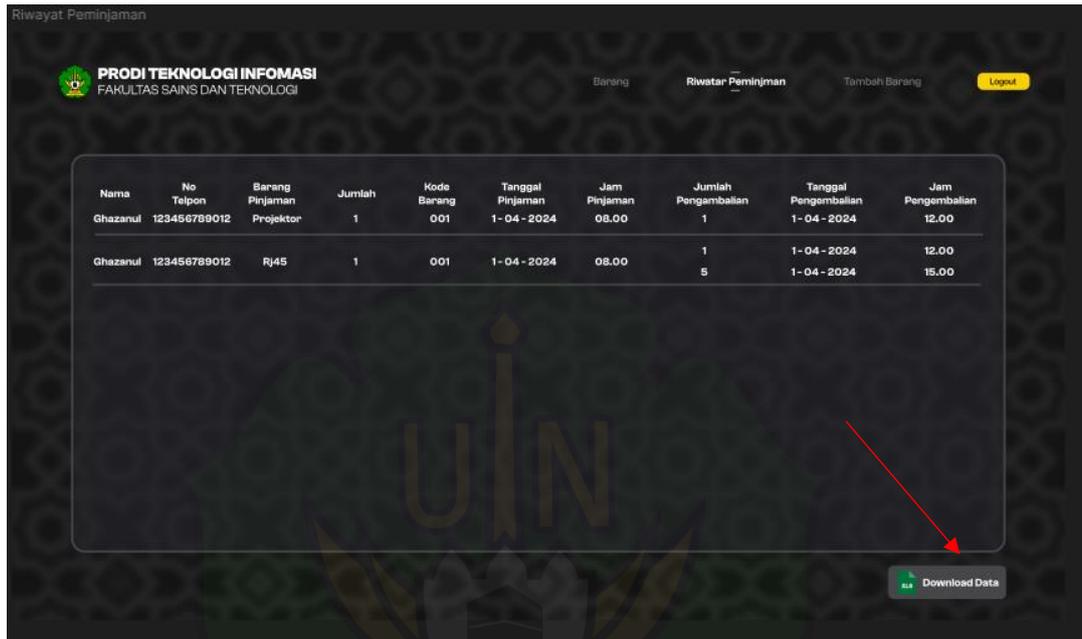
Nama	No Telpon	Barang Pinjaman	Jumlah	Kode Barang	Tanggal Pinjaman	Jam Pinjaman	Jumlah Pengembalian	Tanggal Pengembalian	Jam Pengembalian
Ghazanul	123456789012	Projektor	1	001	1-04-2024	08.00	1	1-04-2024	12.00
Ghazanul	123456789012	RJ45	1	001	1-04-2024	08.00	5	1-04-2024	15.00

Download Data

Gambar 3. 11 Desain *Monitoring Screen*

f. *Download*

Download adalah fitur untuk mengunduh *database* riwayat dari peminjaman dan pengembalian barang oleh mahasiswa. Fiturnya terletak pada bawah kanan yang bernama “Download Data”. Berikut merupakan desain dari fitur *download*.



Gambar 3. 12 Desain *Download Screen*

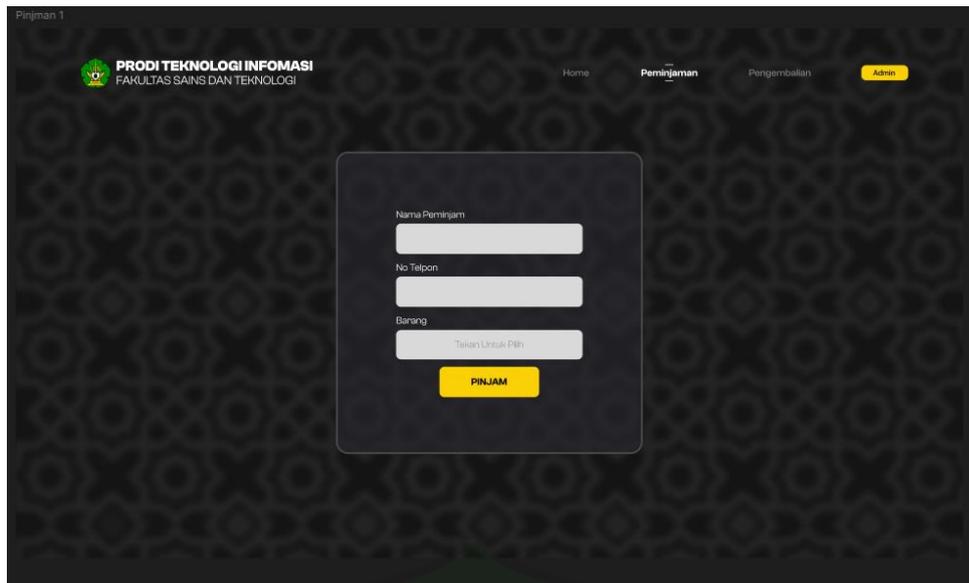
B. *High Fidelity* Mahasiswa

Screen high fidelity mahasiswa di bagi menjadi dua yaitu *screen web* dan *mobile*. Berikut merupakan *Screen* desain *Web* dan *Mobile* mahasiswa.

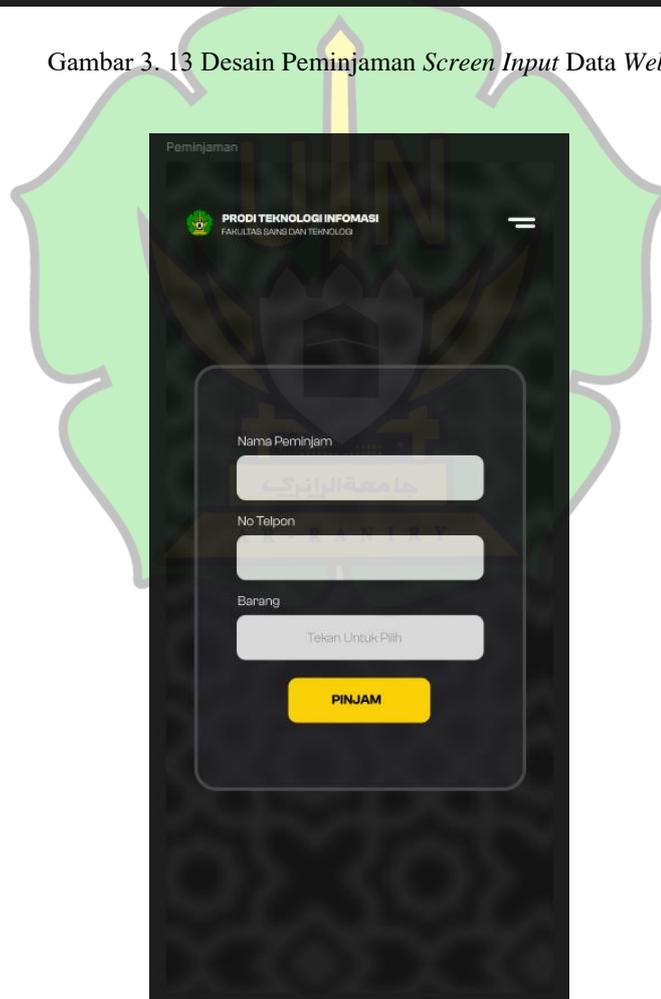
a. *Peminjaman Screen*

Peminjaman Screen adalah halaman untuk mahasiswa meminjam barang. Tampilan *peminjaman Screen* terbagi menjadi 4 yaitu ketika *user* mahasiswa menginputkan data, memilih kategori barang, memilih barang berkode/tidak berkode dan mengkonfirmasi pinjamannya.

Gambar 3.13 dan gambar 3.14 merupakan tampilan ketika *user* menginputkan data. Data yang diinputkan oleh *user* mahasiswa berupa nama dan nomor telpon.



Gambar 3. 13 Desain Peminjaman *Screen Input Data Web*



Gambar 3. 14 Desain Peminjaman *Screen Input Data Mobile*

Gambar 3. 15 dan gambar 3. 16 merupakan tampilan ketika user mahasiswa memilih kategori barang, pada halaman ini terdapat *list* kategori barang yang bisa di pinjam oleh *user* mahasiswa.

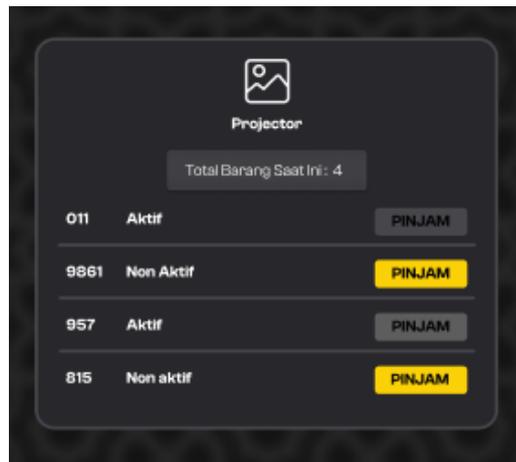


Gambar 3. 15 Desain Peminjaman *Screen* Memilih Kategori Barang *Web*

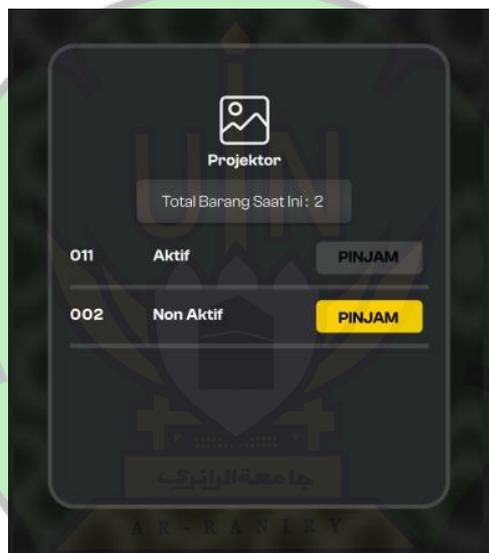


Gambar 3. 16 Desain Peminjaman *Screen* Memilih Barang *Mobile*

Gambar 3. 17 dan Gambar 3. 18 merupakan tampilan ketika user mahasiswa memilih barang berkode, pada halaman ini terdapat *list-list* kode barang yang bisa dipinjam oleh *user* mahasiswa.

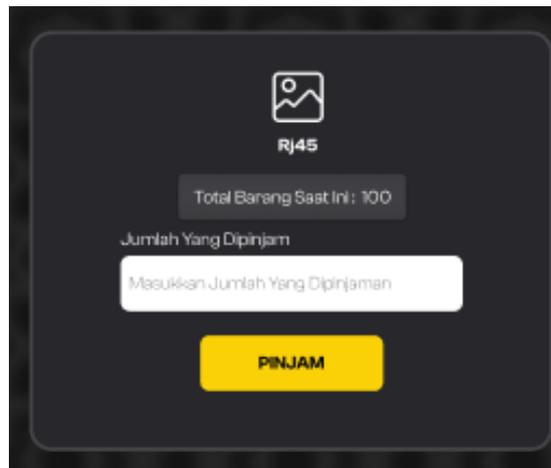


Gambar 3. 17 Desain Peminjaman *Screen* Memilih Barang Berkode *Web*

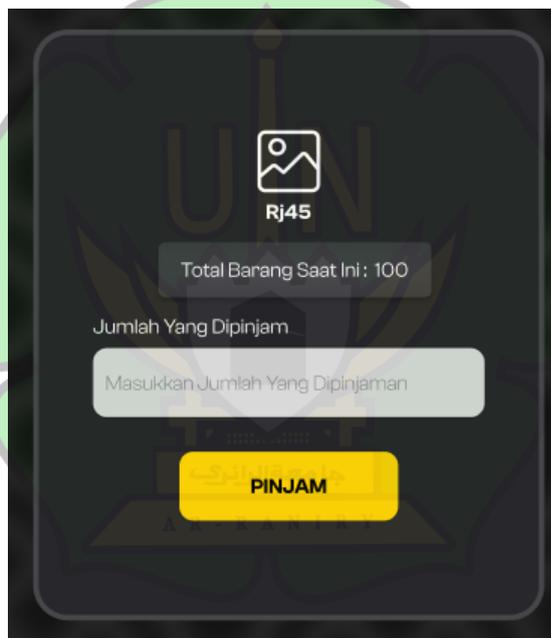


Gambar 3. 18 Desain Peminjaman *Screen* Memilih Barang Berkode *Mobile*

Gambar 3. 19 dan Gambar 3.20 merupakan tampilan ketika user mahasiswa memilih barang tidak berkode, pada halaman ini *user* mahasiswa perlu menginputkan jumlah barang yang ingin dipinjam.



Gambar 3. 19 Desain Peminjaman *Screen* Memilih Barang Tidak Berkode *Web*



Gambar 3. 20 Desain Peminjaman *Screen* Memilih Barang Tidak Berkode *Mobile*

Gambar 3.21 dan Gambar 3.22 merupakan tampilan ketika user mahasiswa mengkonfirmasi pinjamannya, pada halaman ini terdapat beberapa pemberitahuan seperti nama, nomor telpon dan nama barang beserta jumlah barang dipinjam atau kode barang di pinjam.



Gambar 3. 21 Desain Peminjaman *Screen* Konfirmasi Pinjaman *Web*

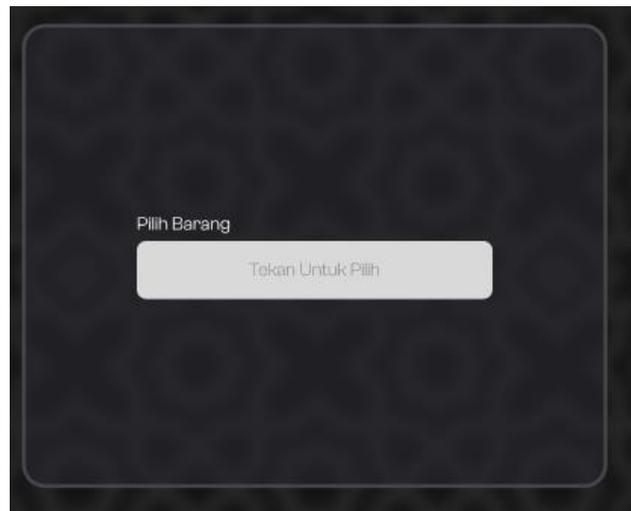


Gambar 3. 22 Desain Peminjaman *Screen* Konfirmasi Pinjaman *Mobile*

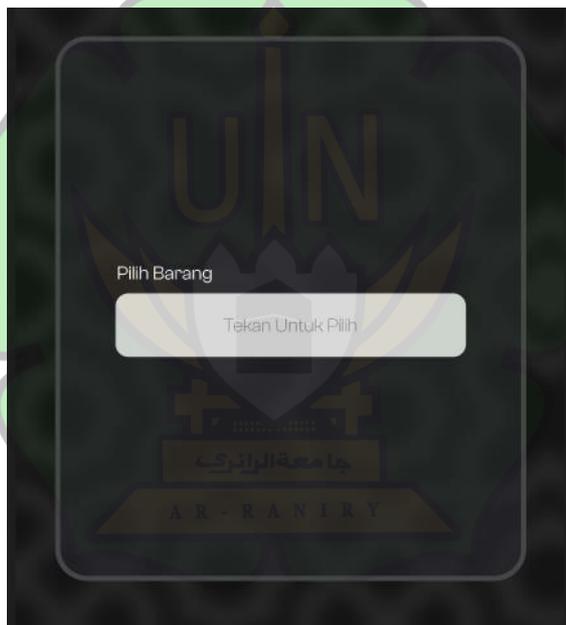
b. Pengambalian *Screen*

Pengambalian *screen* adalah halaman untuk mahasiswa mengembalikan barang yang dipinjam. Tampilan pengambalian *Screen* terbagi menjadi 6, yaitu ketika user mahasiswa ingin memilih barang, *list* kategori barang, memilih barang berkode/tidak berkode, telah memilih barang berkode, telah memilih barang tidak berkode dan konfirmasi barang yang ingin dikembalikan.

Gambar 3. 23 dan Gambar 3.24 merupakan tampilan ketika user mahasiswa ingin memilih barang yang ingin dikembalikan. Pada halaman ini user akan memilih tombol yang disediakan untuk pilihan barang yang ingin dikembalikan.

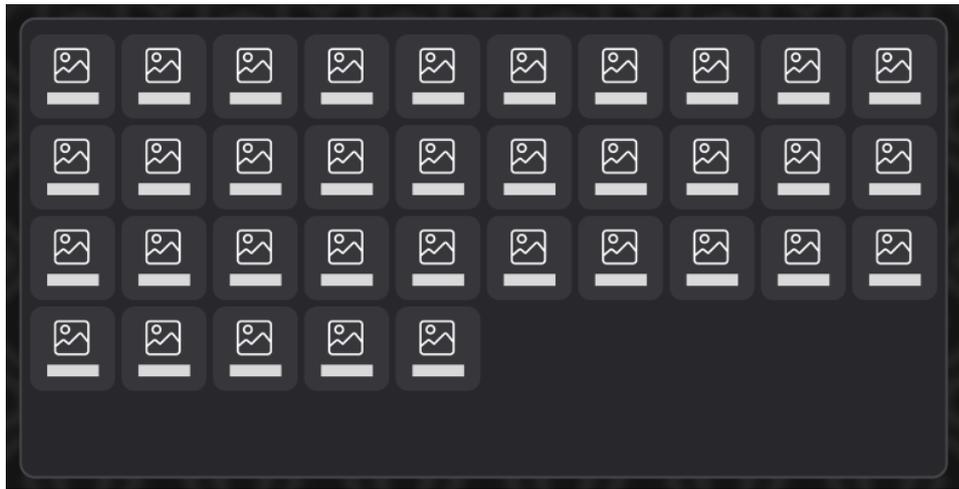


Gambar 3. 23 Desain Pengembalian *Screen* Memilih Barang *Web*

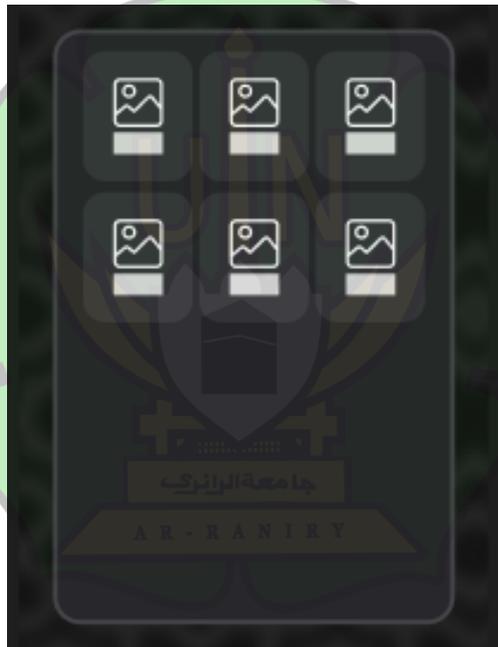


Gambar 3. 24 Desain Pengembalian *Screen* Memilih Barang *Mobile*

Gambar 3. 25 dan gambar 3. 26 merupakan tampilan ketika user mahasiswa memilih *list* kategori barang, pada halaman ini terdapat *list* kategori barang yang bisa di pinjam oleh *user* mahasiswa

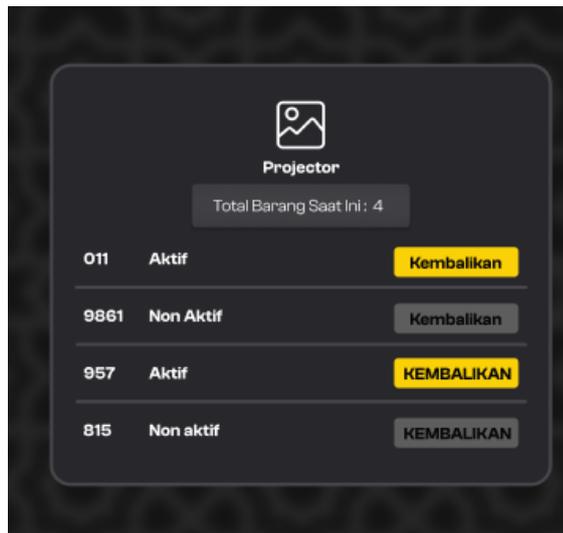


Gambar 3. 25 Desain Pengembalian *Screen List* Kategori Barang *Web*



Gambar 3. 26 Desain Pengembalian *Screen List* Kategori Barang *Mobile*

Gambar 3. 27 dan Gambar 3.28 merupakan tampilan ketika user mahasiswa memilih barang berkode, pada halaman ini terdapat *list-list* kode barang yang bisa dikembalikan oleh *user* mahasiswa.



Gambar 3. 27 Desain Pengembalian *Screen* Barang Berkode *Web*



Gambar 3. 28 Desain Pengembalian *Screen* Barang Berkode *Mobile*

Gambar 3. 29 dan Gambar 3.30 merupakan tampilan ketika *user* mahasiswa memilih barang tidak berkode, pada halaman ini *user* mahasiswa perlu menginputkan jumlah barang yang ingin dikembalikan.

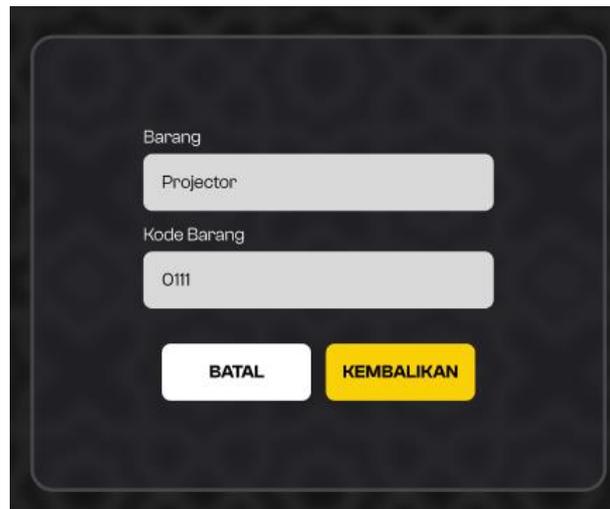


Gambar 3. 29 Desain Pengembalian *Screen* Barang Tidak Berkode *Web*



Gambar 3. 30 Desain Pengembalian *Screen* Barang Tidak Berkode *Mobile*

Gambar 3.31 dan Gambar 3. 32 merupakan tampilan ketika user mahasiswa telah memilih barang berkode untuk dikembalikan. Pada halaman ini akan ditampilkan nama barang dan kode barang.



Gambar 3. 31 Desain Pengembalian *Screen* Memilih Barang Berkode *Web*



Gambar 3. 32 Desain Pengembalian *Screen* Memilih Barang Berkode *Mobile*

Gambar 3.33 dan Gambar 3.34 merupakan tampilan ketika user mahasiswa telah memilih barang tidak berkode untuk dikembalikan. Pada halaman ini akan ditampilkan nama barang dan kode barang.

A screenshot of a web-based return form. The form is set against a dark background and contains the following fields and buttons:

- Nama Peminjam:** A text input field with the placeholder text "Masukkan Nama Seperti saat Meminjam".
- No Telpn:** A text input field with the placeholder text "Masukkan No Telpn Seperti saat Meminjam".
- Barang:** A text input field containing the value "Rj45".
- Jumlah Yang Di kembalikan:** A text input field containing the value "Rj45".
- Buttons:** Two buttons at the bottom: a white button labeled "BATAL" and a yellow button labeled "KEMBALIKAN".

Gambar 3. 33 Desain Pengembalian *Screen* Memilih Barang Tidak Berkode *Web*

A screenshot of a mobile-based return form. The form is set against a dark background and contains the following fields and buttons:

- Nama Peminjam:** A text input field with the placeholder text "Masukkan Nama Seperti saat Meminjam".
- No Telpn:** A text input field with the placeholder text "Masukkan No Telpn Seperti saat Meminjam".
- Barang:** A text input field containing the value "RJ45".
- Jumlah kembalikan:** A text input field containing the value "50".
- Buttons:** Two buttons at the bottom: a white button labeled "BATAL" and a yellow button labeled "KEMBALIKAN".

Gambar 3. 34 Desain Pengembalian *Screen* Memilih Barang Tidak Berkode *Mobile*

Gambar 3.35 dan Gambar 3.36 merupakan tampilan ketika user mahasiswa mengkonfirmasi barang yang ingin dikembalikan. pada halaman ini terdapat beberapa pemberitahuan seperti nama, no telpon, nama barang dan jumlah barang yang dikembalikan atau kode barang yang dikembalikan.



Gambar 3. 35 Desain Pengembalian *Screen* Konfirmasi Pengembalian *Web*



Gambar 3. 36 Desain Pengembalian *Screen* Konfirmasi Pengembalian *Mobile*

3.2. Implementasi

Pada tahap ini merupakan tahap dimana penulis membuat kode pemrograman. Penulis melakukan implementasi antarmuka pengguna yang telah

direncanakan pada fase desain. Pembangunan sistem pada penelitian ini dilakukan hanya berfokus pada penulisan kode. Pada penelitian ini, fase implementasi dibagi menjadi dua tahapan, yaitu *frontend* dan *backend*. Pada tahap *frontend*, dilakukan penelisan kode yang berfokus pada tampilan aplikasi *Web* peminjaman inventaris laboratorium. Sedangkan tahap *backend*, dilakukan penelisan kode yang berfokus pada pengelolaan *database* aplikasi *web* peminjaman inventaris laboratorium.

Hasil dari proses implementasi ini nantinya akan menciptakan sebuah aplikasi *web* yang dapat diakses oleh mahasiswa dan staf Prodi Teknologi Informasi. Aplikasi ini akan berperan penting dalam mempermudah peminjaman inventaris laboratorium.

3.3. System Testing

Pengujian sistem, yang juga dikenal sebagai pengujian integrasi, merupakan tahap penting dalam validasi keseluruhan fungsionalitas fitur yang telah diimplementasikan dan terintegrasi dalam sistem. Pengujian sistem ini termasuk dalam kategori teknik pengujian *black box*, yang berarti pengujian dilakukan tanpa perlu memperhatikan detail kode program. Fokusnya adalah menguji seluruh aliran aplikasi dari awal hingga akhir, dari perspektif admin, yang memungkinkan pengujian *end-to-end*.

Selama pengujian sistem, aplikasi *web* dijalankan dalam lingkungan yang sesungguhnya, baik itu di perangkat nyata maupun dalam emulator. Tujuan utama pengujian sistem adalah memastikan bahwa sistem atau aplikasi dapat merespons input yang diberikan dengan benar, menghasilkan output yang sesuai dengan harapan, dan berfungsi dengan baik dalam kondisi lingkungan yang sesungguhnya. Ini membantu menentukan apakah sistem atau aplikasi tersebut siap untuk digunakan secara luas oleh mahasiswa atau admin dan memastikan bahwa semua fitur telah teruji secara menyeluruh. Pengujian ini akan dilakukan oleh operator prodi Teknologi Informasi. Tabel 3.8 adalah test case pengujian *black box* yang digunakan.

Tabel 3. 8 Test Case Pengujian Black Box

No	Feature	User story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
1.	Authenticat ion	Sign In	Melakukan Sign In pada aplikasi web	<ol style="list-style-type: none"> 1. Mengklik button admin pada menu 2. Masuk ke halaman Sign In 3. Masukkan email 4. Masukkan password mengklik button login 	email = admin@gmail.com password = admin123	User berhasil masuk ke halaman admin	
		Logout	Melakukan logout pada aplikasi web	<ol style="list-style-type: none"> 1. User admin menekan menu logout. 		User keluar dari halaman admin dan mengganti status Authentication user dari yang sebelumnya login menjadi mahasiswa (tidak login).	

No	Feature	User story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/ Fail
2.	Add	Tambah Kategori Barang	Melakukan penambahan kategori barang pada aplikasi web	<ol style="list-style-type: none"> 1. Mengklik menu tambah barang. 2. User admin menekan button tambah kategori 3. User admin mengisi nama barang, icon barang dan memilih tipe barang(berkode/ tidak berkode, sekali pakai) 4. Menakan button tambah 		<ol style="list-style-type: none"> 1. User admin menekan button tambah kategori 2. Muncul pop up untuk memastikan data yang di perlukan 3. Mengirim data yang telah di inputkan sebelumnya 4. ke database setelah menekan button tambah 5. Menampilkan data kategori barang baru ke list kategori barang pada halaman menu tambah barang. 	



No	Feature	User story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/ Fail
		Tambah Barang	Melakukan penambahan barang pada aplikasi web	<ol style="list-style-type: none"> 1. Mengklik menu tambah barang. 2. User admin memilih kategori barang yang ingin ditambahkan barangnya. 3. User admin menambahkan kode barang jika kategori barang yang dipilih adalah barang berkode, user admin menambah jumlah total barang jika kategori barang yang dipilih barang tidak berkode/sekali pakai. 4. Mengklik button tambah. 		<ol style="list-style-type: none"> 1. User admin memilih kategori barang yang ingin ditambahkan barangnya. 2. Muncul pop up untuk menambahkan barang. 3. User menginputkan kode atau jumlah barang, lalu menekan button tambah. 4. Jumlah barang bertambah. 	
3.	Delete	Hapus Kategori Barang	Melakukan penghapusan kategori barang pada aplikasi web.	<ol style="list-style-type: none"> 1. Mengklik menu tambah barang. 2. User admin memilih kategori barang yang ingin dihapus. 		<ol style="list-style-type: none"> 1. User admin memilih kategori barang yang ingin dihapus. 2. Muncul pop up, lalu user admin 	

No	Feature	User story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
				<ol style="list-style-type: none"> 3. User admin menekan icon hapus. 4. Menekan button hapus. 		<ol style="list-style-type: none"> 3. menekan icon hapus. 4. Muncul pop up konfirmasi penghapusan kategori barang. Lalu user admin menekan button hapus. 5. Menampilkan halaman menu tambah barang dengan data yang telah diperbaharui. 	
		Hapus Barang	Melakukan penghapusan barang pada aplikasi web.	<ol style="list-style-type: none"> 1. Mengklik menu barang. 2. User admin memilih kategori barang yang ingin hapus barangnya. 3. Jika user admin memilih kategori barang berkode, maka user admin akan memilih barang yang ingin dihapus. Namun jika user admin memilih kategori barang tidak berkode/sekalipakai, maka user admin menginputkan jumlah dari barang yang ingin dihapus. 4. Mengklik button hapus. 		<ol style="list-style-type: none"> 1. User admin memilih kategori barang yang ingin hapus barangnya. 2. Muncul pop up Jika kategori barang yang dipilih barang berkode, maka user admin memilih barang yang ingin dihapus. Namun jika kategori barang yang dipilih barang tidak berkode/sekali pakai, maka user admin menginputkan jumlah barang yang ingin dihapus. 	

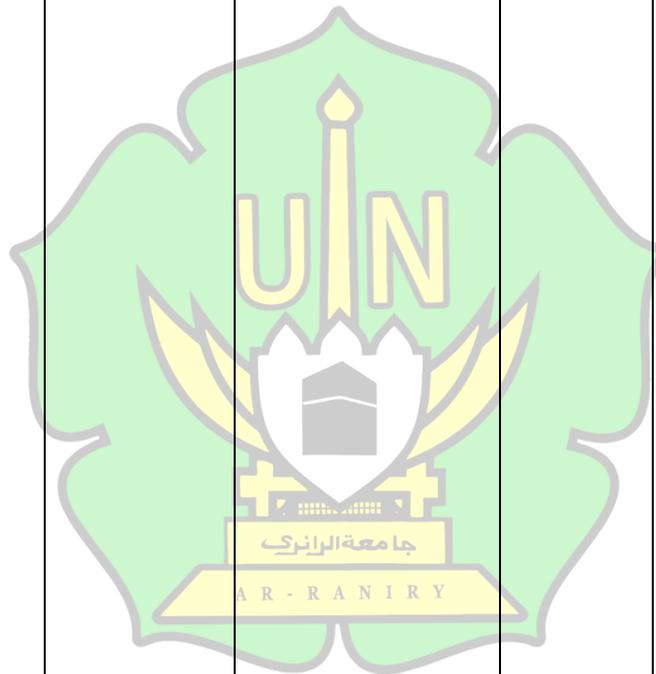
No	Feature	User story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
						4. Mengklik button hapus. 5. Kembali ke halaman menu barang, yang datanya telah diperbaharui.	
4.	Edit	Edit kategori Barang	Melakukan pengeditan data pada kategori barang pada aplikasi web.	1. Mengklik menu tambah barang. 2. User admin memilih kategori barang yang ingin diedit barangnya. 3. User admin menekan icon edit. 4. Memperbaharui data yang ingin diganti. 5. User admin menekan button ubah.		1. User admin memilih kategori barang yang ingin diedit barangnya. 2. Muncul pop up, lalu user admin menekan icon edit. 3. Muncul pop up, lalu user admin memperbaharui data yang ingin diganti. 4. Menekan button ubah. 5. Menampilkan halaman menu tambah barang yang telah diperbaharui datanya.	
5.	Monitoring	Melihat riwayat peminjaman dan pengembalian barang.	Menampilkan data peminjaman dan pengembalian barang.	Mengklik menu riwayat peminjaman.		1. Menampilkan halaman riwayat peminjaman dan pengembalian, setelah mengklik menu riwayat peminjaman 2. Menampilkan list data peminjan dan	

No	Feature	User story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
						pengembalian user mahasiswa.	
6.	Download	Mendownload database report	Melakukan Download data peminjaman dan pengembalian	<ol style="list-style-type: none"> 1. Mengklik menu riwayat peminjaman. 2. Mengklik button download data. 3. Melihat data pada google sheet. 4. Mendownload google sheet. 		<ol style="list-style-type: none"> 1. Setelah mengklik button download data maka akan ditambah tab baru yang mengarah ke google sheets. 2. Google sheets mengambil data dari database melalui apps script pada extantion 	
7.	Peminjaman	Peminjaman	Melakukan peminjaman barang pada aplikasi web.	<ol style="list-style-type: none"> 1. Mengklik menu peminjaman. 2. Mengisi data yang diperlukan berupa nama dan no telpon. 3. Menekan button "tekan untuk pilih" untuk memilih barang. 4. Memilih kategori barang yang ingin dipinjam. 5. Jika kategori barang berkode yang dipilih, maka user mahasiswa memilih barang untuk dipinjam. Namun jika 		<ol style="list-style-type: none"> 1. Mengisi data yang diperlukan berupa nama dan no telpon. 2. Menekan button "tekan untuk pilih" untuk memilih barang. 3. Muncul pop up memilih kategori barang. 4. Menekan kategori barang yang ingin dipinjam. 5. Jika kategori barang yang dipilih barang berkode, maka 	

No	Feature	User story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/ Fail
				kategori barang yang di pilih barang tidak berkode/sekali pakai, maka user mahasiswa menginputkan jumlah barang yang ingin dipinjam. 6. Menekan button pinjam. 7. kembali ke halaman input data peminjaman. 8. mengklik button pinjam. 9. Menekan pinjam pada pop up konfirmasi peminjaman.		user mahasiswa memilih barang yang ingin dipinjam. Namun jika kategori barang yang dipilih barang tidak berkode/sekali pakai, maka user mahasiswa menginputkan jumlah barang yang ingin dipinjam. 6. Mengklik button pinjam. 7. Kembali ke halaman input data peminjaman, lalu mengklik button pinjam. 8. Muncul pop up konfirmasi peminjaman yang berisi nama, no telpon dan nama barang beserta kode atau jumlah barang sesuai inputan user mahasiswa sebelumnya. 9. Menekan button pinjam. 10. Menampilkan halaman input data peminjaman.	

No	Feature	User story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
8.	pengembalian	pengembalian	Melakukan pengembalian barang pada aplikasi web.	<ol style="list-style-type: none"> 1. Menekan menu pengembalian. 2. Menekan button "tekan untuk pilih" untuk memilih barang. 3. Memilih kategori barang yang ingin dikembalikan 4. Jika kategori barang berkode yang dipilih, maka user mahasiswa memilih barang untuk dikembalikan. Namun jika kategori barang yang di pilih barang tidak berkode, maka user mahasiswa menginputkan jumlah barang yang ingin dikembalikan. 5. Mengklik button kembalikan pada pop up pemilihan barang. 6. Jika yang dipilih barang tidak berkode, maka user mahasiswa perlu mengisi nama dan no telpon lalu mengklik button kembalikan. Namun jika barang yang dipilih barang 		<ol style="list-style-type: none"> 1. Menekan button "tekan untuk pilih" untuk memilih barang. 2. Muncul pop up memilih kategori barang. 3. Menekan kategori barang yang ingin dikembalikan 4. Jika kategori barang yang dipilih barang berkode, maka user mahasiswa memilih barang yang ingin dikembalikan. Namun jika kategori barang yang dipilih barang tidak berkode, maka user mahasiswa menginputkan jumlah barang yang ingin dikembalikan. 5. Mengklik button kembalikan. 6. Menampilkan halaman pengembalian dengan data yang sesuai pilihan user mahasiswa sebelumnya. 	

No	Feature	User story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/ Fail
				<p>berkode, maka user mahasiswa akan langsung mengklik button kembalikan</p> <p>7. Lalu Menakan kembalikan pada pop up konfirmasi pengembalian.</p>		<p>7. Jika barang yang dikembalikan barang berkode, maka yang ditampilkan adalah nama barang dan kode barang. Namun jika barang yang dikembalikan barang tidak berkode, maka yang ditampilkan adalah inputan nama, inputan no telpon, nama barang dan jumlah yang dikembalikan.</p> <p>8. Mengklik button kembalikan.</p> <p>9. Muncul pop up konfirmasi pengembalian yang berisi nama, no telpon, nama barang dan kode atau jumlah barang sesuai inputan user mahasiswa sebelumnya.</p> <p>10. Menekan button pinjam.</p> <p>11. Menampilkan halaman pengembalian.</p>	



3.4. Tempat penelitian

Penelitian ini akan di lakukan di Prodi Teknologi Informasi Universitas Islam Negeri Ar-Raniry.



BAB IV

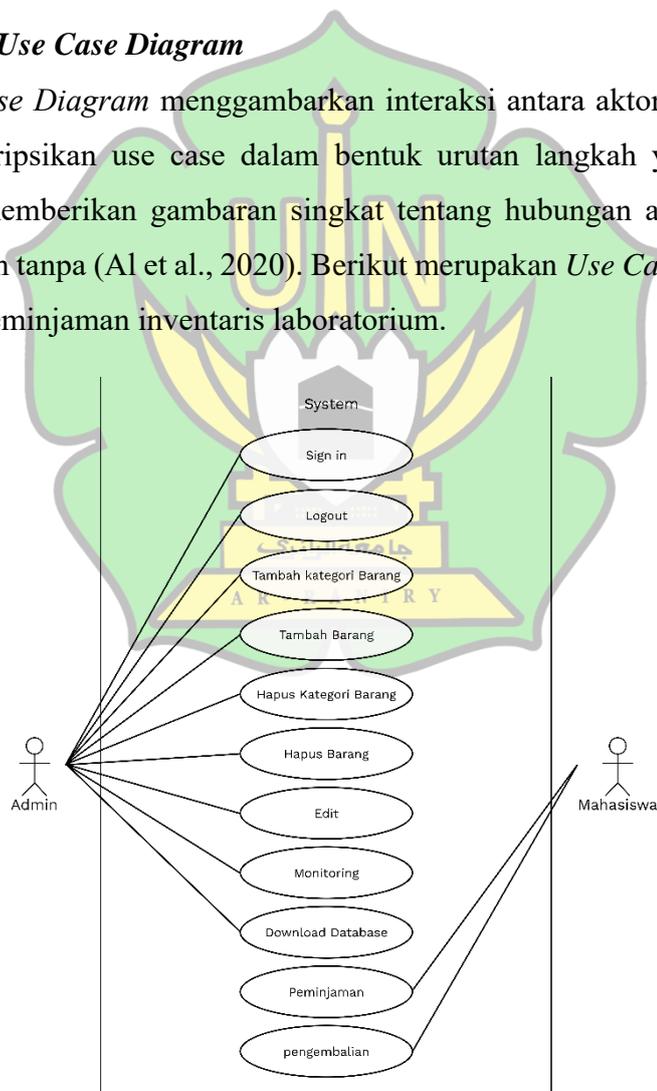
HASIL DAN PEMBAHASAN

4.1 Design

Pada bagian ini, akan dibahas mengenai desain dan pemodelan pengembangan aplikasi *web* peminjaman inventaris laboratorium. Fokus utama desain ini adalah untuk memastikan bahwa aplikasi yang dibuat sesuai dengan kebutuhan yang telah dijelaskan dalam bentuk *user story* pada bab sebelumnya, sebagaimana tercantum dalam Tabel 3.3. dan Tabel 3.4

4.3.1 Use Case Diagram

Use Case Diagram menggambarkan interaksi antara aktor dengan sistem, serta mendeskripsikan use case dalam bentuk urutan langkah yang sederhana. Diagram ini memberikan gambaran singkat tentang hubungan antara aktor, use case, dan sistem tanpa (Al et al., 2020). Berikut merupakan *Use Case Diagram* dari aplikasi *web* peminjaman inventaris laboratorium.



Gambar 4. 1 *Use Case Diagram*

Use Case Diagram pada gambar 4.1 memiliki 2 aktor yaitu admin dan mahasiswa. Aktor admin memiliki hubungan dengan 9 *use case* yaitu *sign in*, *logout* tambah kategori barang, tambah barang, hapus kategori barang, hapus barang, edit, *monitoring* dan *download database*. Sedangkan aktor mahasiswa memiliki hubungan dengan 2 *use case* yaitu *peminjaman* dan *pengembalian*.

4.3.2 Database Schema

Database Schema adalah struktur yang menentukan bagaimana data disusun dalam sebuah database relasional. Ini mencakup elemen-elemen seperti nama tabel, kolom, tipe data, dan hubungan antara entitas-entitas yang terkait (ibm, 2024). Berikut merupakan *database schema* dari aplikasi *web* peminjaman inventaris laboratorium.



Gambar 4.2 Database Schema

Dari hasil perancangan *database schema* didapat 5 tabel, yaitu tabel *barang*, *User*, *BarangPinjaman*, *KumpulanFileIconGambar*, dan *Report* dengan rincian sebagai berikut:

Tabel 4. 1 Tabel Barang

Field Name	Data Type	Description
id	int	Primary Key
Berkode	boolean	
Gambar	varchar	

Field Name	Data Type	Description
JumlahTerpakai	int	
JumlahTotalBarang	int	
Nama	varchar	
SatuanMeter	boolean	
SekaliPakai	boolean	
kode	varchar	
Status	boolean	

Sumber: Data diolah (2024)

Tabel 4. 2 Tabel *User*

Field Name	Data Type	Description
id	int	<i>Primary Key</i>
Nama	varchar	
Notelpon	varchar	
BarangId	int	
JumlahTotalBarang	int	

Sumber: Data diolah (2024)

Tabel 4. 3 Tabel BarangPinjaman

Field Name	Data Type	Description
id	int	<i>Primary Key</i>
Berkode	boolean	
JumlahatauKodeBarang	varchar	
NamaBarang	varchar	
UserId	int	

Sumber: Data diolah (2024)

Tabel 4. 4 Tabel KumpulanFileIconGambar

Field Name	Data Type	Description
id	int	Primary Key
NamaBarang	varchar	
namaFileGambar	varchar	
BarangId	int	

Sumber: Data diolah (2024)

Tabel 4. 5 Tabel Report

Field Name	Data Type	Description
Id	int	Primary Key
BarangPinjamanId	varchar	
JumlahPeminjaman	int	
KodeBarang	varchar	
NamaBarang	varchar	
NamaPeminjam	varchar	
NoTelpon	varchar	
PercobaanPengembalian	varchar	
barangBerkode	boolean	
berM	boolean	
jamPeminjaman	timestamp	
jumlahPengembalian	int	
percobaanPeminjaman	int	
tanggalPeminjaman	datetime	
tanggalPengembalian	datetime	

Sumber: Data diolah (2024)

4.2 Activity Diagram

Activity Diagram digunakan untuk menggambarkan alur proses dan urutan aktivitas yang terjadi ketika pengguna berinteraksi dengan aplikasi. Di bawah ini ditampilkan activity diagram untuk fitur-fitur yang dikembangkan atau dirancang dalam aplikasi *web* peminjaman inventaris laboratorium.

1. Fitur-Fitur Admin

Ada beberapa fitur pada aplikasi *web* peminjaman inventaris laboratorium yang hanya dapat diakses oleh user dengan tipe admin diantaranya adalah *authentication*, *add*, *delete*, *edit*, *monitoring* dan *download*. Berikut merupakan *Activity Diagram* dari setiap fitur tersebut.

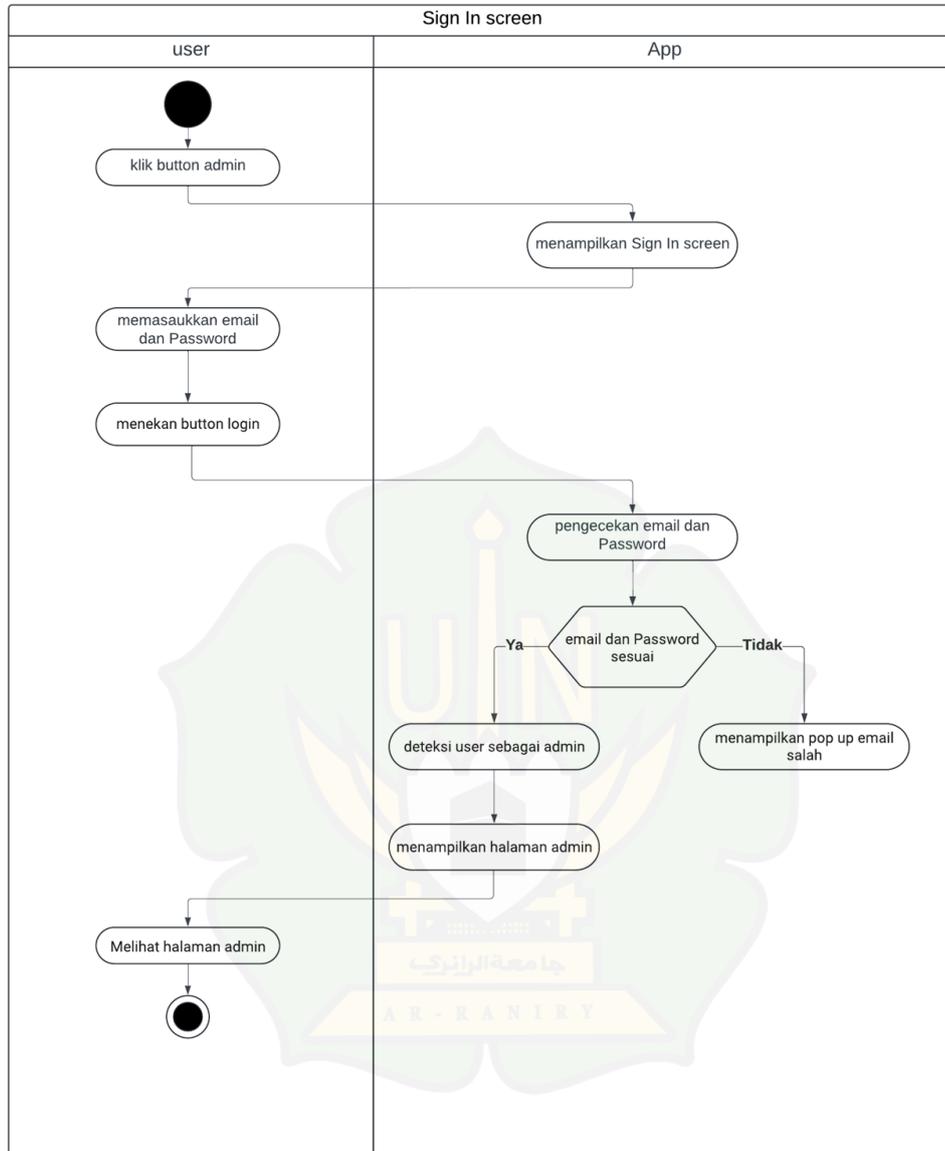
A. Authentication

Fitur *authentication* dibagi menjadi 2, yaitu *Sign In* dan *Logout*. Berikut merupakan *activity diagramnya*

a. Sign In

Gambar 4.3 menggambarkan rangkaian proses *Sign in* yang dilakukan *user* di aplikasi *web* peminjaman inventaris laboratorium. Adapun rinciannya ialah sebagai berikut:

- *User* mengklik *button* admin pada menu
- Kemudian aplikasi menampilkan halaman *Sign In*
- *User* memasukan *email* dan *password*, kemudian menekan *button login*
- kemudian aplikasi melakukan pengecekan apakah *email* dan *password* telah benar atau salah sesuai dengan *email* dan *password* admin *firebase authentication*, jika salah maka akan muncul *pop up* notifikasi bahwa *email* yang diinput salah, dan jika benar maka aplikasi akan menampilkan halaman admin.



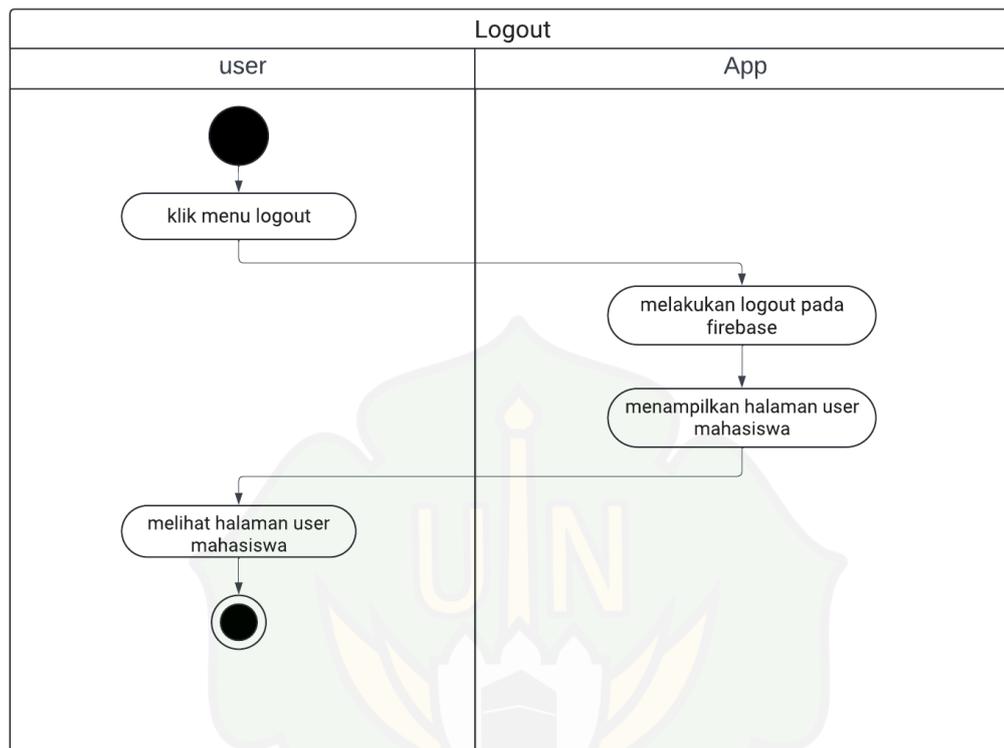
Gambar 4. 3 Activity Diagram Sign Up

Sumber: Data diolah (2024)

b. Logout

Gambar 4.4 menggambarkan rangkaian proses *logout* yang dilakukan *user* di aplikasi *web* peminjaman inventaris laboratorium. Adapun rinciannya ialah sebagai berikut:

- *User* mengklik *button logout* pada menu
- Kemudian aplikasi akan melakukan *logout* pada *database* dan menampilkan halaman user mahasiswa



Gambar 4. 4 *Activity Diagram logout*

Sumber: Data diolah (2024)

B. Add

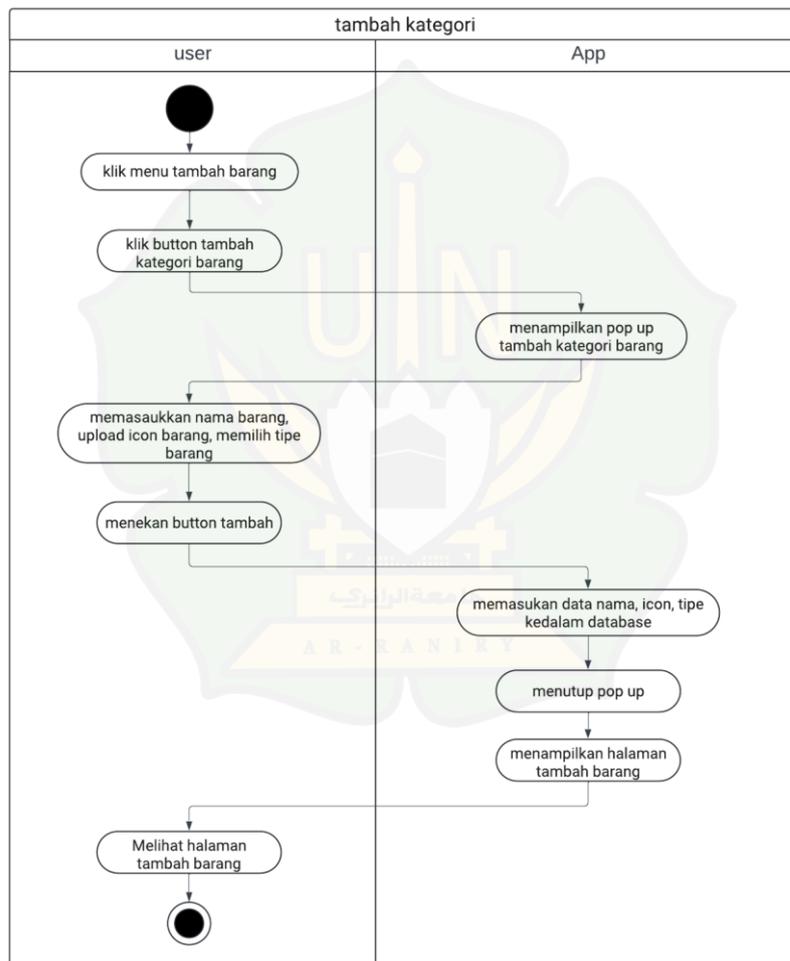
Fitur *add* dibagi menjadi 2, yaitu tambah kategori barang dan tambah barang. Berikut merupakan *activity diagramnya*

a. Tambah Kategori Barang

Gambar 4.5 menggambarkan rangkaian aktifitas untuk melakukan tambah kategori barang yang dilakukan *user* di aplikasi *web* peminjaman inventaris laboratorium. Adapun rinciannya ialah sebagai berikut:

- *User* mengklik menu tambah barang, lalu mengklik *button* tambah kategori.

- Kemudian aplikasi akan menampilkan *pop up* tambah kategori barang.
- Lalu *user* melakukan proses penambahan kategori barang dengan mengisi nama barang, mengunggah *icon* barang dan memilih tipe dari barang. Selanjutnya *user* mengklik *button* Tambah.
- Kemudian aplikasi akan memasukkan data yang telah *diinputkan* oleh *user* sebelumnya ke dalam *firebase* lalu melakukan *refresh* data, menutup *pop up* dan menampilkan halaman tambah barang kembali.



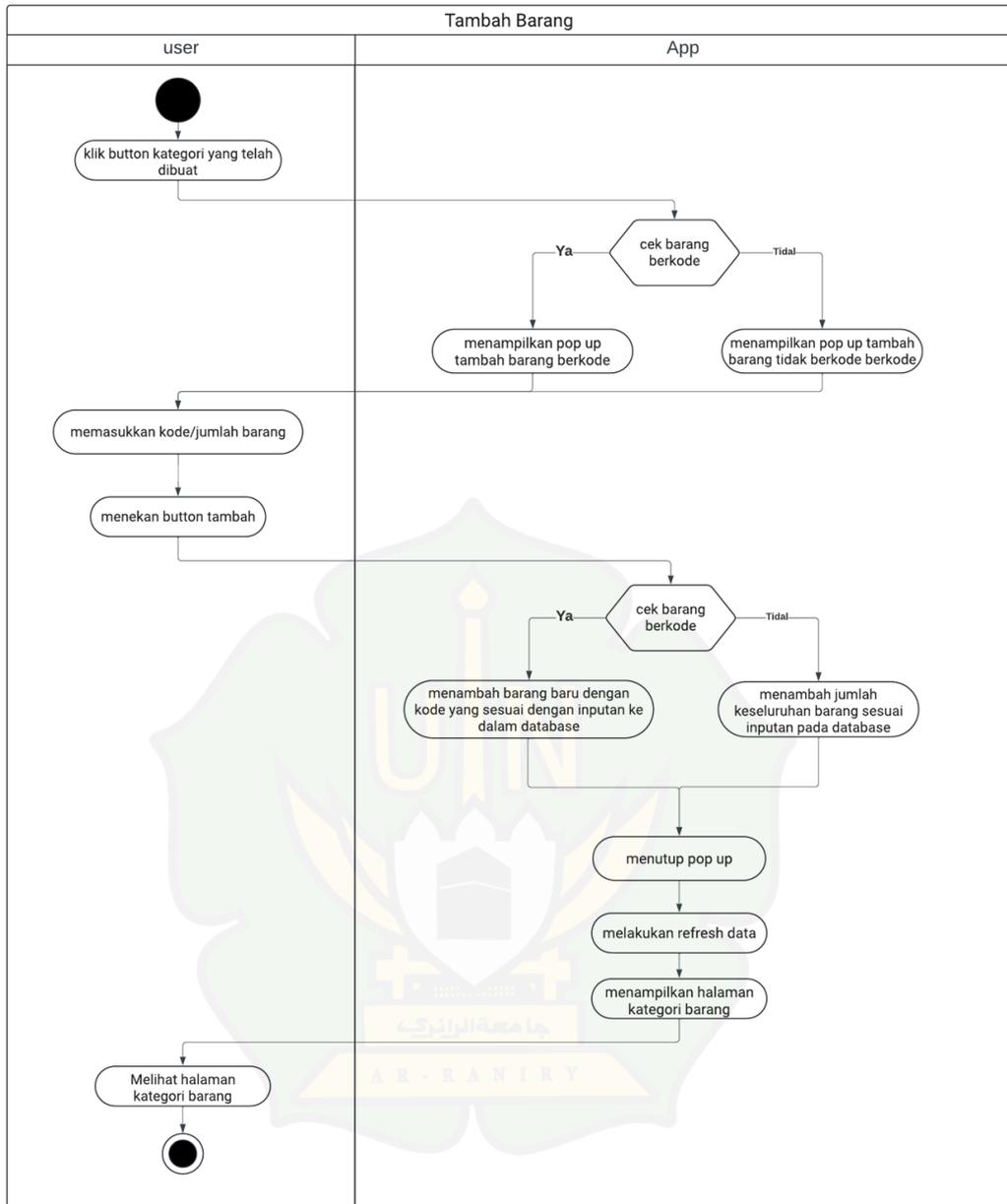
Gambar 4. 5 Activity Diagram Tambah Kategori Barang

Sumber: Data diolah (2024)

b. Tambah Barang

Gambar 4.6 menggambarkan rangkaian aktifitas untuk melakukan tambah barang yang dilakukan *user* di aplikasi *web* peminjaman inventaris laboratorium. Adapun rinciannya ialah sebagai berikut:

- *User* mengklik salah satu *button* dari *list* kategori barang yang telah dibuat sebelumnya.
- Kemudian aplikasi akan melakukan proses pengecekan apakah kategori yang dipilih adalah barang berkode atau tidak. Jika iya, maka aplikasi akan menampilkan *pop up* tambah barang berkode. Dan jika tidak, maka aplikasi akan menampilkan *pop up* tambah barang tidak berkode.
- Jika yang ditampilkan adalah *pop up* barang berkode, maka *user* akan mengisi kode barang. Namun, jika yang di tampilkan adalah *pop up* barang tidak berkode, maka *user* akan mengisi jumlah barang. Kemudian *user* menekan *button* tambah.
- Selanjutnya, aplikasi akan melakukan pengecekan apakah barang yang ditambah barang berkode atau tidak. Jika iya, maka aplikasi akan menambahkan barang baru dengan kode yang sesuai inputan sebelumnya ke dalam *database*. Dan jika tidak, maka aplikasi akan menambahkan jumlah keseluruhan barang sesuai inputan ke dalam *database*. Lalu, aplikasi akan menutup *pop up*, melakukan *refresh* data dan menampilkan halaman kategori barang.



Gambar 4. 6 Activity Diagram Tambah Barang

Sumber: Data diolah (2024)

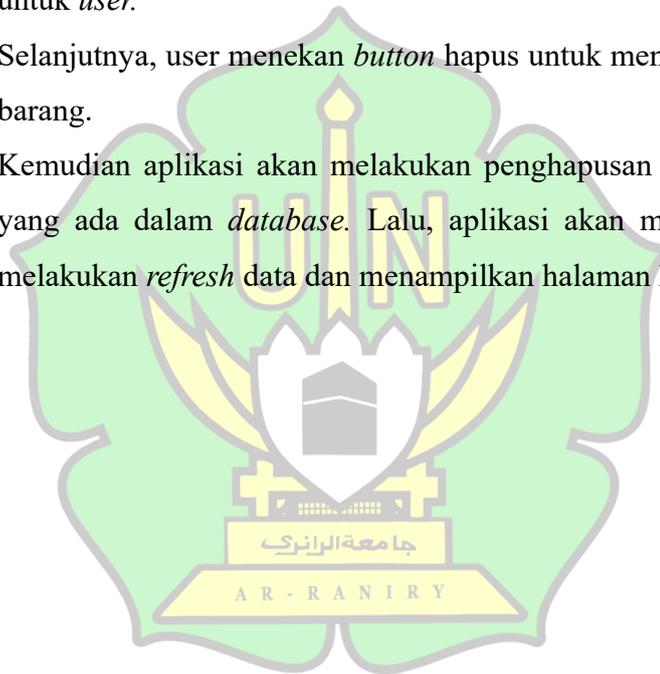
C. Delete

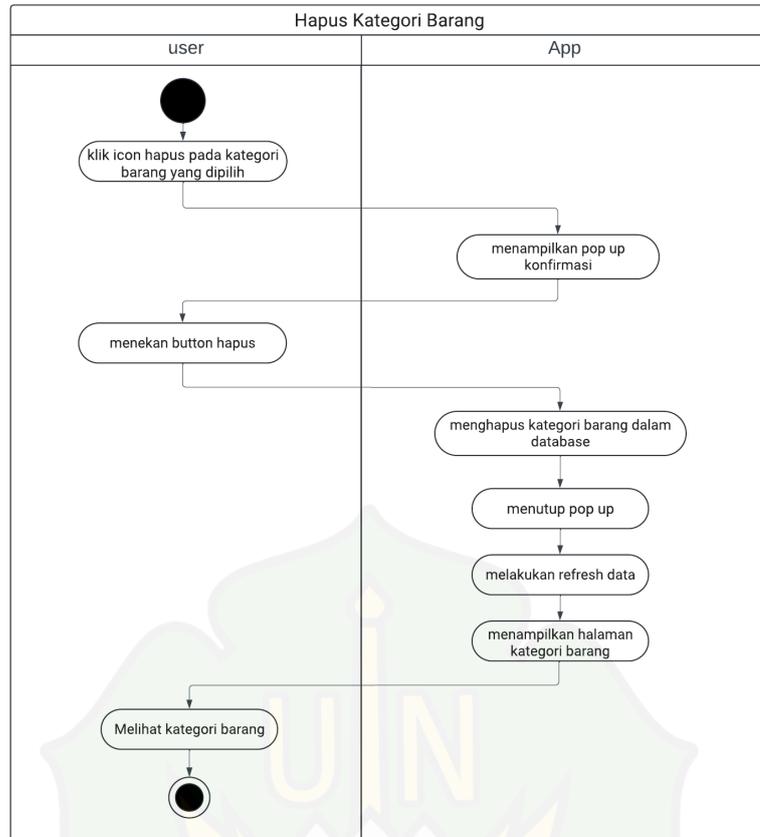
Fitur *delete* dibagi menjadi 2, yaitu hapus kategori barang dan hapus barang. Berikut merupakan *activity diagramnya*,

a. Hapus Kategori Barang

Gambar 4.7 menggambarkan rangkaian aktifitas untuk melakukan hapus kategori barang yang dilakukan *user* di aplikasi *web* peminjaman inventaris laboratorium. Adapun rinciannya ialah sebagai berikut:

- *User* mengklik *icon* hapus pada kategori barang yang telah dipilih.
- Kemudian aplikasi akan menampilkan *pop up notifikasi* konfirmasi untuk *user*.
- Selanjutnya, *user* menekan *button* hapus untuk menghapus kategori barang.
- Kemudian aplikasi akan melakukan penghapusan kategori barang yang ada dalam *database*. Lalu, aplikasi akan menutup *pop up*, melakukan *refresh* data dan menampilkan halaman kategori barang.





Gambar 4. 7 Activity Diagram Hapus Kategori Barang

Sumber: Data diolah (2024)

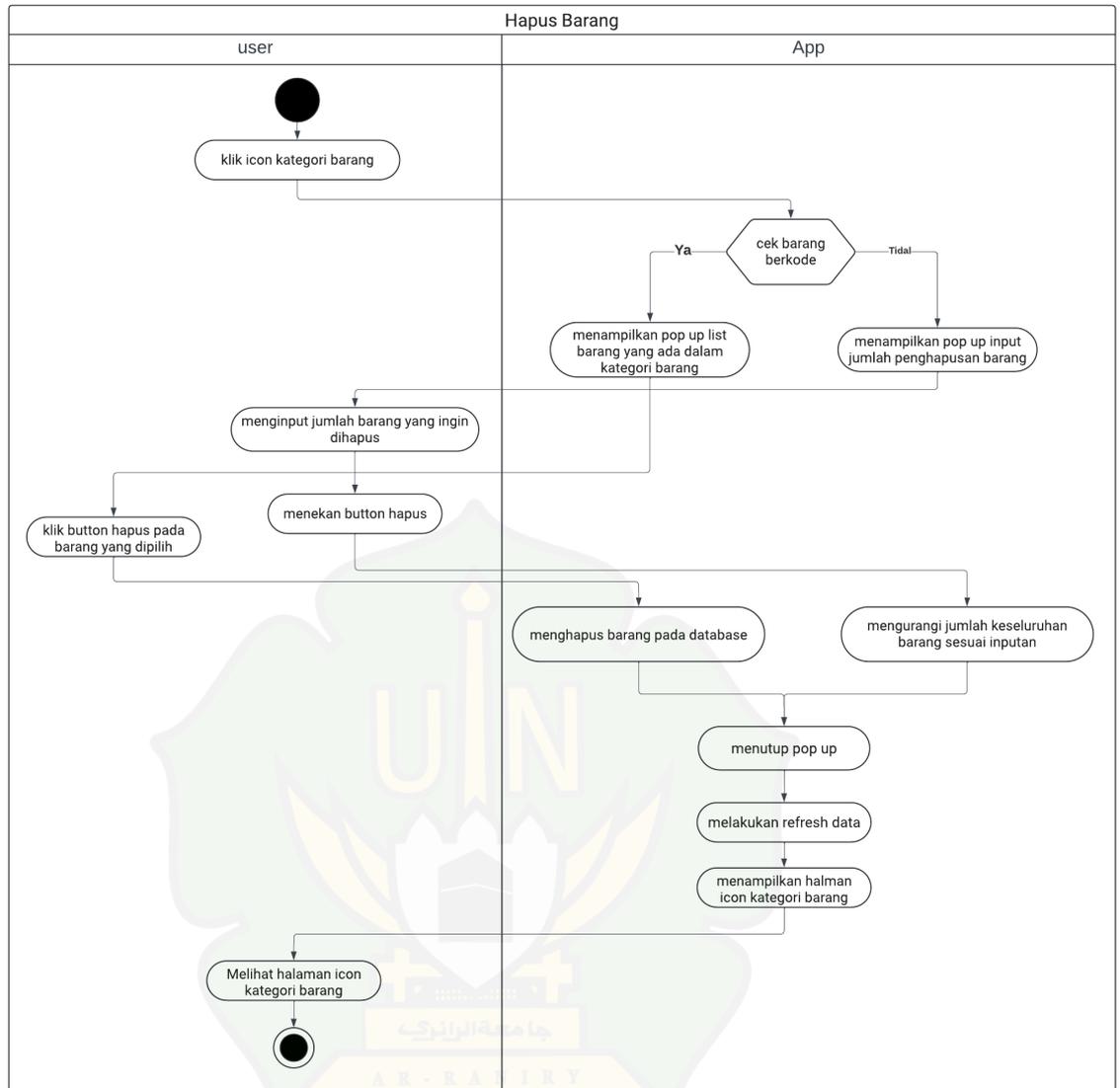
b. Hapus Barang

Gambar 4.8 menggambarkan rangkaian aktifitas untuk melakukan hapus barang yang dilakukan *user* di aplikasi *web* peminjaman inventaris laboratorium. Adapun rinciannya ialah sebagai berikut:

- *User* mengklik *icon* kategori barang untuk memilih barang yang ingin dihapus.
- Kemudian aplikasi akan melakukan proses pengecekan apakah kategori yang dipilih adalah barang berkode atau tidak. Jika iya, maka aplikasi akan menampilkan *pop up list* barang yang ada dalam kategori barang. Dan jika tidak, maka aplikasi akan menampilkan *pop up input* jumlah penghapusan barang.

- Jika yang ditampilkan adalah *pop up* barang berkode, maka *user* akan memilih salah satu dari *list* barang. Namun, jika yang ditampilkan adalah *pop up* barang tidak berkode, maka *user* akan mengisi jumlah barang yang ingin dihapus. Kemudian *user* menekan *button* hapus.
- Jika yang dipilih oleh *user* sebelumnya adalah barang berkode, maka aplikasi akan melakukan proses penghapusan barang pada *database*. Dan jika yang dipilih oleh *user* sebelumnya adalah barang tidak berkode, maka aplikasi akan melakukan proses pengurangan jumlah keseluruhan barang sesuai dengan yang diisi oleh *user* sebelumnya. Kemudian aplikasi akan menutup *pop up*, melakukan *refresh* data dan menampilkan halaman *icon* barang.





Gambar 4. 8 Activity Diagram Hapus Barang

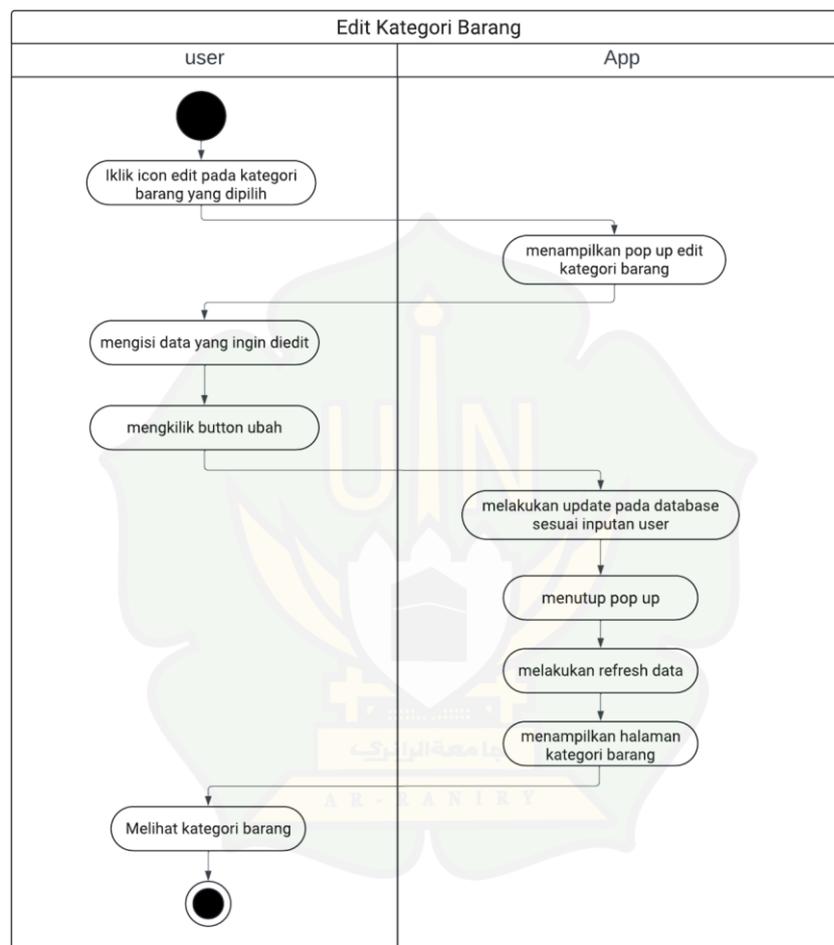
Sumber: Data diolah (2024)

D. Edit

Gambar 4.9 menggambarkan rangkaian aktifitas untuk melakukan edit kategori barang yang dilakukan *user* di aplikasi *web* peminjaman inventaris laboratorium. Adapun rinciannya ialah sebagai berikut:

- *User* mengklik *icon* edit pada kategori barang yang telah dipilih.
- Kemudian aplikasi akan menampilkan *pop up* edit kategori barang.

- Selanjutnya, *user* akan mengisi data yang ingin diedit seperti nama barang baru, *icon* barang baru, dan tipe barang. Lalu *user* akan mengklik *button* ubah.
- Kemudian aplikasi akan melakukan *update* pada *database*. Lalu, menutup *pop up*, melakukan *refresh* data dan menampilkan halaman kategori barang.



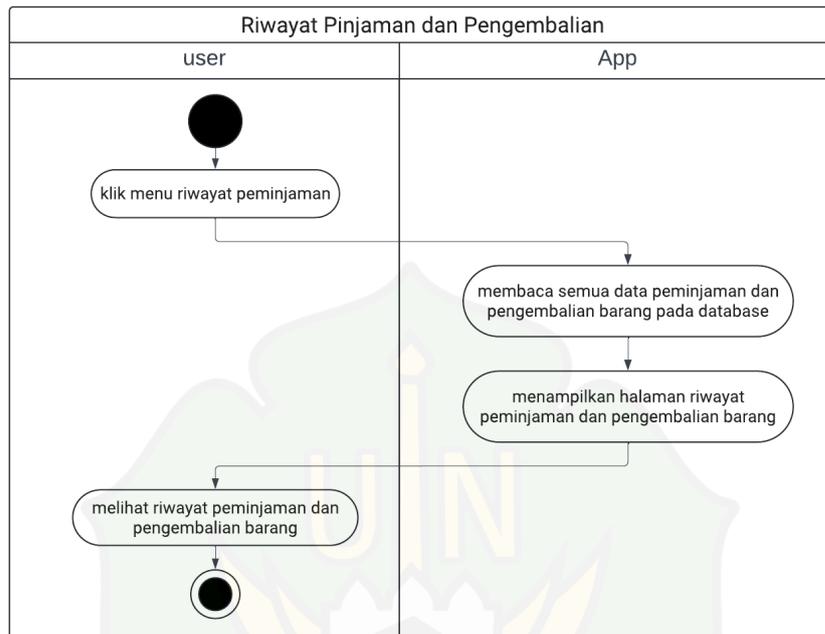
Gambar 4. 9 *Activity Diagram* Edit Barang

Sumber: Data diolah (2024)

E. *Monitoring*

Gambar 4.10 menggambarkan rangkaian aktifitas untuk melihat riwayat dari peminjaman dan pengembalian yang dilakukan *user* di aplikasi *web* peminjaman inventaris laboratorium. Adapun rinciannya ialah sebagai berikut:

- *User* menekan menu riwayat peminjaman
- Kemudian aplikasi akan mengambil semua data peminjaman dan pengembalian barang yang ada dalam *database*, lalu menyajikan data tersebut pada halaman riwayat peminjam.



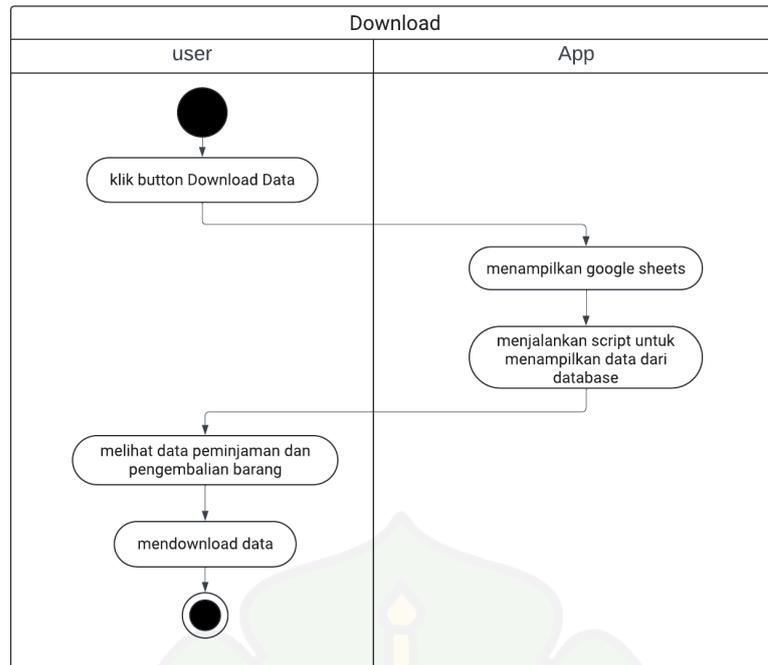
Gambar 4. 10 Activity Diagram Monitoring

Sumber: Data diolah (2024)

F. Download

Gambar 4.11 menggambarkan rangkaian aktifitas untuk *user* admin mengunduh peminjaman dan pengembalian barang yang dilakukan oleh *user* mahasiswa. Adapun rinciannya ialah sebagai berikut:

- *User* mengklik button download data
- Kemudian aplikasi akan menampilkan tab baru berupa google sheets, lalu script pada *apps script* pada *extantion* tersebut akan dijalankan. *Script* tersebut berisi perintah-perintah untuk mengambil data dari *database* kemudian akan ditampilkan pada *google sheets*.
- Kemudian *user* melihat data peminjaman dan pengembalian pada *google sheets*, lalu *user* mendownload data tersebut.



Gambar 4. 11 Activity Diagram Download Database

Sumber: Data diolah (2024)

2. Fitur-Fitur Mahasiswa

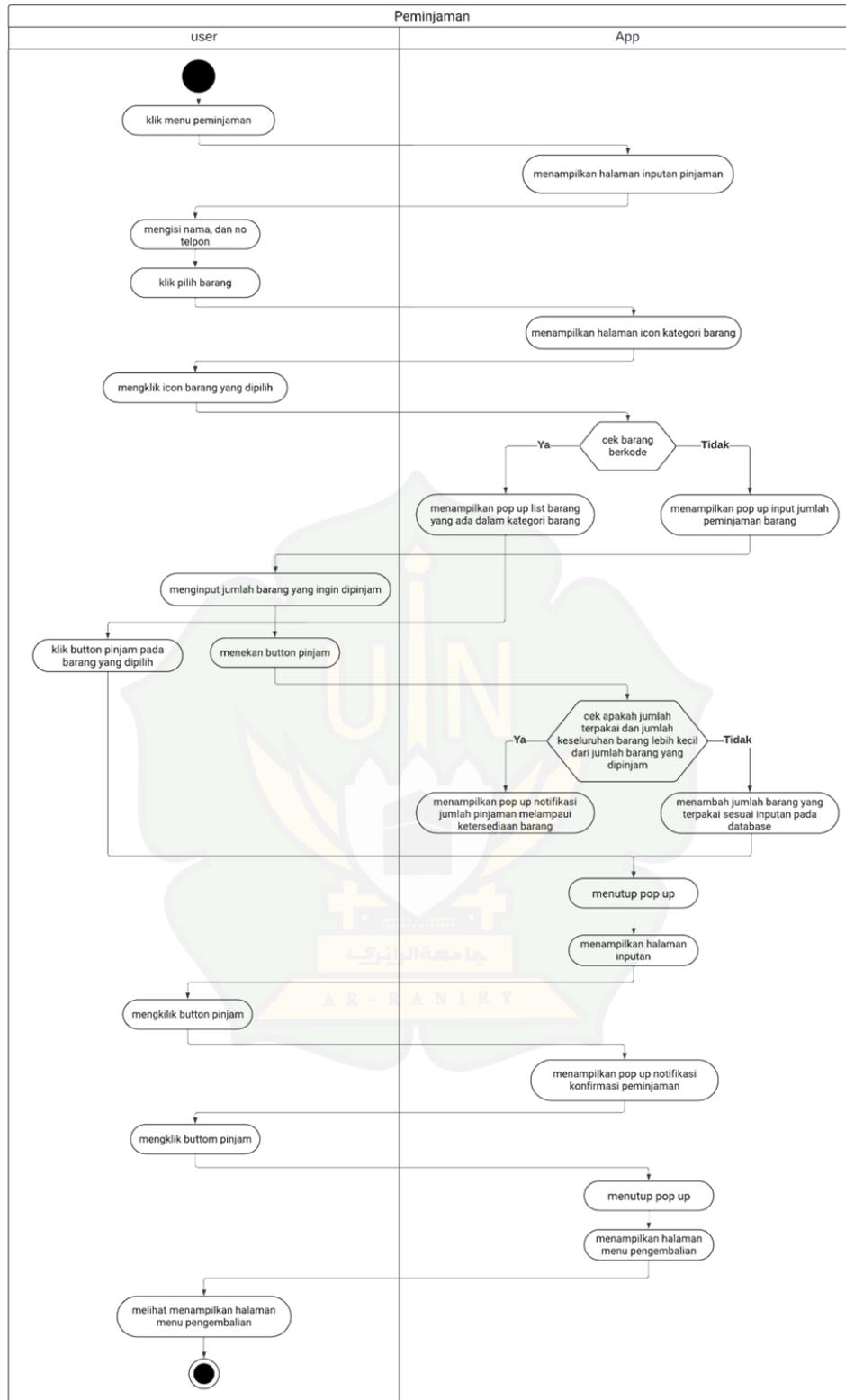
Mahasiswa merupakan tipe user yang tidak melakukan login sama sekali dalam aplikasi *web* peminjaman inventaris laboratorium. Hanya terdapat dua fitur pada user tipe ini, yaitu peminjaman dan pengembalian. Berikut merupakan *activity diagram* dari setiap fitur tersebut.

A. Peminjaman

Gambar 4.12 menggambarkan rangkaian aktifitas untuk melakukan peminjaman barang yang dilakukan *user* di aplikasi *web* peminjaman inventaris laboratorium. Adapun rinciannya ialah sebagai berikut:

- *User* mengklik menu peminjaman
- Aplikasi menampilkan halaman inputan peminjaman.
- *User* mengisi nama no telpon dan mengkilik pilih barang.
- Kemudian aplikasi menampilkan *pop up* halaman *icon* kategori barang
- Lalu *user* mengklik *icon* kategori barang yang ingin dipilih.

- Kemudian aplikasi akan melakukan proses pengecekan apakah kategori yang dipilih adalah barang berkode atau tidak. Jika iya, maka aplikasi akan menampilkan *pop up list* barang yang ada dalam kategori barang. Dan jika tidak, maka aplikasi akan menampilkan *pop up input* jumlah pinjaman barang.
- Jika yang ditampilkan adalah *pop up* barang berkode, maka *user* akan memilih salah satu dari *list* barang. Namun, jika yang ditampilkan adalah *pop up* barang tidak berkode, maka *user* akan mengisi jumlah barang yang ingin dipinjam. Kemudian *user* menekan *button* pinjam.
- jika *user* meminjam barang tidak berkode, maka aplikasi akan melakukan pengecekan apakah jumlah yang diisi oleh *user* tidak melebihi barang yang tersisa untuk dipinjam. Jika melebihi maka, akan muncul *pop up notifikasi* jumlah peminjaman melebihi jumlah ketersediaan barang. Dan jika tidak, maka aplikasi akan menambahkan jumlah barang yang terpakai di *database* sesuai dengan yang diisi oleh *user*. Kemudian aplikasi akan menutup *pop up* dan menampilkan halaman inputan.
- Lalu, *user* mengklik *button* pinjam.
- Kemudian aplikasi akan menampilkan *pop up notifikasi* konfirmasi peminjaman barang.
- Selanjutnya *user* mengklik *button* pinjam.
- Kemudian aplikasi akan menampilkan halaman menu peminjaman.



Gambar 4. 12 Activity Diagram peminjaman

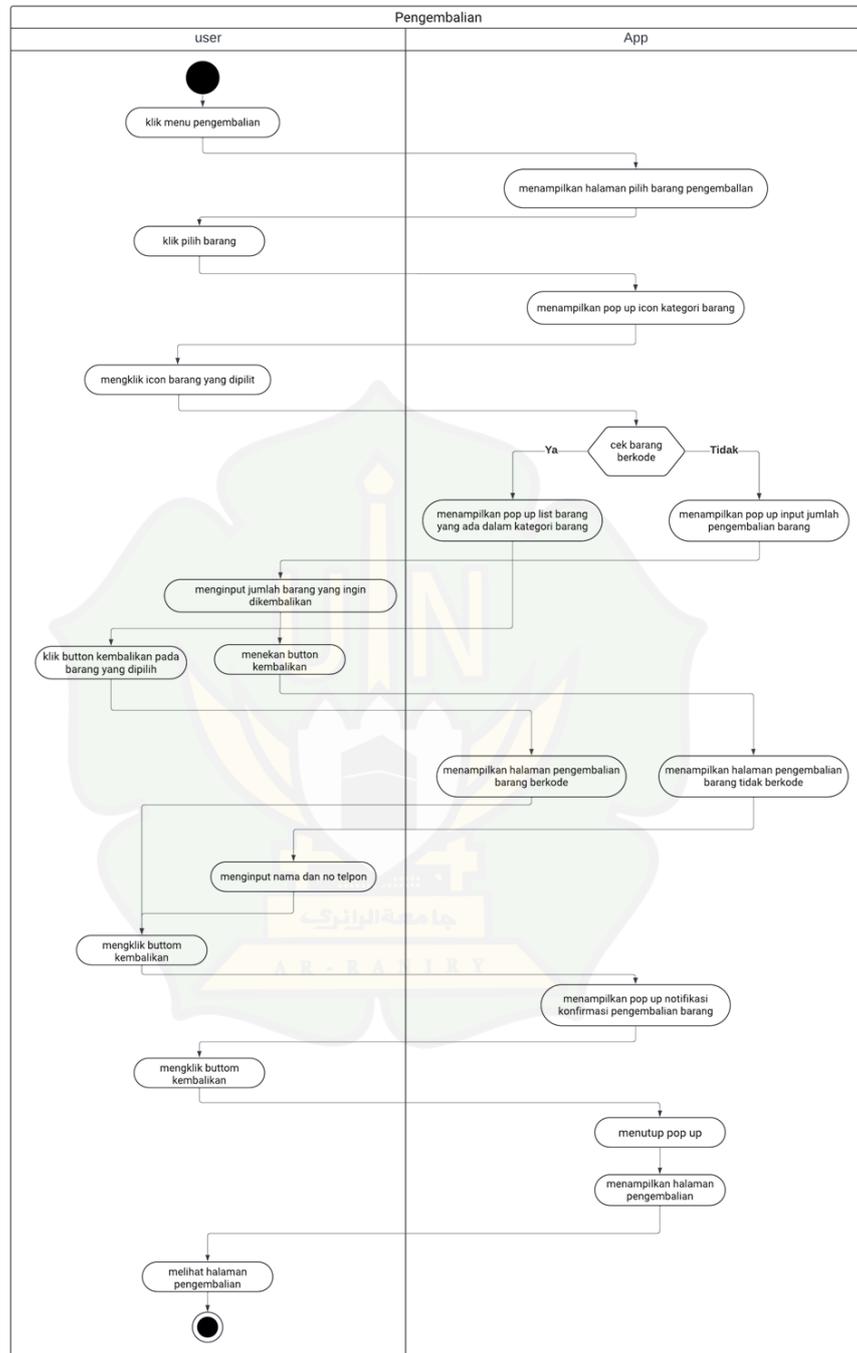
Sumber: Data diolah (2024)

B. Pengembalian

Gambar 4.13 menggambarkan rangkaian aktifitas untuk melakukan pengembalian barang yang dilakukan *user* di aplikasi *web* peminjaman inventaris laboratorium. Adapun rinciannya ialah sebagai berikut:

- *User* mengklik menu pengembalian
- Aplikasi menampilkan halaman pilih barang pengembalian.
- *User* mengklik *button* pilih barang
- Kemudian aplikasi menampilkan *pop up* halaman *icon* kategori barang
- Lalu *user* mengklik *icon* kategori barang yang ingin dipilih.
- Kemudian aplikasi akan melakukan proses pengecekan apakah kategori yang dipilih adalah barang berkode atau tidak. Jika iya, maka aplikasi akan menampilkan *pop up list* barang yang ada dalam kategori barang. Dan jika tidak, maka aplikasi akan menampilkan *pop up input* jumlah pengembalian barang.
- Jika yang ditampilkan adalah *pop up* barang berkode, maka *user* akan memilih salah satu dari *list* barang yang ditampilkan. Namun, jika yang ditampilkan adalah *pop up* barang tidak berkode, maka *user* akan mengisi jumlah barang yang ingin dikembalikan. Kemudian *user* menekan *button* kembalikan.
- Kemudian jika barang yang dikembalikan adalah barang berkode, maka aplikasi akan menampilkan halaman pengembalian barang berkode. Dan jika yang dikembalikan barang tidak berkode maka aplikasi akan menampilkan halaman barang tidak berkode.
- Ketika halaman yang ditampilkan adalah halaman barang tidak berkode, maka *user* akan mengisi nama dan no telpon, lalu mengklik *button* kembalikan. Namun jika yang ditampilkan adalah halaman barang berkode, maka *user* akan langsung mengklik *button* kembalikan.
- Kemudian aplikasi akan menampilkan *pop up notifikasi* pengembalian konfirmasi pengembalian barang.
- Selanjutnya *user* mengklik *button* kembalikan.

- Kemudian aplikasi akan menutup *pop up* dan menampilkan halaman menu pengembalian.



Gambar 4. 13 Activity Diagram Peminjaman

Sumber: Data diolah (2024)

4.3 Implementasi

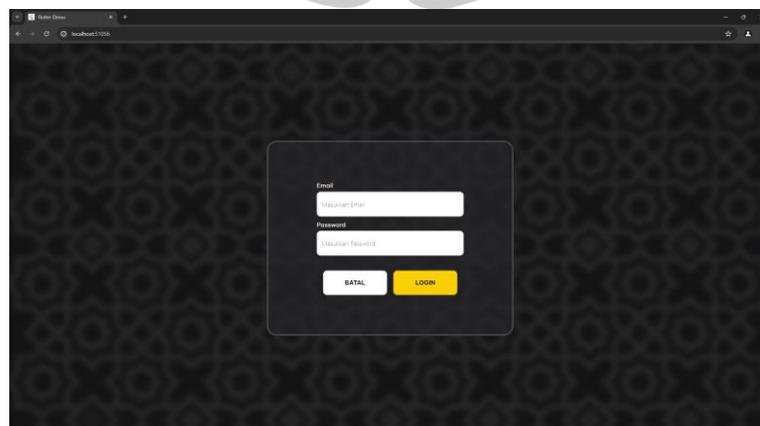
Pada poin ini, hasil dari tahap implementasi disajikan bersama dengan rinciannya. Secara umum, proses implementasi dimulai dari *frontend* dengan pembuatan tampilan dari aplikasi *web* peminjaman inventaris laboratorium, menggunakan bahasa pemrograman Dart dan Flutter. Kemudian dilanjutkan dengan *backend* untuk melakukan pembuatan *database* menggunakan bahasa pemrograman yang sama dengan tahap *frontend*. Hasil implementasi pengembangan aplikasi *web* peminjaman inventaris laboratorium dapat ditemukan pada subbab berikut lengkap dengan rinciannya dalam bentuk tampilan aplikasi dan penggalan kode.

4.3.1 Fitur Authentication

Adapun hasil dari tampilan aplikasi *web* untuk fitur *authentication*, diantaranya adalah halaman *sign in screen*, dan *logout*, Penjelasan lebih lanjut dikemas pada poin-poin berikut.

1. Halaman Sign In Screen

Tampilan pada Gambar 4.14 adalah halaman dari hasil *user story sign in screen* merupakan bagian dari fitur *authentication*. Halaman ini berfungsi untuk proses *login user* admin. Pada halaman ini, *user* admin diminta untuk mengisi dua data, yaitu *email* dan *password*, sebagai kredensial untuk keperluan otentikasi saat melakukan login ke aplikasi.

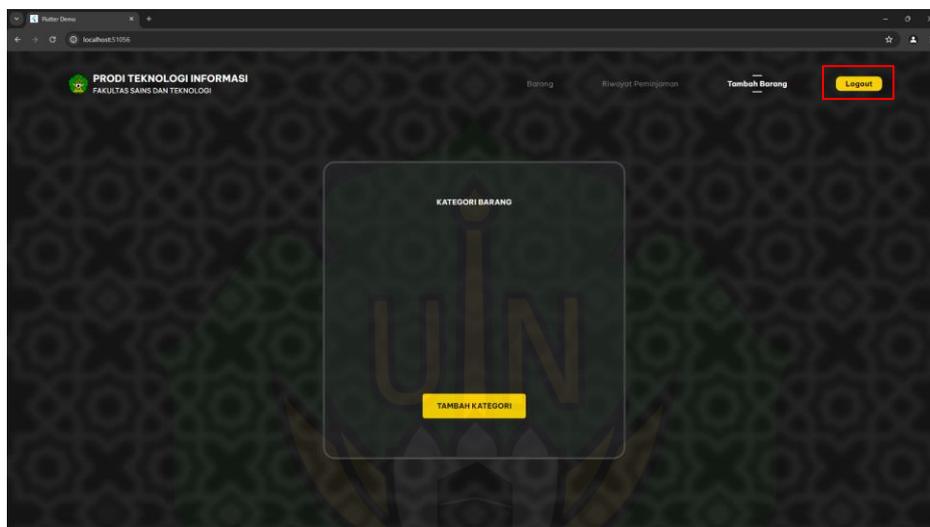


Gambar 4. 14 Hasil Sign In Screen

Sumber: Data diolah (2024)

2. Fitur *Logout*

Tampilan pada Gambar 4.15 adalah halaman dari hasil *user story logout* merupakan bagian dari fitur *authentication*. Halaman ini berfungsi untuk proses *logout user* admin. Pada halaman ini, *user* admin akan mengklik *button* “Logout” yang telah disediakan diatas kanan layar untuk mengganti status *authentication user* dari yang sebelumnya *login* menjadi mahasiswa (tidak *login*).



Gambar 4. 15 Hasil *Logout* Fitur

Sumber: Data diolah (2024)

Adapun hasil dari *backend* aplikasi *web* untuk fitur *authentication* terdapat dua diantaranya adalah halaman *sign in screen*, dan *logout*. Penjelasan lebih lanjut dikemas pada poin-poin berikut.

1. *Sign In*

Gambar 4.16 merupakan *function* untuk menangani fitur *login*. Pada *function* terdapat 2 parameter string yaitu email dan password. Parameter pada *function* ini berfungsi sebagai data yang akan dikirimkan ke *firebase*. Ketika pengiriman data ke *firebase* berhasil serta *email* dan *passwordnya* sesuai dengan data pada *firebase*, maka *function* akan mengembalikan nilai *true*. Nama jika terjadi kesalahan pada

saat pengiriman data, maka akan dikembalikan pesan *error* yang dikirim oleh *firebase*.

```
//login
Future loginWithEmailAndPassword(String email, String password) async {
  try{
    User user =
      (await firebaseAuth.signInWithEmailAndPassword(
        email: email, password: password)).user!;

    if (user != null){
      return true;
    }
  } on FirebaseAuthException catch (e) {
    return e.message;
  }
}
```

Gambar 4. 16 *Backend* Fitur *Sign In*

Sumber: Data diolah (2024)

2. *Logout*

Gambar 4.17 merupakan *function* untuk menangani fitur *logout*. Pada *function* dengan nama *signOut* ini, akan dilakukan *sign out* (*logout*). Ketika *logout* yang dilakukan gagal maka akan dikembalikan pesan *error* yang dikirim oleh *firebase*.

```
//signout
Future signOut() async {
  try{

    await firebaseAuth.signOut();
  } on FirebaseAuthException catch (e) {
    return e.message;
  }
}
```

Gambar 4. 17 *Backend* Fitur *Logout*

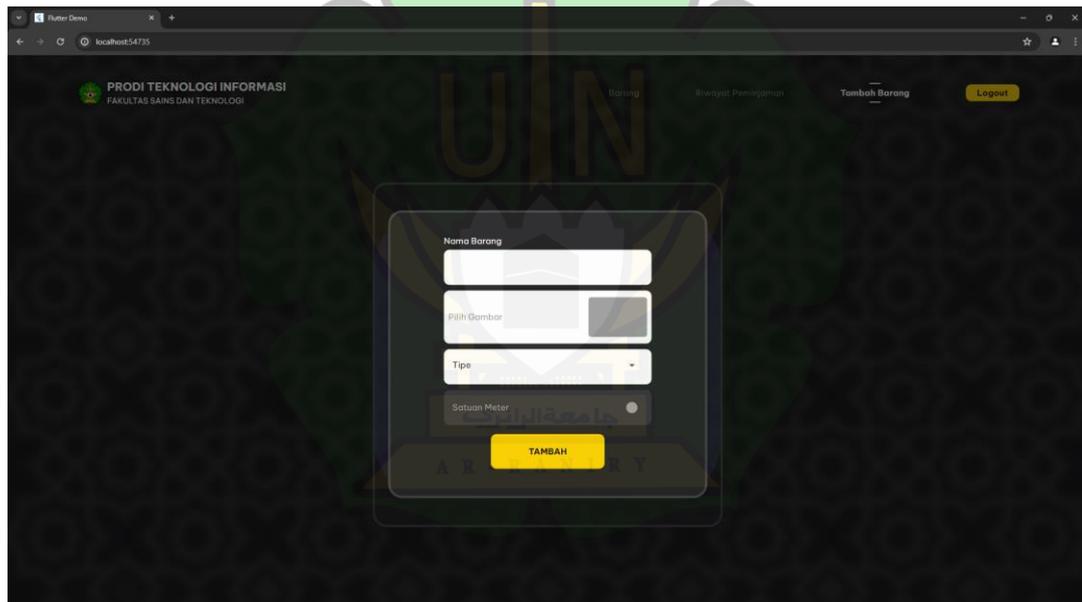
Sumber: Data diolah (2024)

4.3.2 Fitur Add

Adapun hasil dari tampilan aplikasi *web* untuk fitur *add* terdapat dua halaman, diantaranya adalah halaman tambah kategori barang *screen*, dan tambah barang *screen*, Penjelasan lebih lanjut dikemas pada poin-poin berikut.

1. Tambah Kategori Barang *Screen*

Tampilan pada Gambar 4.18 adalah halaman dari hasil *user story* tambah kategori barang *screen* merupakan bagian dari fitur *add*. Halaman ini berfungsi untuk proses penambahan kategori barang yang dilakukan oleh admin. Pada halaman ini, *user* admin diminta untuk mengisi beberapa data untuk penambahan kategori barang, yaitu mengisi nama barang, *icon* barang dan tipe dari barang yang ingin dibuat.



Gambar 4. 18 Hasil Tambah kategori Barang *Screen*

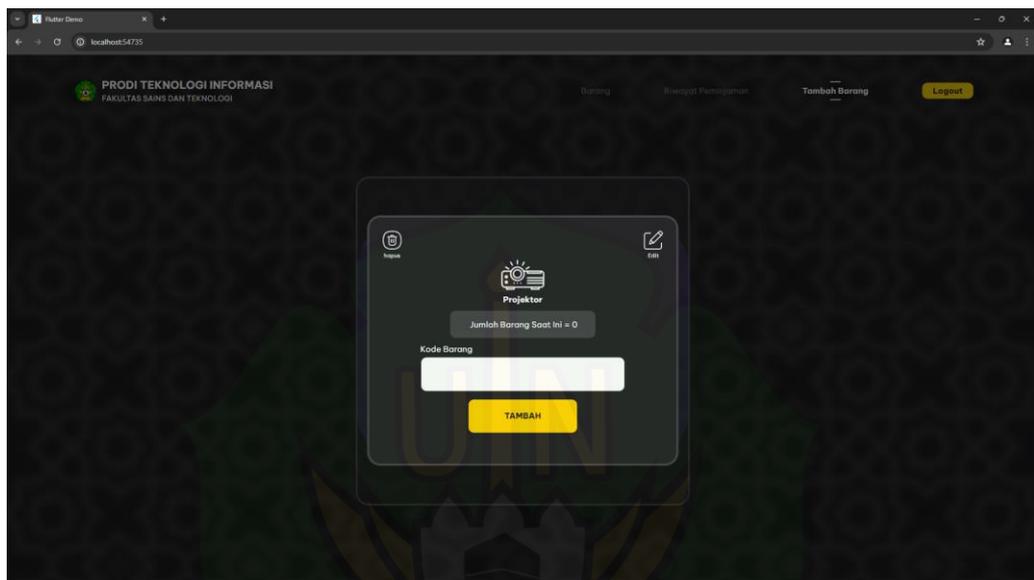
Sumber: Data diolah (2024)

2. Tambah Barang *Screen*

Halaman tambah barang *screen* dibagi menjadi 2 tampilan yaitu tambah barang berkode dan tambah barang tidak berkode.

A. Tambah barang berkode

Tampilan pada Gambar 4.19 adalah halaman dari hasil *user story* tambah barang *screen* merupakan bagian dari fitur *add*. Fungsi dari halaman tambah barang berkode *screen* adalah untuk menambah barang ke dalam kategori barang yang telah dibuat sebelumnya. Pada halaman ini *user* admin diminta untuk mengisi atau menginputkan kode barang untuk di tambahkan kedalam kategori barang.

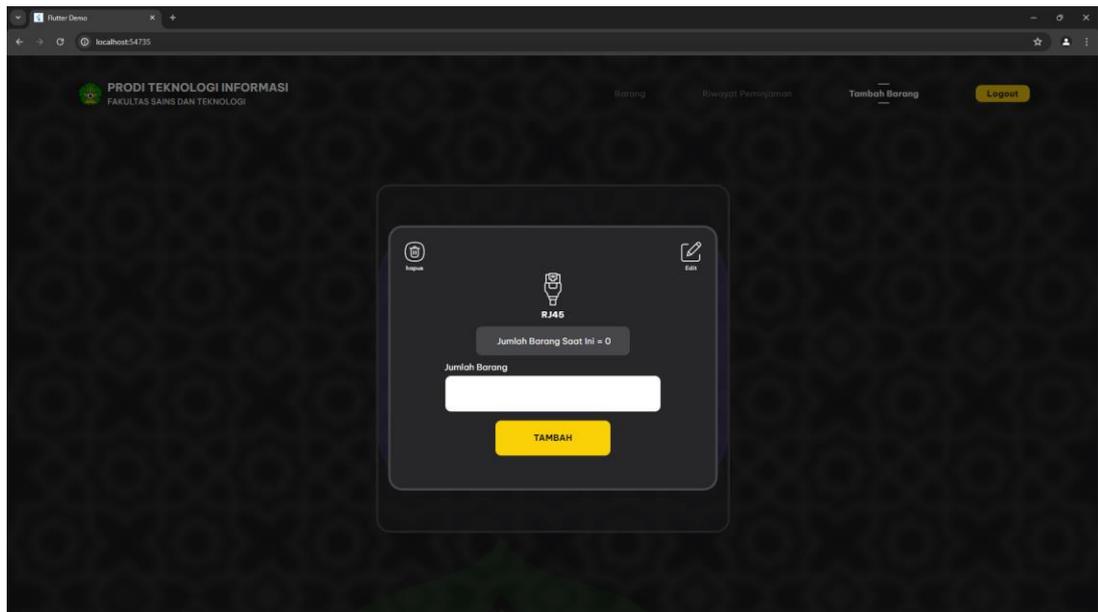


Gambar 4. 19 Hasil Tambah Barang Berkode *Screen*

Sumber: Data diolah (2024)

B. Tambah Barang Tidak Berkode

Gambar 4.20 adalah halaman tambah barang tidak berkode *screen* fungsi dari halaman ini adalah untuk menambah barang ke dalam kategori barang yang telah dibuat sebelumnya. Pada halaman ini *user* admin diminta untuk mengisi atau menginputkan jumlah dari barang untuk ditambahkan kedalam kategori barang.



Gambar 4. 20 Hasil Tambah Barang Tidak Berkode *Screen*

Sumber: Data diolah (2024)

Adapun hasil dari *Backend* aplikasi *web* untuk fitur *Add* terdapat dua diantaranya adalah halaman tambah kategori barang, dan tambah barang. Penjelasan lebih lanjut dikemas pada poin-poin berikut.

1. Tambah kategori Barang

Pada gambar 4.21 merupakan *function* untuk menangani fitur tambah kategori barang. *Function* ini diawali dengan pembuatan *id* untuk dimasukkan ke dalam *database*. Pembuatan *id* ini dilakukan dengan mengambil data *id* paling besar pada *database* kemudian akan menambah 1 untuk *id* berikutnya, ini dilakukan untuk membedakan *id* yang ada pada *database*. Setelah dibuat kemudian akan dicek apakah kategori barang yang dibuat berkode atau tidak. Jika berkode maka “SekaliPakai” dan “SatuanMeter” akan langsung diisi *false* ketika pengiriman data ke *database*. Namun jika kategori barang yang dibuat barang tidak berkode, maka “SekaliPakai” dan “SatuanMeter” diisi sesuai dengan apa yang dipilih oleh *user* admin ketika memilih tipe barang.

```

//tambah Barang
addBarang(String NamaBarang, String Gambar, bool Berkode, bool? SekaliPakai,
    bool SatuanMeter) async {
    int id = 0;
    await barangCollection.get().then((QuerySnapshot snapshot) {
        snapshot.docs.forEach((element) {
            if (int.parse(element['Id'].toString()) >= id) {
                id = int.parse(element['Id'].toString()) +
                    1; //+ 1 untuk memisahkan tiap data yang masuk
            }
        });
    });
    if (Berkode) {
        await barangCollection.doc(id.toString()).set({
            "Nama": NamaBarang,
            "Gambar": Gambar,
            "JumlahTotalBarang": 0,
            "JumlahTerpakai": 0,
            "Berkode": Berkode,
            "Id": id,
            "SekaliPakai": false,
            "SatuanMeter": false
        });
    } else
        (await barangCollection.doc(id.toString()).set({
            "Nama": NamaBarang,
            "Gambar": Gambar,
            "JumlahTotalBarang": 0,
            "JumlahTerpakai": 0,
            "Berkode": Berkode,
            "Id": id,
            "SekaliPakai": SekaliPakai,
            "SatuanMeter": SatuanMeter
        }));
}

```

Gambar 4. 21 *Backend*. Fitur Tambah Kategori Barang

Sumber: Data diolah (2024)

2. Tambah Barang

Tambah barang terbagi menjadi 2 yaitu tambah barang berkode dan tambah barang tidak berkode.

A. Tambah barang berkode

Pada gambar 4.22 merupakan *function* untuk menangani fitur tambah barang berkode. Ketika *function* ini berjalan, akan ada 2 pekerjaan yang dilakukan. Yang pertama, akan di lakukan penambahan data pada database kategori barang yang dipilih. Dan yang kedua akan dilakukan penambahan jumlah barang sebesar satu pada field “JumlahTotalBarang” pada database kategorie barang.

```

//tambah Barang Berkode
EditBarangBerkode(
  int id,
  String KodeBarang,
) async {
  int JumlahBarang = 0;
  DocumentReference DataBarang = await barangCollection.doc(id.toString());

  await barangCollection
    .doc(id.toString())
    .collection('BarangBarang')
    .doc(KodeBarang.toString())
    .set({
      "Kode": KodeBarang,
      "Status": false,
    });
  await barangCollection
    .doc(id.toString())
    .collection('BarangBarang')
    .get()
    .then((QuerySnapshot snapshot) {
      snapshot.docs.forEach((element) {
        JumlahBarang++;
      });
    });
  DataBarang.update({"JumlahTotalBarang": JumlahBarang});
}

```

Gambar 4. 22 *Backend* Fitur Tambah Barang Berkode

Sumber: Data diolah (2024)

B. Tambah barang berkode

Pada gambar 4.23 merupakan *function* untuk menangani fitur tambah barang berkode. Ketika *function* ini berjalan, akan dilakukan penambahan data pada field “JumlahTotalBarang” sebesar jumlah yang diinputkan oleh *user* admin.

```

//tambah Barang Tidak Berkode
EditBarangTidakBerkode(
  int id,
  int JumlahBarang,
) async {
  DocumentReference DataBarang = await barangCollection.doc(id.toString());
  DocumentSnapshot snapshot = await DataBarang.get();
  await DataBarang.update({
    "JumlahTotalBarang":
      JumlahBarang + int.parse(snapshot['JumlahTotalBarang'].toString())
  });
}

```

Gambar 4. 23 *Backend* Fitur Tambah Barang Tidak Berkode

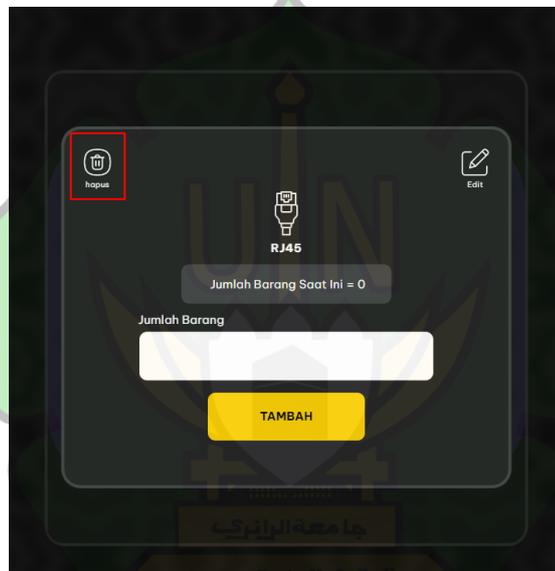
Sumber: Data diolah (2024)

4.3.3 Fitur Delete

Adapun hasil dari tampilan aplikasi *web* untuk fitur *delete* terdapat dua, diantaranya adalah fitur hapus kategori barang, dan halaman hapus barang *Screen*. Penjelasan lebih lanjut dikemas pada poin-poin berikut.

1. Hapus kategori barang

Tampilan pada Gambar 4.24 adalah hasil *user story* hapus kategori barang merupakan bagian dari fitur *delete*. Fitur ini berfungsi untuk menghapus kategori barang yang dibuat sebelumnya oleh admin. Pada halaman ini, *user* admin di minta untuk mngkilk *icon* hapus yang telah disediakan pada kiri atas.



Gambar 4. 24 Hasil Fitur Hapus Kategori Barang

Sumber: Data diolah (2024)

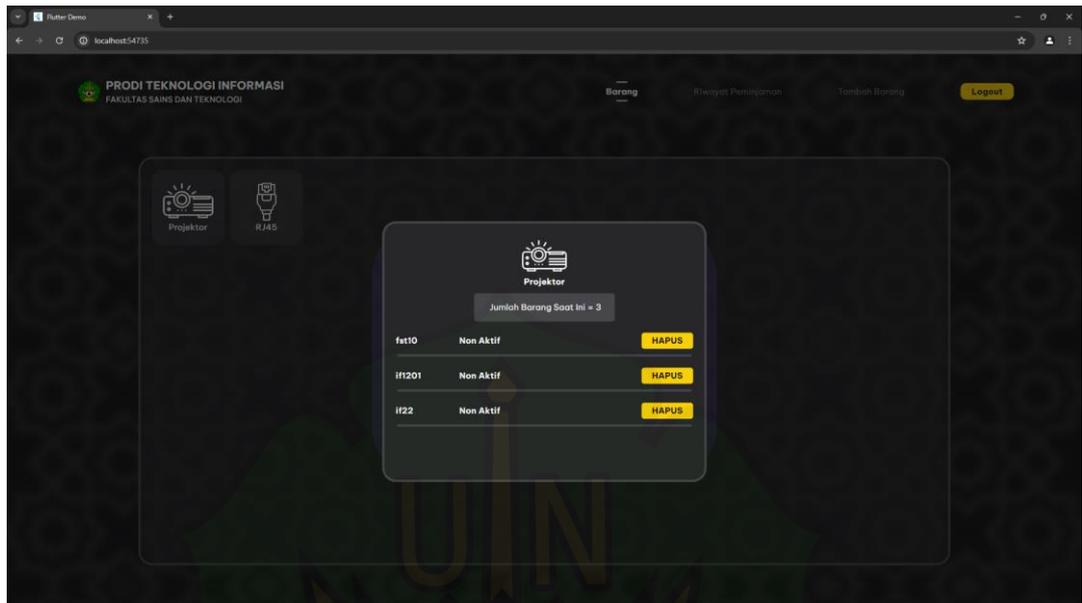
2. Hapus Barang

Halaman hapus barang *screen* dibagi menjadi 2 tampilan yaitu hapus barang berkode dan hapus barang tidak berkode.

A. Hapus Barang Berkode

Tampilan pada Gambar 4.25 adalah halaman dari hasil *user story* hapus barang *screen* merupakan bagian dari fitur *delete*. Fungsi dari halaman hapus barang berkode *screen* adalah untuk menghapus barang yang ada pada kategori

barang tipe barang berkode yang telah di buat sebelumnya. Barang-barang yang ada disajikan dalam bentuk *list*. Pada halaman ini *user* admin diminta untuk mengklik *button* Hapus yang telah disediakan untuk menghapus barang yang diinginkan.

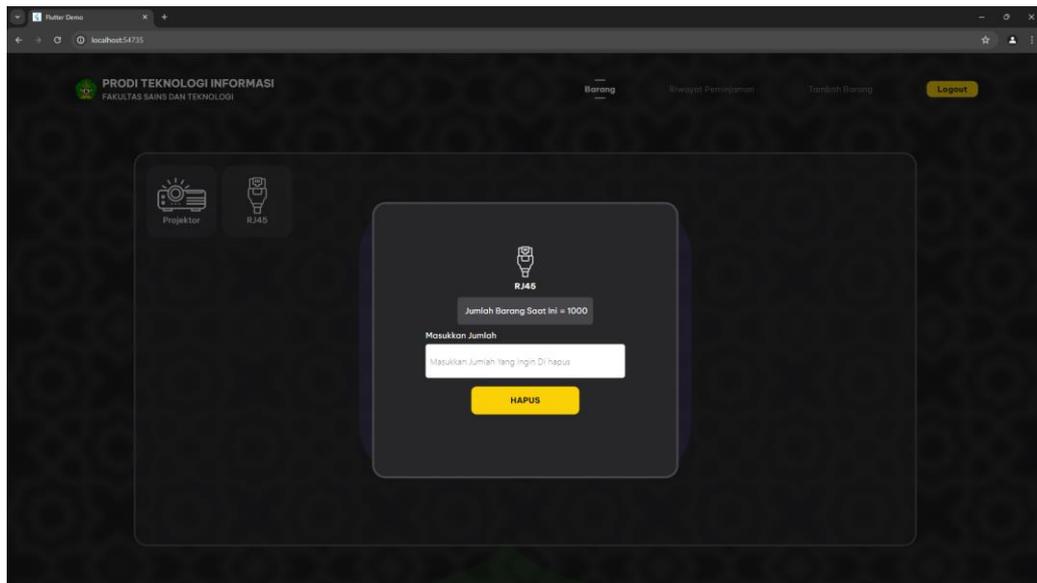


Gambar 4. 25 Hasil Hapus Barang Berkode *Screen*

Sumber: Data diolah (2024)

B. Hapus Barang Tidak berkode

Tampilan pada Gambar 4.26 adalah halaman dari hasil *user story* hapus barang *screen* merupakan bagian dari fitur *delete*. Fungsi dari halaman hapus barang tidak berkode *screen* adalah untuk menghapus barang kategori barang tipenya barang tidak berkode yang telah di buat sebelumnya. Pada halaman ini *user* admin diminta untuk mengisi atau menginputkan jumlah barang yang ingin di hapus.



Gambar 4. 26 Hasil Hapus Barang Tidak Berkode *Screen*

Sumber: Data diolah (2024)

Adapun hasil dari *backend* aplikasi *web* untuk fitur *delete* terdapat dua diantaranya adalah hapus kategori barang, dan hapus barang. Penjelasan lebih lanjut dikemas pada poin-poin berikut.

1. Hapus Kategori Barang

Pada gambar 4.27 merupakan *function* untuk menangani fitur tambah barang berkode. Ketika *function* ini berjalan, hal yang pertama dilakukan adalah menghapus semua barang yang ada dalam kategori barang yang dipilih. Kemudian, setelah kategori barang sudah tidak memiliki barang, maka dilakukan penghapusan kategori barang tersebut. Pengosoangan kategori barang sebelum menghapus kategori barang itu sendiri dilakukan karena adanya masalah ketika dilakukan penghapusan kategori barang yang ada isi barang didalamnya.

```

//hapus Barang semua atau hapus kategori
HapusBarang(int id) async {
  await barangCollection
    .doc(id.toString())
    .collection('BarangBarang')
    .get()
    .then((QuerySnapshot snapshot) {
      snapshot.docs.forEach((element) {
        barangCollection
          .doc(id.toString())
          .collection("BarangBarang")
          .doc(element.id)
          .delete();
      });
    });
  await barangCollection.doc(id.toString()).delete();
}

```

Gambar 4. 27 *Backend* Fitur Hapus Kategori Barang

Sumber: Data diolah (2024)

2. Hapus Barang

Hapus barang terbagi menjadi 2 yaitu hapus barang berkode dan hapus barang tidak berkode.

A. Hapus Barang Berkode

Pada gambar 4.28 merupakan *function* untuk menangani fitur hapus barang berkode. Ketika *function* ini berjalan, hal yang pertama di lakukan adalah menghapus barang pada kategori barang yang dipilih. Kemudian, melakukan pengurangan data sebesar 1 pada *field* “JumlahTotalBarang” di *database*.

```

//hapus Barang berkode
HapusSatuan(int id, String KodeBarang) async {
  await barangCollection
    .doc(id.toString())
    .collection('BarangBarang')
    .doc(KodeBarang)
    .delete();
  print("Penghapusan Berhsil");
  DocumentReference DataBarang = await barangCollection.doc(id.toString());
  DocumentSnapshot snapshot = await DataBarang.get();
  await DataBarang.update(
    {"JumlahTotalBarang": snapshot['JumlahTotalBarang'] - 1});
}

```

Gambar 4. 28 Backend Fitur Hapus Barang Berkode

Sumber: Data diolah (2024)

B. Hapus Barang Tidak Berkode

Pada gambar 4.29 merupakan *function* untuk menangani fitur hapus barang berkode. Ketika *function* ini berjalan, maka proses penghapusan barang tidak berkode akan dilakukan dengan cara mengecek apakah jumlah yang di ingin di hapus itu tidak lebih besar dari barang yang tersedia. Jika iya, maka data pada *field* “JumlahTotalBarang” pada *database* akan dikurangi sebesar jumlah inputan *user* admin. Namun jika tidak, maka penghapusan data tidak akan dilakukan dan sistem menampilkan penghapusan data tidak sesuai.

```

HapusBarangTidakBerkode(int id, int JumlahBarangYangDiHapus) async {
  DocumentSnapshot data = await FirebaseFirestore.instance
    .collection('Barang')
    .doc(id.toString())
    .get();
  if (int.parse(data['JumlahTotalBarang'].toString()) >=
    JumlahBarangYangDiHapus) {
    await FirebaseFirestore.instance
      .collection("Barang")
      .doc(id.toString())
      .update({
        'JumlahTotalBarang': int.parse(data['JumlahTotalBarang'].toString()) -
        JumlahBarangYangDiHapus
      });
  } else {
    print("Penghapusan Data Tidak Sesuai");
  }
}

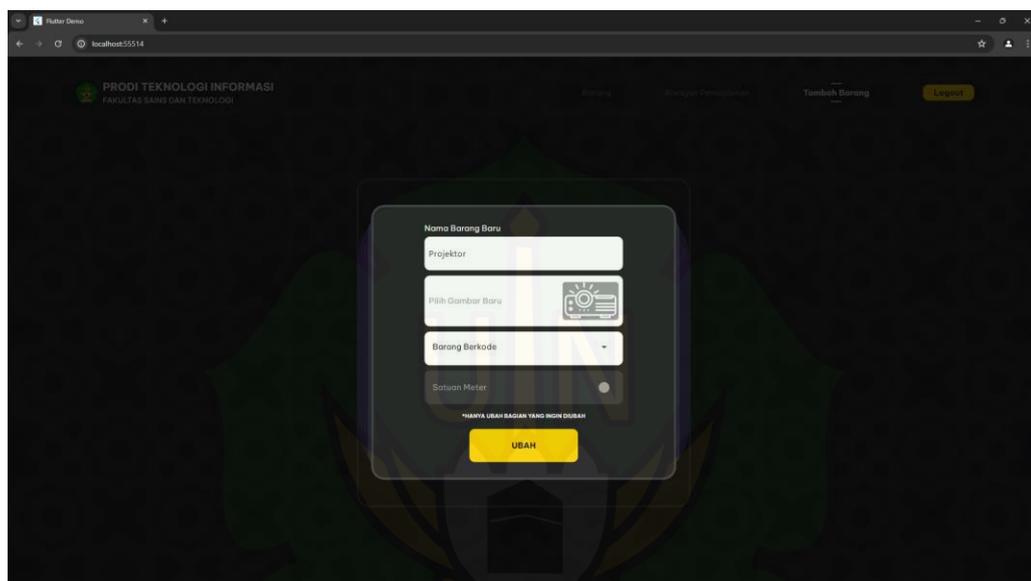
```

Gambar 4. 29 Backend Fitur Hapus Barang Tidak Berkode

Sumber: Data diolah (2024)

4.3.4 Fitur Edit

Tampilan pada Gambar 4.30 adalah halaman dari hasil *user story* edit kategori barang *screen* merupakan bagian dari fitur edit. Fungsi dari fitur edit kategori barang adalah untuk mengedit atau mengupdate data yang ada pada kategori barang yang telah dibuat sebelumnya. Pada halaman ini *user* admin bisa mengedit beberapa data seperti nama barang baru, *icon* barang dan tipe dari barang tersebut.



Gambar 4. 30 Hasil Edit kategori Barang *Screen*

Sumber: Data diolah (2024)

Gambar 4.31 merupakan *function* dari *Backend* aplikasi *web* untuk fitur edit. Ketika *function* ini berjalan, akan di lakukan *update* data pada *id* kategori barang yang dipilih, semua data dalam *field* kategori barang yang diedit akan diubah sesuai dengan parameter pada *function* ini. Jika ada perubahan data pada *field* “Berkode” ketika update, maka isi *field* “JumlahTotalBarang” dan *field* “JumlahTerpakai” akan diubah menjadi 0.

```

//edit kategori barang
EditBarang(String NamaBarang, String Gambar, bool Berkode, int id,
bool SatuanMeter, bool sekaliPakai, int jumlahTotalBarang, int JumlahTerpakai) async {
await barangCollection.doc(id.toString()).get().then((value) async {
if (Berkode == value["Berkode"]) {
await barangCollection.doc(id.toString()).update({
"Nama": NamaBarang,
"Gambar": Gambar,
"JumlahTotalBarang": jumlahTotalBarang,
"JumlahTerpakai": JumlahTerpakai,
"Berkode": Berkode,
"SatuanMeter": SatuanMeter,
"SekaliPakai": sekaliPakai,
});
} else {
await barangCollection.doc(id.toString()).update({
"Nama": NamaBarang,
"Gambar": Gambar,
"JumlahTotalBarang": 0,
"JumlahTerpakai": 0,
"Berkode": Berkode,
"SatuanMeter": SatuanMeter,
"SekaliPakai": sekaliPakai,
});
}
});
print("edit berhasil");
}

```

Gambar 4. 31 *Backend* Fitur Edit

Sumber: Data diolah (2024)

4.3.5 *Fitur Monitoring*

Tampilan pada Gambar 4.32 adalah halaman dari hasil *user story* riwayat peminjaman dan pengembalian barang *screen* merupakan bagian dari fitur *monitoring*. Fungsi dari fitur ini adalah untuk melihat riwayat peminjaman dan pengembalian barang dari *user* mahasiswa. Pada halaman ini data yang disajikan dalam bentuk *list*. Pada halaman ini *user* admin bisa melihat seluruh data peminjaman dan pengembalian barang yang telah dilakukan oleh *user* mahasiswa seperti Nama, No Telp, Barang Pinjaman, Jumlah, Kode Barang, Tanggal Pinjaman, Jam Pinjaman, Jumlah Pengembalian, Tanggal Pengembalian dan Jam Pengembalian. Data pada halaman ini disajikan dalam bentuk *list*.

The screenshot shows a web application interface for 'PRODI TEKNOLOGI INFORMASI' (Information Technology Program) at 'FAKULTAS SAINS DAN TEKNOLOGI'. The page displays a table of borrowing records under the 'Riwayat Peminjaman' (Borrowing History) section. The table has 10 columns: Nama Peminjam, No Telpn, Nama Barang, Jumlah Peminjaman, Kode Barang, Jam Peminjaman, Tanggal Peminjaman, Jumlah Pengembalian, Jam Pengembalian, and Tanggal Pengembalian. There are two rows of data. A 'Logout' button is visible in the top right, and a 'Download Data' button is at the bottom right.

Nama Peminjam	No Telpn	Nama Barang	Jumlah Peminjaman	Kode Barang	Jam Peminjaman	Tanggal Peminjaman	Jumlah Pengembalian	Jam Pengembalian	Tanggal Pengembalian
budi	082519570816	Projektor	1	fst10	22:23	23-6-2024	1	22:24	23-6-2024
abdullah	08224749388	Tang Krimping	5	-	22:24	23-6-2024	4	22:26	23-6-2024

Gambar 4. 32 Hasil *Monitoring Screen*

Sumber: Data diolah (2024)

Gambar 4.33 merupakan *function* dari *Backend* aplikasi *Web* untuk fitur *monitoring*. Ketika *function* ini berjalan, akan di lakukan pengambilan data dari database pada collection “Report”. Kemudian, data yang diambil tadi akan dimasukkan ke dalam *class* “DataSementara”, lalu ketika seluruh *field* data telah terambil, maka data yang ada di dalam *class* “DataSementara” dimasukkan ke dalam *dataPeminjam*.

```

mengambilDataReport() async {
  //pengambilan data pada database report satu persatu sesuai dengan class pada datapeminjam
  await FirebaseFirestore.instance
    .collection("Report")
    .orderBy("id", descending: false)
    .get()
    .then((value) => value.docs.forEach((element) {
      DataPeminjam DataSementara = new DataPeminjam();
      DataSementara>NamaPeminjam = element["NamaPeminjam"];
      DataSementara.NoTelpon = element["NoTelpon"];
      DataSementara>NamaBarang = element["NamaBarang"];
      DataSementara.berM = element["berM"];
      DataSementara.percobaanPeminjaman = element["percobaanPeminjaman"];
      //ketika telah berisi
      for (int i = 0; i < (int.parse(element["percobaanPeminjaman"].toString())); i++) {
        DataSementara.KodeBarang.add(element["KodeBarang"][i]);
        DataSementara.JumlahPeminjaman.add(element["JumlahPeminjaman"][i]);
        DataSementara.Jampeminjam.add(element["jamPeminjaman"][i]);
        DataSementara.TanggalPeminjaman.add(element["tanggalPeminjaman"][i]);
      }
      DataSementara.PercobaanPengembalian =
        element["PercobaanPengembalian"];
      //kalo pengembalian user tidak ada maka (-) isinya
      if (int.parse(element["PercobaanPengembalian"].toString()) == 0) {
        DataSementara.jumlahPengembalian.add("-");
        DataSementara.jamPengembalian.add("-");
        DataSementara.tanggalPengembalian.add("-");
      } else {
        //ketika telah berisi
        for (int i = 0; i < int.parse(element["PercobaanPengembalian"].toString()); i++) {
          DataSementara.jumlahPengembalian.add(element["jumlahPengembalian"][i]);
          DataSementara.jamPengembalian.add(element["jamPengembalian"][i]);
          DataSementara.tanggalPengembalian.add(element["tanggalPengembalian"][i]);
        }
      }
      dataPeminjam.add(DataSementara);
    }));
}

```

Gambar 4. 33 Backend Fitur Monitoring

Sumber: Data diolah (2024)

4.3.6 Fitur Download

Tampilan pada Gambar 4.34 adalah halaman riwayat peminjaman yang dibawah kanan terdapat *button* dari fitur *download*. Fungsi dari fitur *download* ini adalah untuk mengunduh *database* dari peminjaman dan pengembalian barang yang dilakukan oleh *user* mahasiswa. Ketika *button* diklik maka akan diarahkan ke google sheets, lalu *user* admin dapat mengunduh semua data yang tersedia.

Nama Peminjam	No Telpn	Nama Barang	Jumlah Peminjaman	Kode Barang	Jam Peminjaman	Tanggal Peminjaman	Jumlah Pengembalian	Jam Pengembalian	Tanggal Pengembalian
budi	082519570816	Projektor	1	fst10	22:23	23-6-2024	1	22:24	23-6-2024
abdullah	08224749388	Tang Krimling	5	-	22:24	23-6-2024	4	22:26	23-6-2024
							1	22:26	23-6-2024

Gambar 4. 34 Hasil Fitur Download

Sumber: Data diolah (2024)

Gambar 4.35 merupakan *function* dari *Backend* aplikasi *Web* untuk fitur *download*. Penggalan kode ini merupakan kode *google script* yang ada pada *extantion apps script google sheets*. Ketika *button* “Download Data” diklik maka akan muncul *tab* baru yang akan mengarah pada *google sheets*. Pada saat *google sheet* telah diload, maka *script* pada *apps script* ini berjalan, sehingga data peminjaman dan pengembalian pada database akan ditampilkan di *google sheets* tersebut. Setiap dokumen *user* baik itu peminjaman dan pengembalian pada database “*Report*” akan diterjemahkan menjadi *row* pada *google sheets*. Gambar 4.36 merupakan hasil pengambilan data dari *apps script*.

```

// function utama
function myFunction() {
  const firestore = getFirestore();
  var activeSheet = SpreadsheetApp.getActiveSpreadsheet();
  var sheet = activeSheet.getActiveSheet();
  //membersihkan sheet
  sheet.clear();
  //mengambil data pada collection report
  const allDocument = firestore.getDocuments("Report");
  membuatHeader();
  //menyusun data pada sheet per row
  for (var i = 0; i < allDocument.length; i++) {
    var myArray = [];
    var data = allDocument[i].fields["NamaPeminjam"];
    myArray.push(data.stringValue);
    data = allDocument[i].fields["NoTelpon"];
    myArray.push(data.stringValue);
    data = allDocument[i].fields["NamaBarang"];
    myArray.push(data.stringValue);
    //DATA ARRAY PEMINJAMAN
    data = allDocument[i].fields["JumlahPeminjaman"].arrayValue.values;
    var dataArray = "";
    for (var j = 0; j < data.length; j++) {
      if (j == data.length - 1) {
        dataArray += findchargroup(data[j].stringValue);
      } else {
        dataArray += findchargroup(data[j].stringValue) + "\n";
      }
    }
    myArray.push(dataArray);
    data = allDocument[i].fields["KodeBarang"].arrayValue.values;
    var dataArray = "";
    for (var j = 0; j < data.length; j++) {
    }
    myArray.push(dataArray);
    data = allDocument[i].fields["jamPeminjaman"].arrayValue.values;
    var dataArray = "";
    for (var j = 0; j < data.length; j++) {
    }
    myArray.push(dataArray);
    data = allDocument[i].fields["tanggalPeminjaman"].arrayValue.values;
    var dataArray = "";
    for (var j = 0; j < data.length; j++) {
    }
    myArray.push(dataArray);
    //DATA ARRAY PENGEMBALIAN
    data = allDocument[i].fields["jumlahPengembalian"].arrayValue.values;
    if (data == undefined) {
    } else {
    }
  }
  //mengeksport data array ke row sheet
  sheet.appendRow(myArray)
  //merapikan format sheet
  Alignment();
}

```

Gambar 4. 35 Backend Fitur Download

Sumber: Data diolah (2024)

	A	B	C	D	E	F	G	H	I	J	K	L
1	Nama Peminjam	No Telepon	Nama Barang	Jumlah Peminjaman	Kode Barang	Jam Peminjaman	Tanggal Peminjaman	Jumlah Pengembalian	Jam Pengembalian	Tanggal Pengembalian		
2	budi	82247479388	Projektor	1	IT22	22.46	23-6-2024	1	22.47	23-6-2024		
3	abdullah	85219578888	Tang Krimping	5	-	22.46	23-6-2024	3	22.47	23-6-2024		
4								2	22.47	23-6-2024		
5												

Gambar 4. 36 Google Sheet Pengambilan Data Dari Database

Sumber: Data diolah (2024)

4.3.7 Fitur Peminjaman

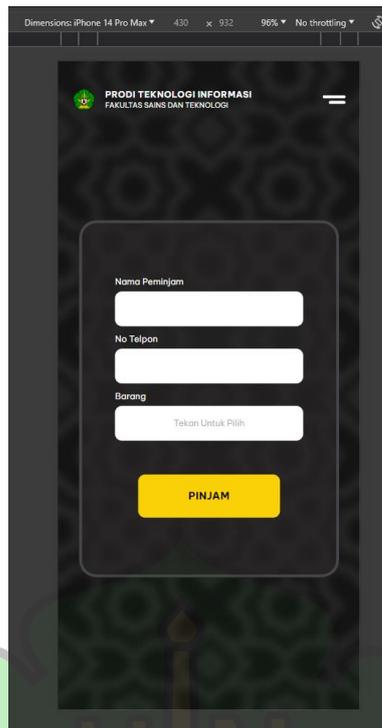
Fitur peminjaman terdapat 2 tampilan *frontend* yaitu tampilan *desktop* dan *mobile*. Fitur peminjaman terbagi menjadi 5 yaitu ketika *user* mahasiswa menginputkan data, memilih kategori barang, memilih barang berkode/tidak berkode dan mengkonfirmasi peminjaman. Penjelasan lebih lanjut dikemas pada poin-poin berikut.

1. Menginput Data pinjaman

Tampilan pada Gambar 4.37 dan 4.38 adalah halaman pertama ketika *user* mahasiswa ingin melakukan peminjaman barang. Pada halaman ini *user* mahasiswa harus mengisi nama dan no telpon terlebih dahulu.

Gambar 4. 37 Hasil Menginput Data Peminjaman Screen Web

Sumber: Data diolah (2024)

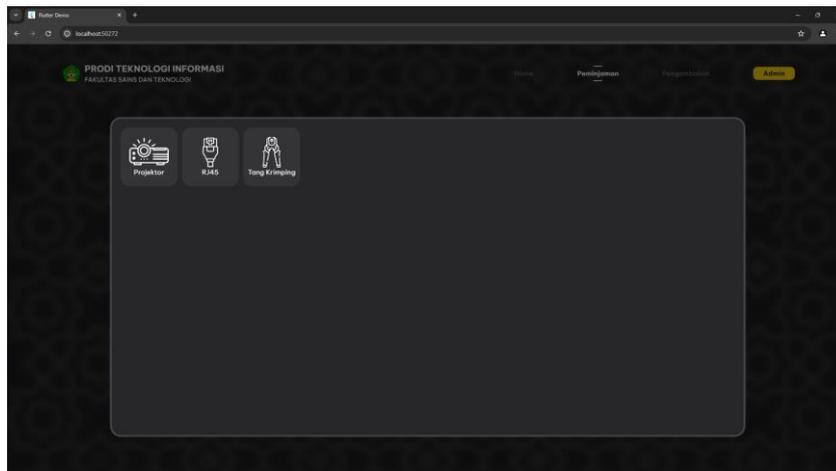


Gambar 4. 38 Hasil Menginput Data Peminjaman *Screen Mobile*

Sumber: Data diolah (2024)

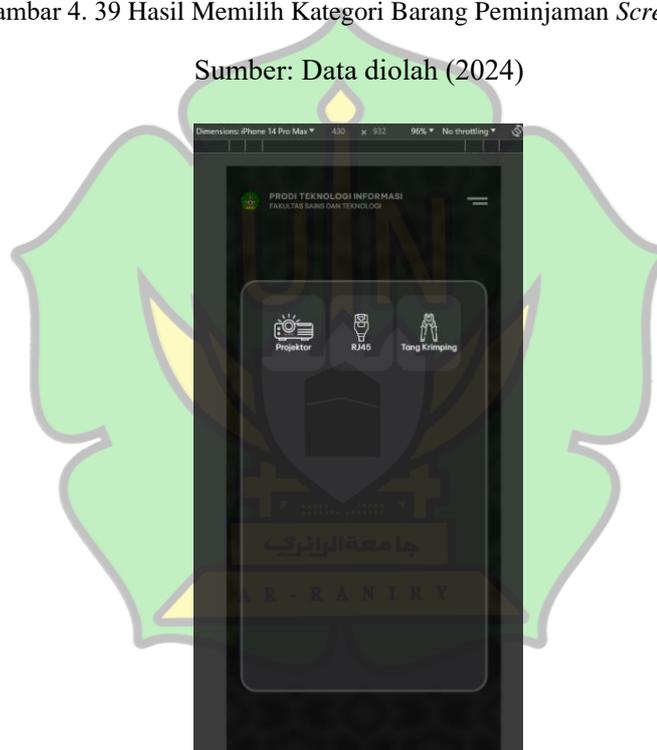
2. Memilih Kategori Barang

Tampilan pada Gambar 4.39 dan 4.40 adalah halaman ke dua ketika *user* mahasiswa meminjam barang. Pada halaman ini terdapat *list* kategori barang yang bisa dipinjam oleh *user* mahasiswa. *User* mahasiswa akan mengklik kategori barang yang ingin dipinjam.



Gambar 4. 39 Hasil Memilih Kategori Barang Peminjaman *Screen Web*

Sumber: Data diolah (2024)



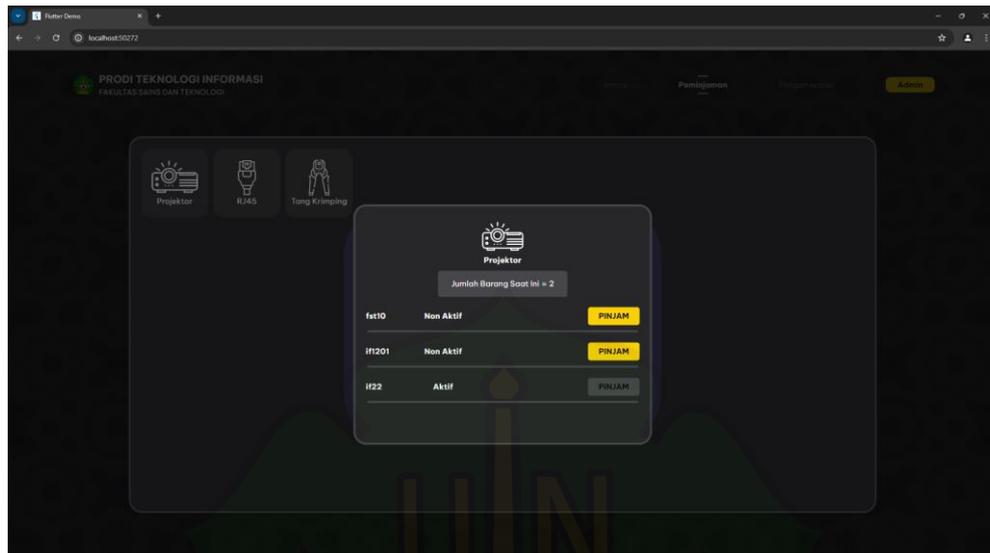
Gambar 4. 40 Hasil Memilih Kategori Barang Peminjaman *Screen Mobile*

Sumber: Data diolah (2024)

3. Memilih Barang Berkode

Tampilan pada Gambar 4.41 dan 4.42 adalah halaman ke tiga ketika *user* mahasiswa memilih kategori barang berkode. Pada halaman ini *user* mahasiswa bisa melihat *list* barang-barang yang tersedia. Di setiap barang memiliki kode barang dan memiliki status barangnya masing-masing. Jika barang sudah dipinjam

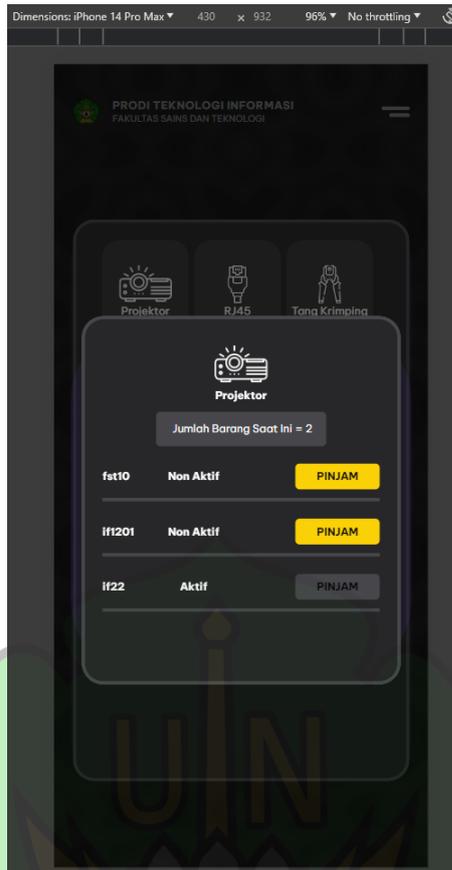
maka akan berstatus Aktif dan *button* Pinjam tidak akan bisa diklik. Dan jika barang belum dipinjam maka statusnya akan Non Aktif dan *button* pinjam bisa diklik, *user* mahasiswa hanya perlu menekan *button* pinjam untuk memilih barang yang ingin dipinjam.



Gambar 4. 41 Hasil Memilih Barang Berkode Ketika Peminjaman *Screen Web*

Sumber: Data diolah (2024)



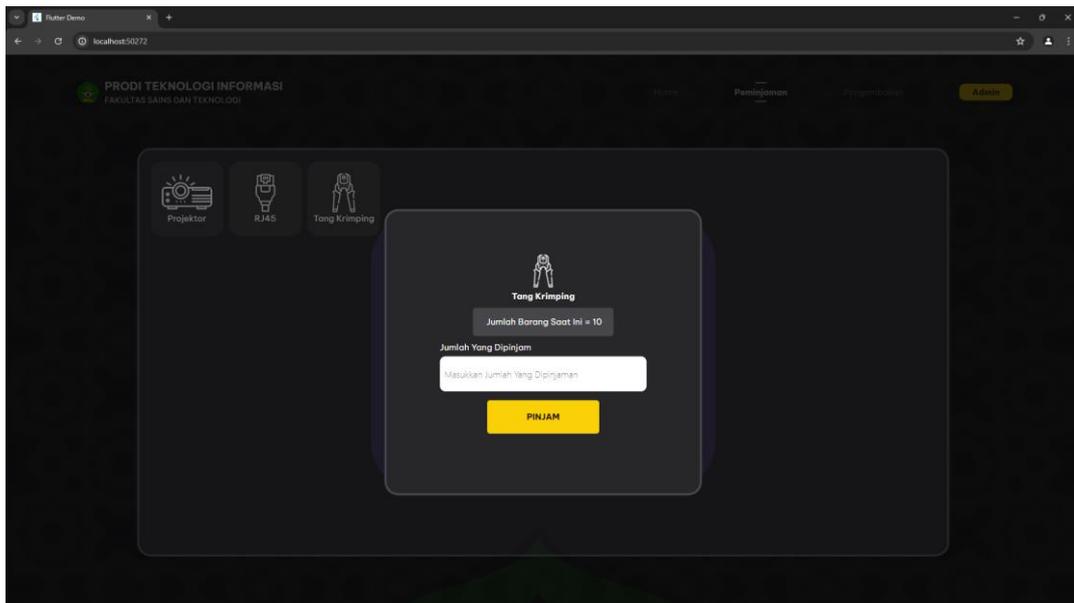


Gambar 4. 42 Hasil Memilih Barang Berkode Ketika Peminjaman *Screen Mobile*

Sumber: Data diolah (2024)

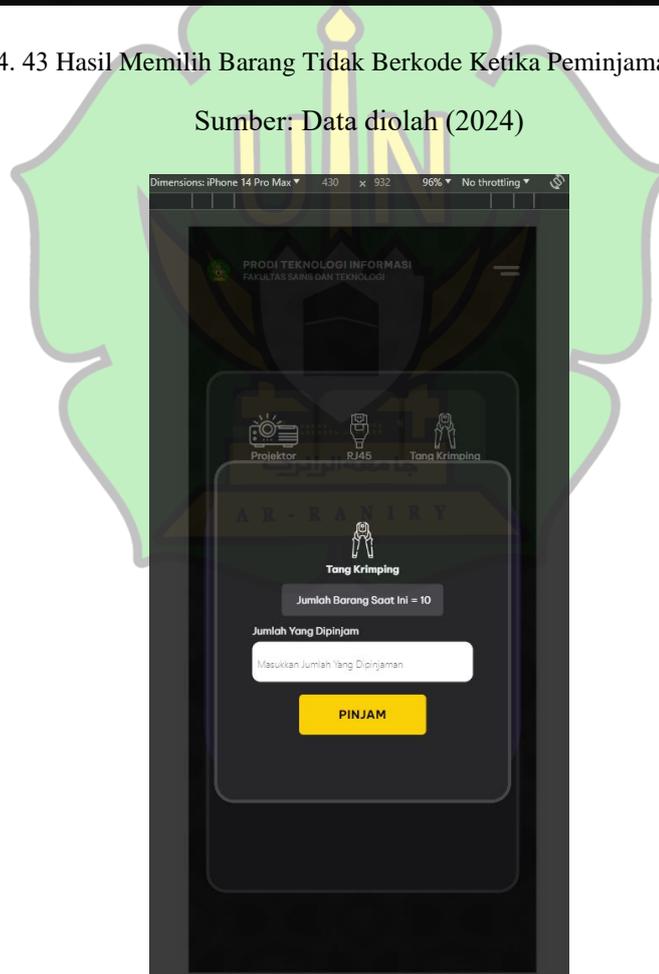
4. Memilih Barang Tidak Berkode

Tampilan pada Gambar 4.43 dan 4.44 adalah halaman ke tiga ketika *user* mahasiswa memilih kategori barang tidak berkode. Pada halaman ini *user* mahasiswa perlu mengisi jumlah barang yang ingin dipinjam, lalu mengklik *button* pinjam yang telah di sediakan.



Gambar 4. 43 Hasil Memilih Barang Tidak Berkode Ketika Peminjaman *Screen Web*

Sumber: Data diolah (2024)

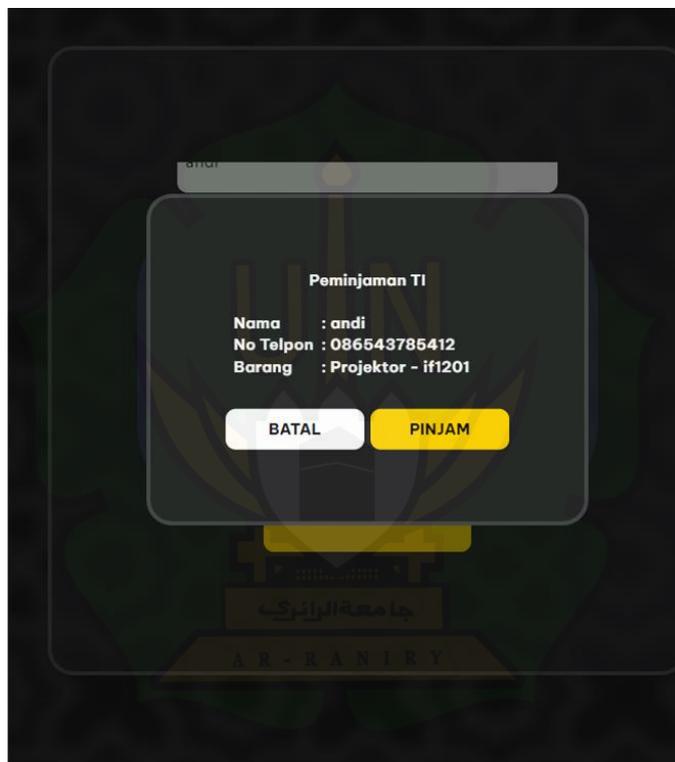


Gambar 4. 44 Hasil Memilih Barang Tidak Berkode Ketika Peminjaman *Screen Mobile*

Sumber: Data diolah (2024)

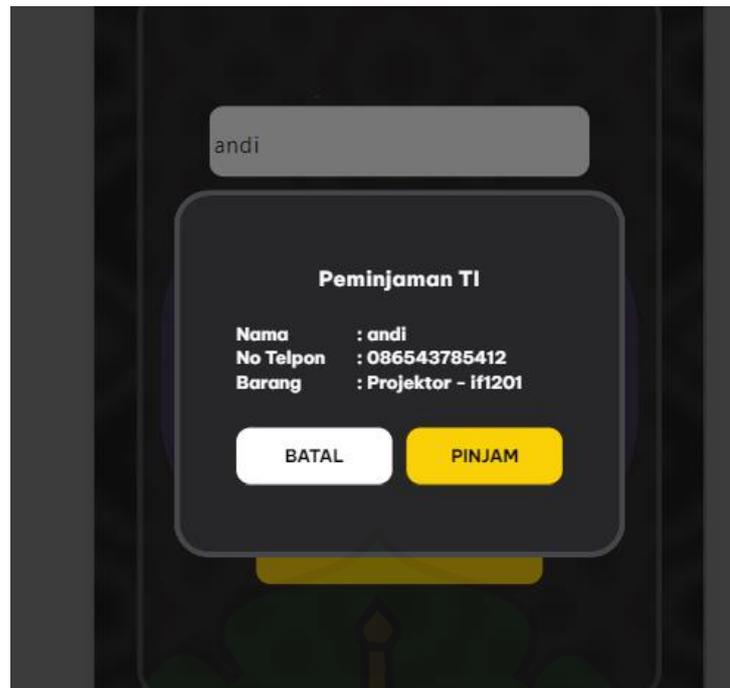
5. Mengkonfirmasi Peminjaman

Tampilan pada Gambar 4.45 dan 4.46 adalah halaman ke empat ketika *user* mahasiswa mengkonfirmasi peminjaman barang yang telah dipilih. Pada halaman ini terdapat beberapa informasi seperti nama, no telepon dan barang yang dipinjam beserta jumlah atau kode dari barang yang dipinjam. Ketika *user* mahasiswa telah yakin ingin meminjam barang tersebut, maka *user* mahasiswa hanya perlu mengklik *button* pinjam yang telah disediakan. Jika tidak yakin maka hanya perlu mengklik *button* batal.



Gambar 4. 45 Hasil Konfirmasi Peminjaman *Screen Web*

Sumber: Data diolah (2024)



Gambar 4. 46 Hasil Konfirmasi Peminjaman *Screen Mobile*

Sumber: Data diolah (2024)

Adapun hasil dari *backend* aplikasi *web* untuk fitur peminjaman terdapat dua diantaranya adalah pinjam barang berkode, dan pinjam barang tidak berkode. Penjelasan lebih lanjut dikemas pada poin-poin berikut.

1. Pinjam Barang Berkode

Pada gambar 4.47 merupakan *function* untuk menangani fitur pinjam barang berkode. Ketika *function* ini berjalan, hal yang pertama di lakukan yaitu mencari barang dengan kode barang yang sama dengan yang dipilih oleh *user* dalam *database*. Jika telah di temukan, maka *field* “JumlahTerpakai” pada kategori barang yang menjadi induk barang yang dipilih akan ditambah satu. Dan *field* “Status” barang pada barang yang telah dipilih akan menjadi *true* (aktif).

```

//pinjam Barang berkode
PinjambarangBerkode(int id, String KodeBarang) async {
  DocumentReference DataBarang = await barangCollection
    .doc(id.toString())
    .collection("BarangBarang")
    .doc(KodeBarang.toString());
  DocumentReference DataBarangsnapshot =
    await barangCollection.doc(id.toString());
  DocumentSnapshot snapshot = await DataBarangsnapshot.get();
  int JumlahTerpakaiSekarang =
    int.parse(snapshot["JumlahTerpakai"].toString()) + 1;
  await DataBarangsnapshot.update({"JumlahTerpakai": JumlahTerpakaiSekarang});
  await DataBarang.update({"Status": true});
}

```

Gambar 4. 47 *Backend* Fitur Pinjam Barang Berkode

Sumber: Data diolah (2024)

2. Pinjam Barang Tidak Berkode

Pada gambar 4.48 merupakan *function* untuk menangani fitur pinjam barang berkode. Ketika *function* ini berjalan, hal yang pertama dilakukan adalah mengambil “JumlahTotalBarang” lalu dikurangi dari “JumlahTerpakai” dalam *field database* barang. Kemudian di lakukan pengecekan apakah barang yang di pinjam oleh user lebih kecil dari jumlah pengurangan yang di lakukan sebelumnya. Jika lebih kecil, maka *field* “JumlahTerpakai” akan ditambah dengan jumlah pinjaman user. Jika tidak lebih kecil, maka akan keluar pesan “Pengambilan Gagal Jumlah Sudah Melampaui Ketersediaan Barang”

```

//pinjam Barang tidak berkode
PinjambarangTidakBerkode(int id, int JumlahBarangDiPinjam) async {
    DocumentReference DataBarang = await barangCollection.doc(id.toString());
    DocumentSnapshot snapshot = await DataBarang.get();
    if (!snapshot["SekaliPakai"]) {
        if (int.parse(snapshot['JumlahTotalBarang'].toString()) -
            int.parse(snapshot['JumlahTerpakai'].toString()) <
            JumlahBarangDiPinjam) {
            print("Pinjaman Jumlah Sudah Melempai Ketersediaan Barang");
        } else {
            int JumlahTerpakaiSekarang =
                int.parse(snapshot["JumlahTerpakai"].toString()) +
                JumlahBarangDiPinjam;
            await DataBarang.update({"JumlahTerpakai": JumlahTerpakaiSekarang});
        }
    } else {
        if (int.parse(snapshot['JumlahTotalBarang'].toString()) <
            JumlahBarangDiPinjam) {
            print("Pengambilan Gagal Jumlah Sudah Melempai Ketersediaan Barang");
        } else {
            int JumlahTotalBarangSekarang =
                int.parse(snapshot["JumlahTotalBarang"].toString()) -
                JumlahBarangDiPinjam;
            await DataBarang.update(
                {"JumlahTotalBarang": JumlahTotalBarangSekarang});
            int JumlahTerpakaiSekarang =
                int.parse(snapshot["JumlahTerpakai"].toString()) +
                JumlahBarangDiPinjam;
            await DataBarang.update({"JumlahTerpakai": JumlahTerpakaiSekarang});
        }
    }
}

```

Gambar 4. 48 *Backend* Fitur Pinjam Barang Tidak Berkode

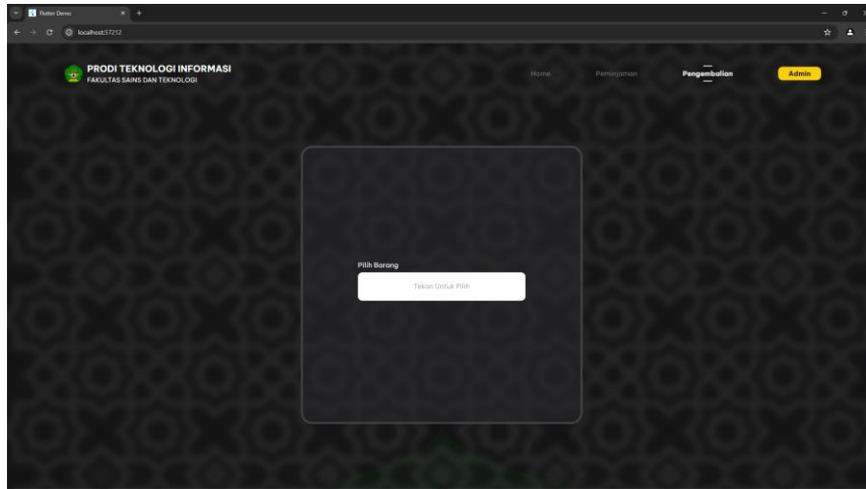
Sumber: Data diolah (2024)

4.3.8 Fitur Pengembalian

Fitur peminjaman terdapat 2 tampilan *frontend* yaitu tampilan *desktop* dan *mobile*. Fitur pengembalian terbagi menjadi 7, yaitu ketika user mahasiswa ingin memilih barang, *list* kategori barang, memilih barang berkode/tidak berkode, telah memilih barang berkode/telah memilih barang tidak berkode dan konfirmasi barang yang ingin dikembalikan. Penjelasan lebih lanjut dikemas pada poin-poin berikut.

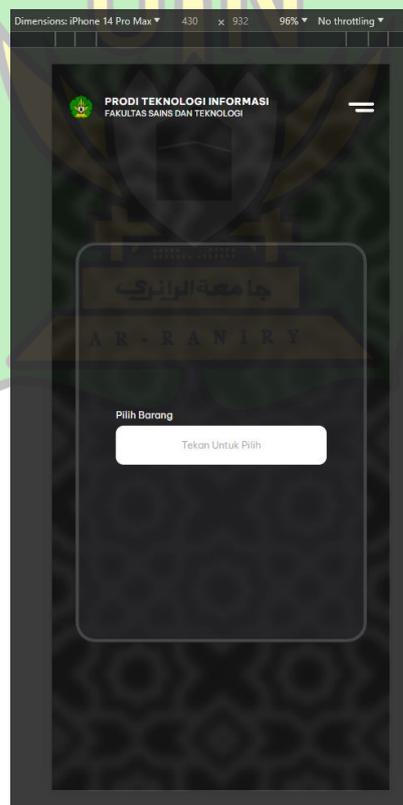
1. Memilih Barang

Tampilan pada Gambar 4.49 dan 4.50 adalah halaman pertama ketika *user* mahasiswa ingin melakukan pengembalian barang. Pada halaman ini *user* mahasiswa hanya perlu mengklik *button* pilih barang yang telah disediakan untuk memilih kategori barang yang ingin dikembalikan.



Gambar 4. 49 Hasil Memilih Barang Pengembalian *Screen Web*

Sumber: Data diolah (2024)

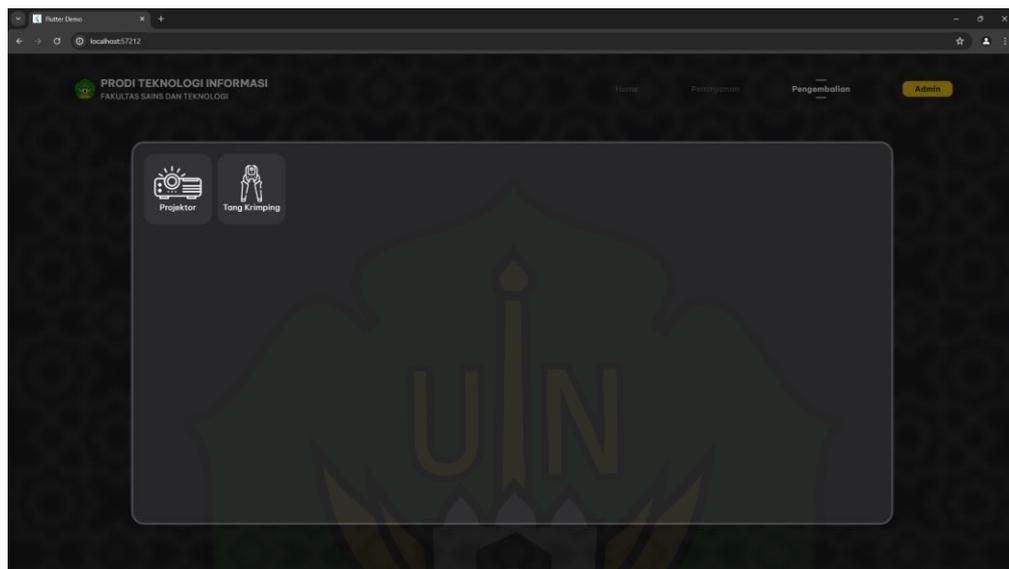


Gambar 4. 50 Hasil Memilih Barang Pengembalian *Screen Mobile*

Sumber: Data diolah (2024)

2. *List* Kategori Barang

Tampilan pada Gambar 4.51 dan 4.52 adalah halaman ke dua ketika *user* mahasiswa melakukan pengembalian barang. Pada halaman ini terdapat *list* kategori barang yang bisa dikembalikan oleh *user* mahasiswa. *User* mahasiswa akan mengklik kategori barang yang ingin dikembalikan sesuai dengan barang yang dipinjam sebelumnya.



Gambar 4. 51 *List* kategori Barang Pengembalian *Screen Web*

Sumber: Data diolah (2024)

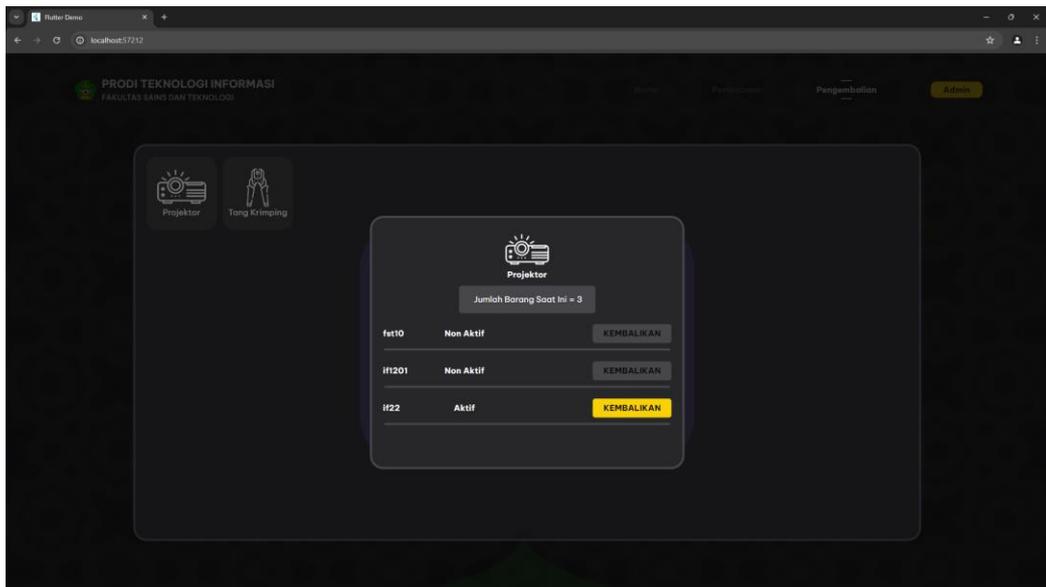


Gambar 4. 52 *List kategori Barang Pengembalian Screen Mobile*

Sumber: Data diolah (2024)

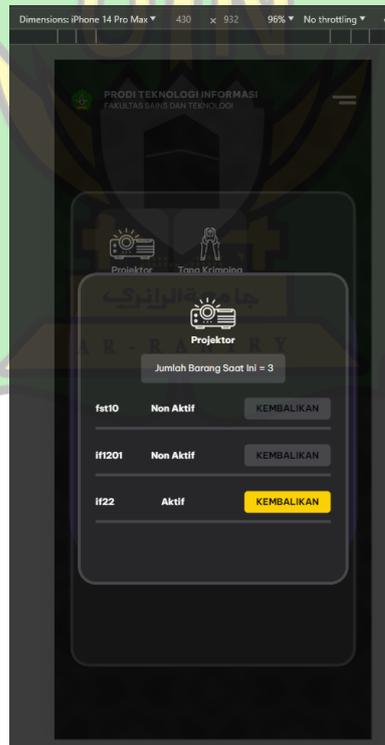
3. Memilih Barang Berkode

Tampilan pada Gambar 4.53 dan 4.54 adalah halaman ke tiga ketika *user* mahasiswa memilih kategori barang berkode untuk dikembalikan. Pada halaman ini *user* mahasiswa bisa melihat *list* barang-barang yang tersedia. Di setiap barang memiliki kode barang dan memiliki status barangnya masing-masing. Jika barang sudah dipinjam maka akan berstatus Aktif dan *button* Kembalikan bisa diklik. Dan jika barang belum dipinjam maka statusnya akan Non Aktif dan *button* kembalikan tidak bisa diklik, *user* mahasiswa hanya perlu menekan *button* kembalikan untuk memilih barang yang ingin dikembalikan.



Gambar 4. 53 Hasil Memilih Barang Berkode Ketika Pengembalian *Screen Web*

Sumber: Data diolah (2024)

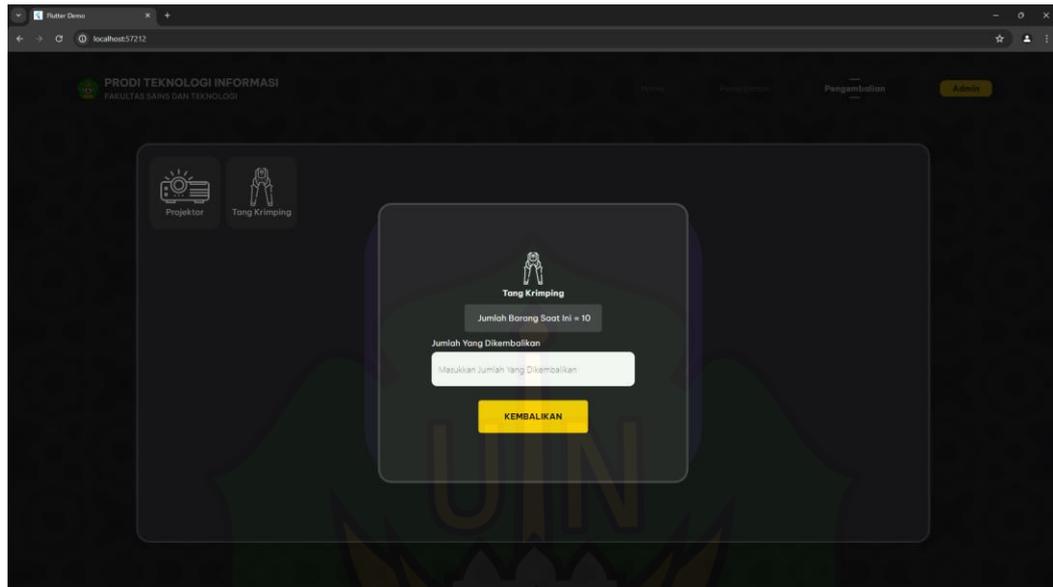


Gambar 4. 54 Hasil Memilih Barang Berkode Ketika Pengembalian *Screen Mobile*

Sumber: Data diolah (2024)

4. Memilih Barang Tidak Berkode

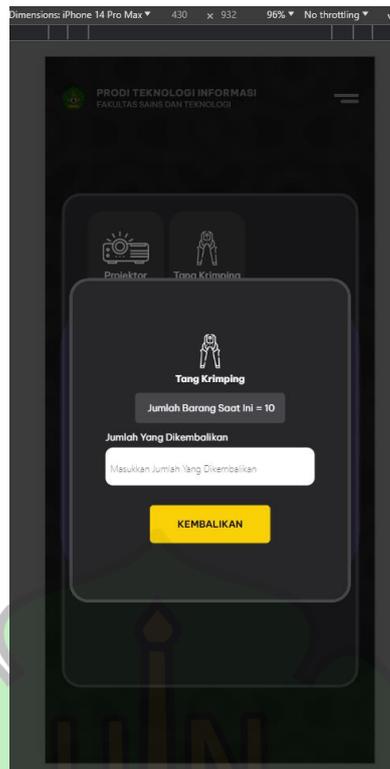
Tampilan pada Gambar 4.55 dan 4.56 adalah halaman ke tiga ketika *user* mahasiswa memilih kategori barang tidak berkode untuk dikembalikan. Pada halaman ini *user* mahasiswa perlu mengisi jumlah barang yang ingin dikembalikan, lalu mengklik *button* kembalikan yang telah di sediakan.



Gambar 4. 55 Hasil Memilih Barang Tidak Berkode Ketika Pengembalian *Screen Web*

Sumber: Data diolah (2024)



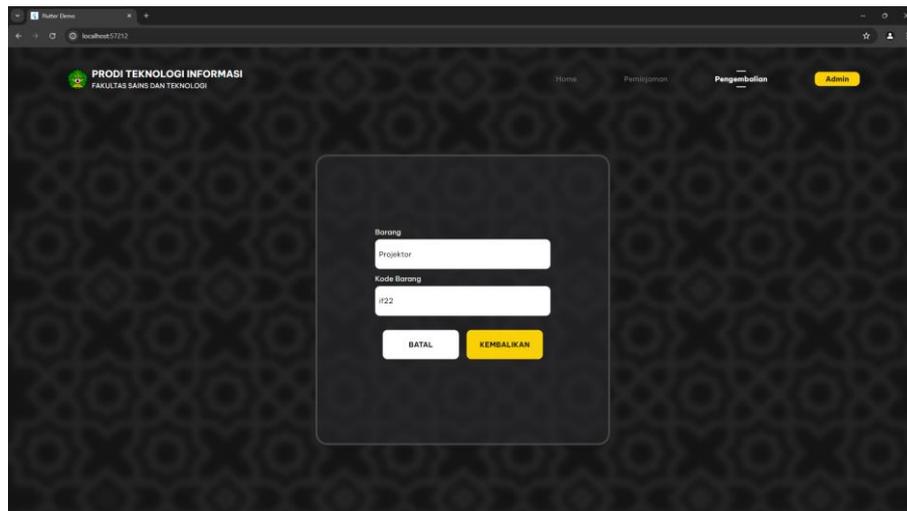


Gambar 4. 56 Hasil Memilih Barang Tidak Berkode Ketika Pengembalian *Screen Mobile*

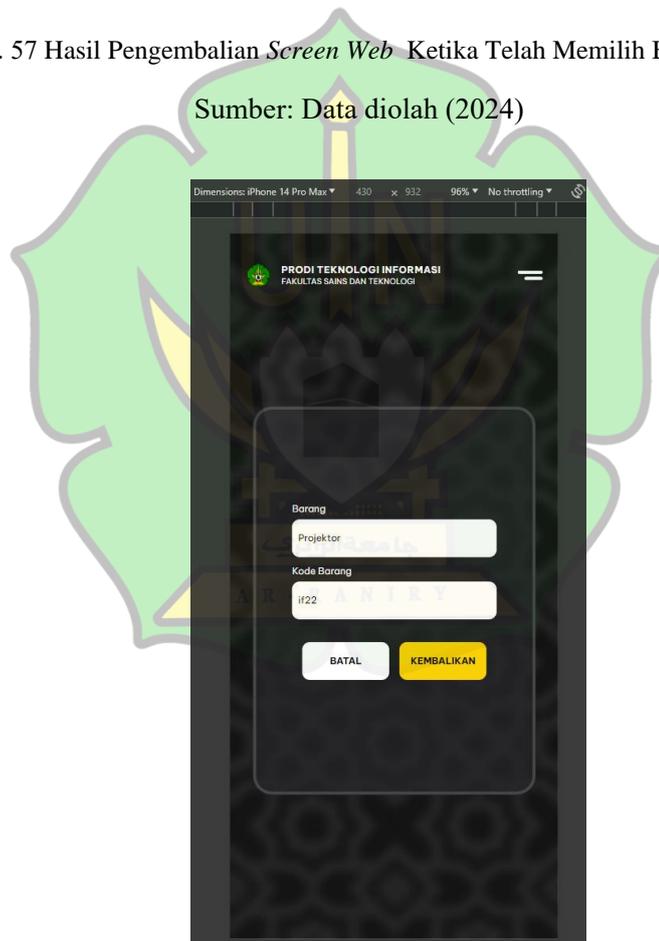
Sumber: Data diolah (2024)

5. Telah Memilih Barang Berkode

Tampilan pada Gambar 4.57 dan 4.58 adalah halaman ke empat ketika *user* mahasiswa telah memilih kategori barang berkode untuk dikembalikan. Pada halaman ini terdapat informasi nama barang yang ingin dikembalikan dan kode barang yang ingin dikembalikan. Jika *user* mahasiswa sudah yakin ingin mengembalikan barang tersebut, maka *user* mahasiswa mengklik *button* kembalikan untuk mengkonfirmasi barang yang ingin dikembalikan. Jika tidak, maka *user* mahasiswa mengklik *button* batal untuk membatalkan pengembalian barang tersebut.



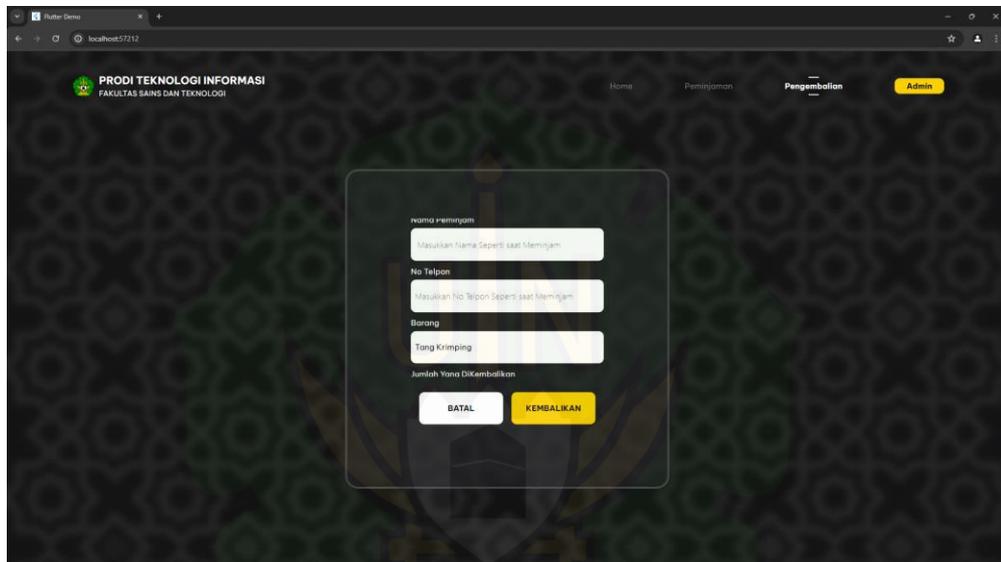
Gambar 4. 57 Hasil Pengembalian *Screen Web* Ketika Telah Memilih Barang Berkode
Sumber: Data diolah (2024)



Gambar 4. 58 Hasil Pengembalian *Screen Mobile* Ketika Telah Memilih Barang Berkode
Sumber: Data diolah (2024)

6. Telah Memilih Barang Tidak Berkode

Tampilan pada Gambar 4.59 dan 4.60 adalah halaman ke empat ketika *user* mahasiswa telah memilih kategori barang tidak berkode untuk dikembalikan. Pada halaman user Mahasiswa perlu mengisi nama dan no telpon untuk dijadikan parameter pencarian data. Jika *user* mahasiswa sudah yakin ingin mengembalikan barang tersebut, maka *user* mahasiswa mengklik *button* kembalikan untuk mengkonfirmasi barang yang ingin dikembalikan. Jika tidak, maka *user* mahasiswa mengklik *button* batal untuk membatalkan pengembalian barang tersebut.



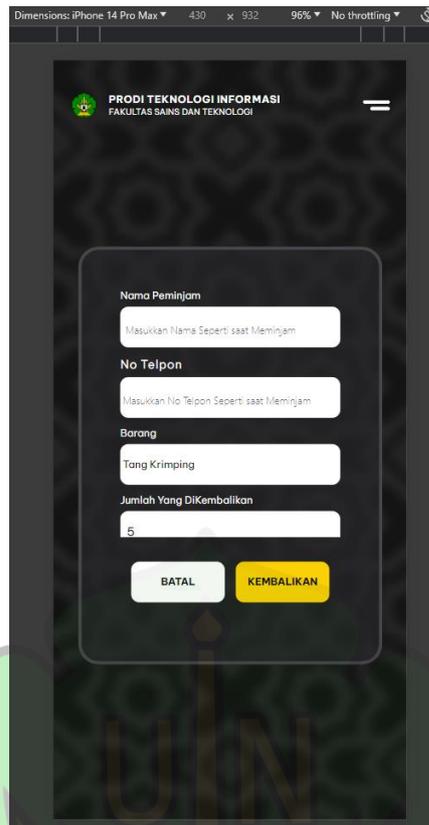
The screenshot shows a web browser window with the URL 'localhost:57212'. The page header includes the logo of 'PRODI TEKNOLOGI INFORMASI FAKULTAS SAINS DAN TEKNOLOGI' and navigation links for 'Home', 'Peminjaman', 'Pengembalian', and 'Admin'. The main content area features a form for returning items. The form has the following fields and labels:

- Nama Peminjam**: Input field with placeholder text 'Masukkan Nama Seperti saat Meminjam'.
- No. Telpin**: Input field with placeholder text 'Masukkan No. Telpin Seperti saat Meminjam'.
- Barang**: Input field with placeholder text 'Tang Krimping'.
- Jumlah Yang Dikembalikan**: Input field.

At the bottom of the form, there are two buttons: 'BATAL' (white) and 'KEMBALIKAN' (yellow).

Gambar 4. 59 Hasil Pengembalian *Screen Web* Ketika Telah Memilih Barang Tidak Berkode

Sumber: Data diolah (2024)

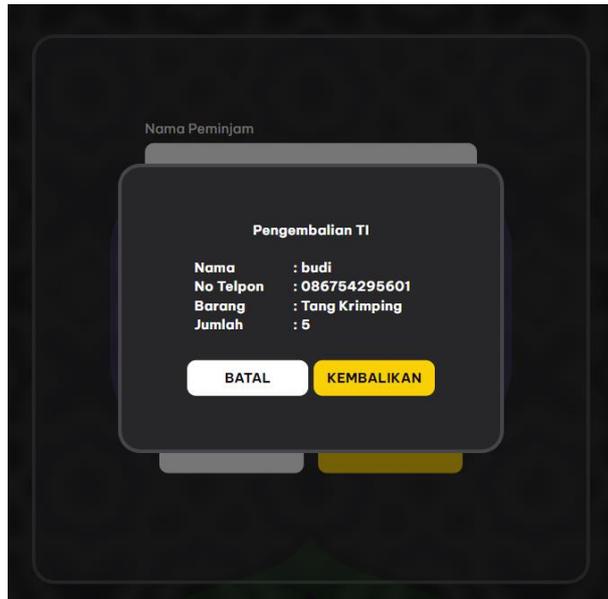


Gambar 4. 60 Hasil Pengembalian *Screen Mobile* Ketika Telah Memilih Barang Tidak Berkode

Sumber: Data diolah (2024)

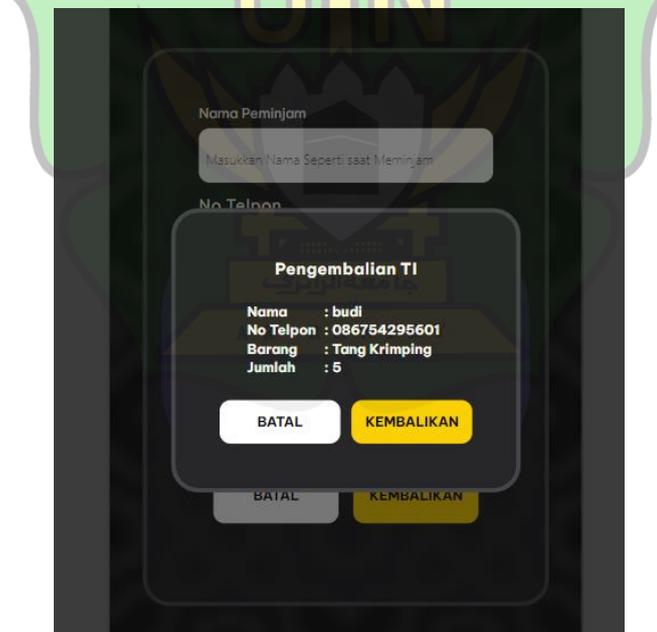
7. Konfirmasi Barang Yang Ingin Dikembalikan

Tampilan pada Gambar 4.61 dan 4.62 adalah halaman ke lima ketika *user* mahasiswa mengkonfirmasi pengembalian barang yang telah dipilih. Pada halaman ini terdapat beberapa informasi seperti nama, no telpn, barang yang dipinjam dan jumlah atau kode dari barang yang dikembalikan. Ketika *user* mahasiswa telah yakin ingin mengembalikan barang tersebut, maka *user* mahasiswa hanya perlu mengklik *button* kembalikan yang telah disediakan. Jika tidak yakin maka hanya perlu mengklik *button* batal.



Gambar 4. 61 Hasil Konfirmasi Pengembalian *Screen Web*

Sumber: Data diolah (2024)



Gambar 4. 62 Hasil Konfirmasi Pengembalian *Screen Mobile*

Sumber: Data diolah (2024)

Adapun hasil dari *Backend* aplikasi *Web* untuk fitur pengembalian terdapat dua diantaranya adalah kembalikan barang berkode, dan kembalikan barang tidak berkode. Penjelasan lebih lanjut dikemas pada poin-poin berikut.

1. kembalikan Barang Berkode

Pada gambar 4.63 merupakan penggalan kode untuk menangani fitur kembalikan barang berkode. Ketika kode ini berjalan, hal yang pertama adalah mengurangi data didalam *field* “JumlahTerpakai” dalam kategori barang di *database* sebesar 1. Lalu, dilakukan penghapusan *user* pada *database* “*User*” jika jumlah barang yang tersisa di *collection* “*BarangBarang*” dalam *database* “*user*” tidak lebih dari 1. Kemudian dilanjutkan dengan merubah *field* “*Status*” menjadi *false* pada *database* barang di kategori barang yang dikembalikan.

```
//proses mengatur jumlah terpakai agar berkurang untyuk barang berkode
int nilaiJumlahTerpakaiSekarang = 0;
await FirebaseFirestore.instance.collection("Barang").doc(docBarangYangDipinjamSekarang).get()
.then((value) => nilaiJumlahTerpakaiSekarang = value["JumlahTerpakai"]);
await FirebaseFirestore.instance.collection("Barang").doc(docBarangYangDipinjamSekarang).update({
  "JumlahTerpakai": nilaiJumlahTerpakaiSekarang - 1
});
print("Mengurangi barang terpakai berhasil");
//penghapusan data peminjam ketika barang sudah tidak tersisa
if (BanyakBarangDipinjamUser == 1) {
  await FirebaseFirestore.instance.collection("User").doc(DocPeminjamSekarang).delete();
  print("hapus data user berhasil");
}
//mengubah data barang berkode menjadi False( non aktif)
if (isBarangBerkodeYangAkanDikembalikan) {
  await FirebaseFirestore.instance.collection("Barang").doc(docBarangYangDipinjamSekarang).collection("BarangBarang").
  doc(KodeAtauJumlahYangDikembalikan[0].toString()).update({
    "Status": false
  });
  print("Mengubah status barang berhasil");
}
```

Gambar 4. 63 *Backend* Penggalan Kode Fitur Kembalikan Barang Berkode

Sumber: Data diolah (2024)

2. Kembalikan Barang Tidak Berkode

Pada gambar 4.64 merupakan penggalan kode untuk menangani fitur kembalikan barang tidak berkode. Ketika kode ini berjalan, tahap pertama dilakukan adalah mengecek apakah barang yang dipinjam oleh *user* lebih besar atau sama dengan barang yang akan dikembalikan oleh *user*. Jika iya, maka akan dilakukan pengurangan pada *field* “*JumlahTerpakai*” di *database* sebesar barang yang dikembalikan oleh *user* mahasiswa. Kemudian, dilakukan pengurangan

jumlah yang di pinjam oleh *user* di *database* “*User*”. Lalu, dilakukan penghapusan data *user* di *database* “*User*” ketika barang yang dipinjam telah 0. Jika tidak, maka pengembalian tidak akan dilakukan dan akan muncul pesan “Pengembalian Gagal Dek” dan beberapa pesan lainnya.

```
//proses mengatur jumlah terpakai agar berkurang untuyk barang tidak berkode
int nilaiJumlahTerpakaiSekarang = 0;
int jumlahPinjamanUser = 0;
await FirebaseFirestore.instance.collection("Barang").doc(docBarangYangDipinjamSekarang).get().
then((value) => nilaiJumlahTerpakaiSekarang = value["JumlahTerpakai"]);
await FirebaseFirestore.instance.collection("User").doc(DocPeminjamSekarang).collection("BarangBarang").doc(
idBarangYangDikembalikan).get().then((value) => jumlahPinjamanUser = int.parse(value["JumlahatauKodeBarang"].toString()));
print(jumlahPinjamanUser.toString() + " - " + kodeAtauJumlahBarangYangDikembalikan.toString() + "");
print("fffff : " + idBarangYangDikembalikan);
if (jumlahPinjamanUser >= int.parse(kodeAtauJumlahBarangYangDikembalikan.toString())) {
//mengurangi jumlah terpakai di database BARANG
await FirebaseFirestore.instance.collection("Barang").doc(docBarangYangDipinjamSekarang).update({
"JumlahTerpakai": nilaiJumlahTerpakaiSekarang - int.parse(kodeAtauJumlahBarangYangDikembalikan.toString())
});
//mengurangi jumlahhataukodebarang di database USER
await FirebaseFirestore.instance.collection("User").doc(DocPeminjamSekarang).collection("BarangBarang").doc(
idBarangYangDikembalikan).update({
"JumlahatauKodeBarang": jumlahPinjamanUser - int.parse(kodeAtauJumlahBarangYangDikembalikan.toString())
});
peminjamanGagalKarenaJumlahYangDiKembalikanKebanyakan = true; // KODE TIDAK EFISIEN
print("Mengurangi barang terpakai berhasil");
//menghapus field barang di BarangBarang ketika kodeataujumlahpinjaman == 0
await FirebaseFirestore.instance.collection("User").doc(DocPeminjamSekarang).collection("BarangBarang").doc(
idBarangYangDikembalikan).get().then((value) => jumlahPinjamanUser = int.parse(value["JumlahatauKodeBarang"].toString()));
if (jumlahPinjamanUser == 0) {
peminjamanGagalKarenaJumlahYangDiKembalikanKebanyakan = false;
}
} else {
peminjamanGagalKarenaJumlahYangDiKembalikanKebanyakan = true;
munculPopUpPengembalianKebanyakan = true;
print("Pengembalian Gagal Dek");
print("doc peminjam : " + DocPeminjamSekarang);
print("id barang peminjam : " + idBarangYangDikembalikan);
}
}
```

Gambar 4. 64 *Backend* Penggalan Kode Fitur Kembalikan Barang Tidak Berkode

Sumber: Data diolah (2024)

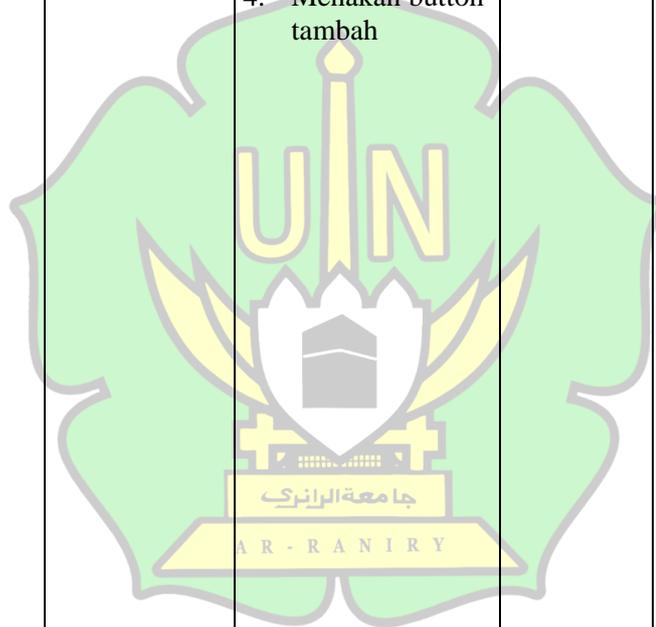
4.4 System Testing

Setelah menyelesaikan pengerjaan kode program pada tahap implementasi, langkah berikutnya adalah melakukan pengujian sistem. Setiap skenario pengujian mencakup pengujian dengan berbagai kasus uji (*test case*) yang tercantum dalam Tabel 4.6 Pengujian sistem dilakukan sesuai dengan yang telah dijelaskan di bab sebelumnya, yaitu dengan melakukan pengujian menyeluruh dari ujung ke ujung (*end to end*) terhadap sistem yang sudah terintegrasi dan diuji oleh operator prodi Teknologi Infomasi. Berikut merupakan tabel hasil dari pengujian *use case* aplikasi *Web* peminjaman inventaris laboratorium.

Tabel 4. 6 Hasil *Tast Case* Pengujian *Black Box*

No	Feature	User story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
1.	Authenticat ion	Sign In	Melakukan Sign In pada aplikasi web	<ol style="list-style-type: none"> 1. Mengklik button admin pada menu 2. Masuk ke halaman Sign In 3. Masukkan email 4. Masukkan password mengklik button login 	email = admin@gmail.com password = admin123	User berhasil masuk ke halaman admin	Berhasil
		Logout	Melakukan logout pada aplikasi web	User admin menekan menu logout.		User keluar dari halaman admin dan mengganti status Authentication user dari yang sebelumnya login menjadi mahasiswa (tidak login).	Berhasil

No	Feature	User story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
2.	Add	Tambah Kategori Barang	Melakukan penambahan kategori barang pada aplikasi web	<ol style="list-style-type: none"> 1. Mengklik menu tambah barang. 2. User admin menekan button tambah kategori 3. User admin mengisi nama barang, icon barang dan memilih tipe barang(berkode/ tidak berkode, sekali pakai) 4. Menakan button tambah 		<ol style="list-style-type: none"> 1. User admin menekan button tambah kategori 2. Muncul pop up untuk memasikkan data yang di perlukan 3. Mengirim data yang telah di inputkan sebelumnya 4. ke database setelah menekan button tambah 5. Menampilkan data kategori barang baru ke list kategori barang pada halaman menu tambah barang. 	Berhasil



No	Feature	User story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
		Tambah Barang	Melakukan penambahan barang pada aplikasi web	<ol style="list-style-type: none"> 1. Mengklik menu tambah barang. 2. User admin memilih kategori barang yang ingin ditambahkan barangnya. 3. User admin menambahkan kode barang jika kategori barang yang dipilih adalah barang berkode, user admin menambah jumlah total barang jika kategori barang yang dipilih barang tidak berkode/sekali pakai. 4. Mengklik button tambah. 		<ol style="list-style-type: none"> 1. User admin memilih kategori barang yang ingin ditambahkan barangnya. 2. Muncul pop up untuk menambahkan barang. 3. User menginputkan kode atau jumlah barang, lalu menekan button tambah. 4. Jumlah barang bertambah. 	Berhasil
3.	Delete	Hapus Kategori Barang	Melakukan penghapusan kategori barang pada aplikasi web.	<ol style="list-style-type: none"> 1. Mengklik menu tambah barang. 2. User admin memilih kategori barang yang ingin dihapus. 		<ol style="list-style-type: none"> 1. User admin memilih kategori barang yang ingin dihapus. 2. Muncul pop up, lalu user admin 	

No	Feature	User story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/ Fail
				<ol style="list-style-type: none"> 3. User admin menekan icon hapus. 4. Menekan button hapus. 		<ol style="list-style-type: none"> 3. menekan icon hapus. 4. Muncul pop up konfirmasi penghapusan kategori barang. Lalu user admin menekan button hapus. 5. Menampilkan halaman menu tambah barang dengan data yang telah diperbaharui. 	
		Hapus Barang	Melakukan penghapusan barang pada aplikasi web.	<ol style="list-style-type: none"> 1. Mengklik menu barang. 2. User admin memilih kategori barang yang ingin hapus barangnya. 3. Jika user admin memilih kategori barang berkode, maka user admin akan memilih barang yang ingin dihapus. Namun jika user admin memilih kategori barang tidak berkode/sekali pakai, maka user admin menginputkan jumlah dari barang yang ingin dihapus. 4. Mengklik button hapus. 		<ol style="list-style-type: none"> 1. User admin memilih kategori barang yang ingin hapus barangnya. 2. Muncul pop up Jika kategori barang yang dipilih barang berkode, maka user admin memilih barang yang ingin dihapus. Namun jika kategori barang yang dipilih barang tidak berkode/sekali pakai, maka user admin menginputkan jumlah barang yang ingin dihapus. 	Berhasil

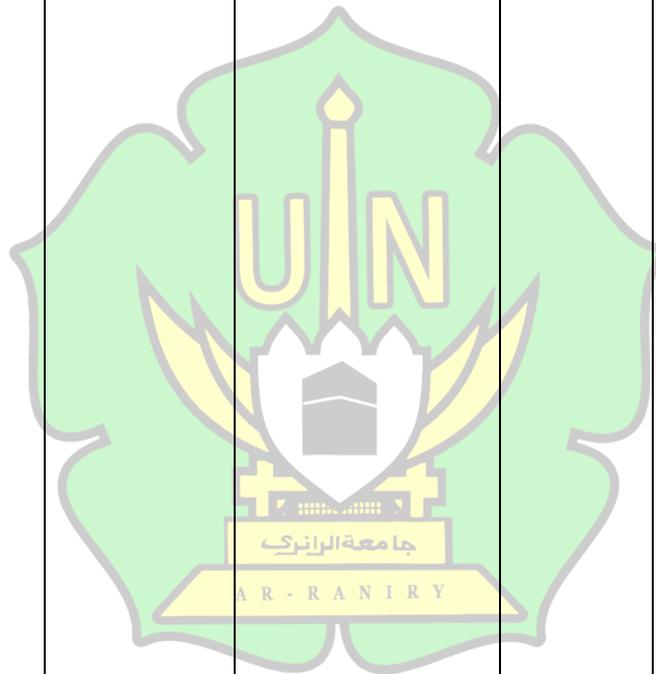
No	Feature	User story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
						3. Mengklik button hapus. 4. Kembali ke halaman menu barang, yang datanya telah diperbaharui.	
4.	Edit	Edit kategori Barang	Melakukan pengeditan data pada kategori barang pada aplikasi web.	1. Mengklik menu tambah barang. 2. User admin memilih kategori barang yang ingin diedit barangnya. 3. User admin menekan icon edit. 4. Memperbaharui data yang ingin diganti. 5. User admin menekan button ubah.		1. User admin memilih kategori barang yang ingin diedit barangnya. 2. Muncul pop up, lalu user admin menekan icon edit. 3. Muncul pop up, lalu user admin memperbaharui data yang ingin diganti. 4. Menekan button ubah. 5. Menampilkan halaman menu tambah barang yang telah diperbaharui datanya.	Berhasil
5.	Monitoring	Melihat riwayat peminjaman dan pengembalian barang.	Menampilkan data peminjaman dan pengembalian barang.	Mengklik menu riwayat peminjaman.		1. Menampilkan halaman riwayat peminjaman dan pengembalian, setelah mengklik menu riwayat peminjaman 2. Menampilkan list data peminjaman dan	Berhasil

No	Feature	User story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
						pengembalian user mahasiswa.	
6.	Download	Mendownload database report	Melakukan Download data peminjaman dan pengembalian	<ol style="list-style-type: none"> 5. Mengklik menu riwayat peminjaman. 6. Mengklik button download data. 7. Melihat data pada google sheet. 8. Mendownload google sheet. 		<ol style="list-style-type: none"> 3. Setelah mengklik button download data maka akan ditambah tab baru yang mengarah ke google sheets. 4. Google sheets mengambil data dari database melalui apps script pada extantion 	Berhasil
7.	Peminjaman	Peminjaman	Melakukan peminjaman barang pada aplikasi web.	<ol style="list-style-type: none"> 1. Mengklik menu peminjaman. 2. Mengisi data yang di perlukan berupa nama dan no telpon. 3. Menekan button "tekan untuk pilih" untuk memilih barang. 4. Memilih kategori barang yang ingin dipinjam. 5. Jika kategori barang berkode yang dipilih, maka user mahasiswa memilih barang untuk dipinjam. Namun jika kategori barang yang di pilih barang tidak berkode/sekali 		<ol style="list-style-type: none"> 1. Mengisi data yang di perlukan berupa nama dan no telpon. 2. Menekan button "tekan untuk pilih" untuk memilih barang. 3. Muncul pop up memilih kategori barang. 4. Menekan kategori barang yang ingin dipinjam. 5. Jika kategori barang yang dipilih barang berkode, maka user mahasiswa memilih barang yang 	Berhasil

No	Feature	User story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/ Fail
				<p>pakai, maka user mahasiswa menginputkan jumlah barang yang ingin dipinjam.</p> <ol style="list-style-type: none"> 6. Menekan button pinjam. 7. kembali ke halaman input data peminjaman. 8. mengklik button pinjam. 9. Menekan pinjam pada pop up konfirmasi peminjaman. 		<p>ingin dipinjam. Namun jika kategori barang yang dipilih barang tidak berkode/sekali pakai, maka user mahasiswa menginputkan jumlah barang yang ingin dipinjam.</p> <ol style="list-style-type: none"> 6. Mengklik button pinjam. 7. Kembali ke halaman input data peminjaman, lalu mengklik button pinjam. 8. Muncul pop up konfirmasi peminjaman yang berisi nama, no telpon dan nama barang beserta kode atau jumlah barang sesuai inputan user mahasiswa sebelumnya. 9. Menekan button pinjam. 10. Menampilkan halaman input data peminjaman. 	

No	Feature	User story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/Fail
8.	pengembalian	pengembalian	Melakukan pengembalian barang pada aplikasi web.	<ol style="list-style-type: none"> 1. Menekan menu pengembalian. 2. Menekan button "tekan untuk pilih" untuk memilih barang. 3. Memilih kategori barang yang ingin dikembalikan 4. Jika kategori barang berkode yang dipilih, maka user mahasiswa memilih barang untuk dikembalikan. Namun jika kategori barang yang di pilih barang tidak berkode, maka user mahasiswa menginputkan jumlah barang yang ingin dikembalikan. 5. Mengklik button kembalikan pada pop up pemilihan barang. 6. Jika yang dipilih barang tidak berkode, maka user mahasiswa perlu mengisi nama dan no telpon lalu mengklik button kembalikan. Namun jika barang yang dipilih barang 		<ol style="list-style-type: none"> 1. Menekan button "tekan untuk pilih" untuk memilih barang. 2. Muncul pop up memilih kategori barang. 3. Menekan kategori barang yang ingin dikembalikan 4. Jika kategori barang yang dipilih barang berkode, maka user mahasiswa memilih barang yang ingin dikembalikan. Namun jika kategori barang yang dipilih barang tidak berkode, maka user mahasiswa menginputkan jumlah barang yang ingin dikembalikan. 5. Mengklik button kembalikan. 6. Menampilkan halaman pengembalian dengan data yang sesuai pilihan user mahasiswa sebelumnya. 	Berhasil

No	Feature	User story	Test Case	Test Steps	Test Data	Expected Result	Result Pass/ Fail
				<p>berkode, maka user mahasiswa akan langsung mengklik button kembalikan</p> <p>7. Lalu Menakan kembalikan pada pop up konfirmasi pengembalian.</p>		<p>7. Jika barang yang dikembalikan barang berkode, maka yang ditampilkan adalah nama barang dan kode barang. Namun jika barang yang dikembalikan barang tidak berkode, maka yang ditampilkan adalah inputan nama, inputan no telpon, nama barang dan jumlah yang dikembalikan.</p> <p>8. Mengklik button kembalikan.</p> <p>9. Muncul pop up konfirmasi pengembalian yang berisi nama, no telpon, nama barang dan kode atau jumlah barang sesuai inputan user mahasiswa sebelumnya.</p> <p>10. Menekan button pinjam.</p> <p>11. Menampilkan halaman pengembalian.</p>	



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari penjelesan pada bab sebelumnya, maka dapat disimpulkan beberapa hal berikut :

1. Aplikasi *web* peminjaman inventaris laboratorium dapat dikembangkan dengan menggunakan metode *Personal Extreme Programming* (XP) dan menghasilkan beberapa fitur yaitu *authentication, add, delete, edit, monitoring, download*, peminjaman dan pengembalian.
2. Fitur-fitur yang dibangun dalam aplikasi *web* peminjaman inventaris laboratorium dapat berjalan dengan baik dari hasil pengujian *black box* yang diuji oleh operator Prodi Teknologi Informasi.

5.2 Saran

Setelah penelitian ini, peneliti memiliki beberapa saran yang dapat diterapkan pada penelitian berikutnya. Adapun saran tersebut adalah sebagai berikut :

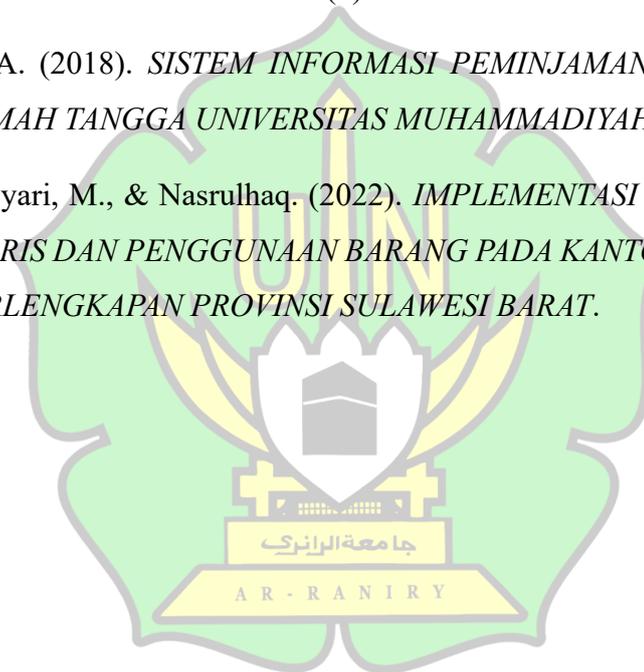
1. Pengembangan aplikasi *web* peminjaman inventaris laboratoriu pada penelitian ini menggunakan metode *Personal Extreme Programming*, pada penelitian berikutnya dapat dilakukan dengan menerapkan metode pengembangan perangkat lunak lainnya.
2. Pengembangan aplikasi *web* peminjaman inventaris laboratoriu pada penelitian ini menggunakan *framework flutter*, pada penelitian berikutnya dapat dilakukan dengan menggunakan *framework* lainnya.
3. Melakukan pengembangan dan memperkaya fitur aplikasi *web* peminjaman inventaris laboratoriu, seperti membuat fitur riwayat penambahan barang.

DAFTAR PUSTAKA

- Al, N., Dhanardana, D., & Rahayu, T. (2020). *APLIKASI PEMINJAMAN RUANGAN BERBASIS WEB PADA DIREKTORAT JENDERAL HAK ASASI MANUSIA*.
- Aminah, S., Bagas Bhaskoro, S., Sunarya, A. S., Teknik, J., Manufaktur, O., Mekatronika, D., & Bandung, P. (2019). *Desain dan Implementasi Aplikasi Inventaris Alat Praktikum Pada Laboratorium Berbasis Android dan QR Code*. In *Gd. Cimahi Technopark*.
- Angga Iswara, R. (2019). *SISTEM INFORMASI LAYANAN PEMBUATAN SURAT PENGANTAR KARTU KELUARGA DAN KARTU TANDA PENDUDUK DI KANTOR DESA BANYUADEM*.
- Costa Pinaria, G., Deo Rindengan, Y., Najoan, N., & Xaverius, B. (n.d.). *Web Based E-Commerce Application Buying and Selling Food Ingredients for Manado City*. <https://ejournal.unsrat.ac.id/index.php/informatika>
- Evanansy, A. (2022). *IMPLEMENTASI PERSONAL EXTREME PROGRAMMING DALAM PENGEMBANGAN INTEGRASI SISTEM INFORMASI AKADEMIK DAN E-LEARNING*.
- Juliasnyah, H. (2020). *PENGGUNAAN FRAMEWORK FLUTTER UNTUK MEMBANGUN APLIKASI AL-QUR'AN BERBASIS ANDROID*.
- karmila. (2019). *PENGEMBANGAN SISTEM INFORMASI PELAYANAN DINAS SOSIAL PADA PROVINSI SULAWESI BARAT BERBASIS WEB*.
- Maloky, E. D. (2022). *RANCANG BANGUN SISTEM PEMINJAMAN PERALATAN DI LABORATORIUM TELEKOMUNIKASI BERBASIS IOT MENGGUNAKAN APLIKASI ANDROID*.
- Muhammad, B. (2018). *PENGEMBANGAN APLIKASI WEB KARYAKU SEBAGAI DIREKTORI KARYA MAHASISWA UNIVERSITAS NEGERI YOGYAKARTA*.

- Nuzul Firdaus, S. (2023). *PENGEMBANGAN APLIKASI KAMUS BAHASA ACEH BERBASIS MOBILE*.
- Pangestu, E. I. (2022). *PERANCANGAN WEBSITE PEMERINTAH DESA SEBAGAI MEDIA PENYEBARAN INFORMASI BAGI MASYARAKAT BERBASIS WEB*.
- Prasetyo, R. E. (2022). *ANALISIS DAN PERANCANGAN USER INTERFACE DAN USER EXPERIENCE PADA STARTUP PICNICKER DENGAN PENDEKATAN USER PERSONA BERBASIS DESIGN THINKING*.
- Putra, W. P., Saputra, H. K., Hadi, A., & Mursyida, L. (2024). Penerapan Teknologi Kompresi Data untuk Optimalisasi Manajemen Bahan Ajar Digital Pada Aplikasi Respository Menggunakan Personal Extreme Programming (PXP). *J-HyTEL: Journal of Hypermedia & Technology-Enhanced Learning*, 2(1), 15–29. <https://doi.org/10.58536/j-hytel.v2i1.108>
- Rangga Bakti, I., Permata Bunda, Y., Supriyanto, A., Riki Mustafa, S., & Maradona, H. (2023). Perancangan Sistem Informasi Inventaris Barang Laboratorium Pada SMKN 1 Ujungbatu. *Jurnal Minfo Polgan*, 12(1), 1163–1172. <https://doi.org/10.33395/jmp.v12i1.12621>
- Rezandy, A. K. (2018). *RANCANG BANGUN APLIKASI PEMINJAMAN ALAT BERBASIS WEB PADA PROGRAM STUDI PSPT DI SMKN 1 SURABAYA*.
- Rinaldy, O. (2019). *PERANCANGAN DAN IMPLEMENTASI LAYANAN INFORMASI DAN PEMASARAN DRONILA MENGGUNAKAN METODE PERSONAL EXTREME PROGRAMMING*.
- Ronal, Yunita, & yuliana. (2022). Desain Unified Modeling Language (UML) Dalam Perancangan Aplikasi Hauling Trip Di Industri Tambang Batubara. *Jurnal Teknik Informatika Dan Sistem Informasi*, 9(4). <http://jurnal.mdp.ac.id>
- Sonita, A., & Fardianitama, R. F. (2018). *APLIKASI E-ORDER MENGGUNAKAN FIREBASE DAN ALGORITME KNUTH MORRIS PRATT BERBASIS ANDROID*.

- Vallendito, B. (2020). *PEMODELAN USER INTERFACE DAN USER EXPERIENCE MENGGUNAKAN DESIGN THINKING*.
- Wibowo, M. I., Hendra Brata, A., & Brata, K. C. (2019). *Pengembangan Aplikasi Perangkat Bergerak Pengobatan Tuberculosis Berbasis Android Menggunakan Personal Extreme Programming (Studi Kasus : Puskesmas Polowijen)* (Vol. 3, Issue 10). <http://j-ptiik.ub.ac.id>
- Widarma, A., & Rahayu, S. (2018). *PERANCANGAN APLIKASI GAJI KARYAWAN PADA PT. PP LONDON SUMATRA INDONESIA Tbk. GUNUNG MALAYU ESTATE-KABUPATEN ASAHAN. 1(2)*.
- Widhiatmojo, A. (2018). *SISTEM INFORMASI PEMINJAMAN DI BIRO ASET DAN RUMAH TANGGA UNIVERSITAS MUHAMMADIYAH MAGELANG*.
- Wirna, Isa Ansyari, M., & Nasrulhaq. (2022). *IMPLEMENTASI PENGELOLAAN INVENTARIS DAN PENGGUNAAN BARANG PADA KANTOR BIRO UMUM DAN PERLENGKAPAN PROVINSI SULAWESI BARAT*.



LAMPIRAN

Lampiran 1 *Back End*

```
class DatabaseServie {
    final CollectionReference barangCollection =
        FirebaseFirestore.instance.collection("Barang");
    final CollectionReference userCollection =
        FirebaseFirestore.instance.collection("User");
    final CollectionReference ReportCollection =
        FirebaseFirestore.instance.collection("Report");
    final CollectionReference ReportPengembalianCollection =
        FirebaseFirestore.instance.collection("ReportPengembalian");
    //getting user data
    Future gettingUserData(String email) async {
        QuerySnapshot snapshot =await barangCollection.where("email", isEqualTo:
        email).get();
        return snapshot;}-----
        ----- else {
            //untuk barang tidak berkode
            PercobaanPengembalian++;
            await ReportCollection.doc(idYangDipilih.toString()).update({
                "PercobaanPengembalian": PercobaanPengembalian,
                "jumlahPengembalian": FieldValue.arrayUnion([
                    jumlahYangDiKembalikan.toString() +
                    " " + PercobaanPengembalian.toString() ]),
                "jamPengembalian": FieldValue.arrayUnion([ jamSekarang +":" +
                    menitSekarang + " " + PercobaanPengembalian.toString()]),
                "tanggalPengembalian": FieldValue.arrayUnion(
                    [date.toString() + " " + PercobaanPengembalian.toString()]),});}
```

Lampiran 2 *Front End Admin File Barang*

```
return FutureBuilder(  
  future: getBarang(),  
  builder: (context, snapshot) {  
    print(KumpulanNama.length);  
    if (snapshot.connectionState == ConnectionState.done) {  
      return SingleChildScrollView(  
        child: Column(  
          mainAxisAlignment: MainAxisAlignment.center,  
          children: [  
            SizedBox(height: 35,),  
            Container(  
              decoration: BoxDecoration(  
                borderRadius: BorderRadius.circular(25),  
                color: Color.fromARGB(163, 39, 39, 41),  
border: Border.all(width: 4, color: Color.fromARGB(255, 71, 71, 75))),  
              height: 735, width: 1450,  
-----  
-----else {  
      return Column(  
        mainAxisAlignment: MainAxisAlignment.center,  
        children: [  
          Container(  
            width: 50,  
            height: 50,  
            child: CircularProgressIndicator(  
              color: Colors.white,  
            ),],); } } } } }
```

Lampiran 3 *Front End Admin File Riwayat*

```
Widget build(BuildContext context) {
  return FutureBuilder(
    future: mengambilDataReportPinjaman(),
    builder: (context, snapshot) {
      return Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          ///SizedBox(height: MediaQuery.of(context).size.height*0.1 ),
          Center(
            child: Container(
              decoration: BoxDecoration(
                borderRadius: BorderRadius.circular(25),
                color: Color.fromARGB(163, 39, 39, 41),-----
-----BoxDecoration(
                borderRadius: BorderRadius.circular(10),
                color: Color.fromARGB(255, 71, 71, 75)), width: 215, height: 61,
              child: Padding(
                padding: const EdgeInsets.symmetric(horizontal: 13),
                child: Row(
                  mainAxisAlignment: MainAxisAlignment.spaceBetween,
                  children: [Image.asset(
                    "assets/images/xls.png",height: 44, width: 44,),
                    Text("Download Data",
                      style: GoogleFonts.beVietnamPro(fontWeight: FontWeight.bold,
                        fontSize: 15, color: Colors.white,
                      ),),),),),),),), ],); }}

```

Lampiran 4 *Front End Admin File* tambah barang

```
Widget build(BuildContext context) {
  //mengambil data dari text fil
  TextEditingController Namabarang = new TextEditingController();
  TextEditingController DataBarang = new TextEditingController();
  return FutureBuilder(
    future: getBarang(),
    builder: (context, snapshot) {
      if (snapshot.connectionState == ConnectionState.done) {
        return Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            //SizedBox(height: MediaQuery.of(context).size.height*0.1 ),
            Center(
              child: Container(
                decoration: BoxDecoration(
                  borderRadius: BorderRadius.circular(25),
                ),
              ),
            ),
          ],
        );
      } else {
        return Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Container(
              width: 50,
              height: 50,
              child: CircularProgressIndicator(
                color: Colors.white,
              ),
            ),
          ],
        );
      }
    },
  );
}
```

Lampiran 5 *Front End* Mahasiswa File Barang

```
Widget build(BuildContext context) {  
  return FutureBuilder(  
    future: getBarang(),  
    builder: (context, snapshot) {  
      print(KumpulanNama.length);  
      if (snapshot.connectionState == ConnectionState.done) {  
        return SingleChildScrollView(  
          child: Column(  
            mainAxisAlignment: MainAxisAlignment.center,  
            children: [  
              SizedBox(height: 35,),  
              Container(  
                decoration: BoxDecoration(  
                  borderRadius: BorderRadius.circular(25),  
                  color: Color.fromARGB(163, 39, 39, 41),  
                  border: Border.all(  
                    width: 4, color: Color.fromARGB(255, 71, 71, 75))),  
                height: 735,  
                width: 1450,  
                -----  
                ----- else {  
                  return Column(  
                    mainAxisAlignment: MainAxisAlignment.center,  
                    children: [  
                      Container(width: 50, height: 50,  
                        child: CircularProgressIndicator(  
                          color: Colors.white,  
                          ),,); }));}}
```

Lampiran 6 *Front End Mahasiswa File Home*

```
Widget build(BuildContext context) {
  return SingleChildScrollView(
    child: Column(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        SizedBox(
          height: 250,),
        Text(
          "SELAMAT DATANG DI",
          style: GoogleFonts.beVietnamPro(
            fontSize: 70, color: Colors.white, fontWeight: FontWeight.bold)),
        Row(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text(
              "PEMINJAMAN TI", style: GoogleFonts.beVietnamPro(fontSize: 70,
                color: Colors.white, fontWeight: FontWeight.bold),
            ),
            -----
            -----
            -----
            -----
            -----
            -----
            Text("Bawah",
              style: GoogleFonts.beVietnamPro(
                fontSize: 15,
                color: Colors.white,
                fontWeight: FontWeight.w200))),)],)],),
        SizedBox(
          height: 430,),
        Barang_User(],,);}}}
```

Lampiran 7 *Front End* Mahasiswa File Peminjaman

```
Widget build(BuildContext context) {
  TextEditingController JumlahyangDipinjamController =
    new TextEditingController();
  return Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      //SizedBox(height: MediaQuery.of(context).size.height*0.1 ),
      Center(
        child: loading
      ),
      ? Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Container(width: 50, height: 50,
          -----
          -----
          -----
          -----
          -----
          -----
          Container(
            decoration: BoxDecoration(
              borderRadius: BorderRadius.circular(10),
              color: Color.fromARGB(255, 250, 208, 7),
            ), height: 62, width: 202,
            child: Center(
              child: Text(
                "PINJAM",
                style: GoogleFonts.beVietnamPro(
                  fontSize: 15,
                  fontWeight: FontWeight.bold,
                  color: Color.fromARGB(255, 7, 7, 10)),
              ),),),),),),),),),),),); } }
```

Lampiran 8 *Front End* Mahasiswa File Pengembalian

```
Widget build(BuildContext context) {
  TextEditingController JumlahYangDikembalikanController =
    new TextEditingController();
  TextEditingController NamaPeminjamController = new TextEditingController();
  TextEditingController NoTelponController = new TextEditingController();
  return Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      //SizedBox(height: MediaQuery.of(context).size.height*0.1 ),
      Center(
        child: munculPopUpPengembalianKebanyakan
          ? //pop up pengembalian berlebihan
            Container(
              decoration: BoxDecoration(borderRadius: BorderRadius.circular(25),
                color: Color.fromARGB(255, 39, 39, 41), border: Border.all(
                  width: 4, color: Color.fromARGB(255, 71, 71, 75))),
            )
          : Container(
              decoration: BoxDecoration(
                borderRadius:
                  BorderRadius.circular(10), color: Colors.white,),
                height: 63, width: 378,
                child: Center(
                  child: Text("Tekan Untuk Pilih",style: GoogleFontsbeVietnamPro(
                    fontSize: 15, color: Colors.grey, fontWeight:
                      FontWeight.w300,)),),), ],),), ),
                SizedBox(height: 15,], ),),),),]); } }
```

Lampiran 9 *Front End Mobile Mahasiswa File Home*

```
Widget build(BuildContext context) {  
  return SingleChildScrollView(  
    child: Column(  
      mainAxisAlignment: MainAxisAlignment.start,  
      crossAxisAlignment: CrossAxisAlignment.start,  
      children: [  
        Text(  
          "Selamat Datang Di",  
          style: GoogleFonts.beVietnamPro(  
            fontSize: 20, color: Colors.white, fontWeight: FontWeight.bold),  
          ),  
        Text(  
          "Peminjaman TI",  
          style: GoogleFonts.beVietnamPro(  
            fontSize: 20, color: Colors.white, fontWeight: FontWeight.bold),  
          ),  
        //jarak anantara titel dan komponen  
        SizedBox(  
          height: 25,  
        ),  
        Barang_User_Mobile()  
      ],  
    ),  
  );  
}
```

The image contains a large, semi-transparent watermark logo of UIN Ar-Raniry. The logo is circular with a green background and a yellow border. It features the letters 'UIN' in a large, bold, yellow font at the top. Below 'UIN' is a white building icon with a grey roof. At the bottom of the logo, there is a yellow banner with the Arabic text 'جامعة الرانيري' and the English text 'AR-RANIRY' below it.

Lampiran 11 *Front End Mobile* Mahasiswa File Pengembalian

```

Widget build(BuildContext context) {
  TextEditingController JumlahYangDikembalikanController =
    new TextEditingController();
  TextEditingController JumlahYangRusakController =
    new TextEditingController();
  TextEditingController NamaPeminjamController = new
  TextEditingController();
  TextEditingController NoTelponController = new TextEditingController();
  return Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      //SizedBox(height: MediaQuery.of(context).size.height*0.1 ),
      Center( child: munculPopUpPengembalianKebanyakan-----
----- child: Container(
  decoration: BoxDecoration(borderRadius:
BorderRadius.circular(10),
color: Colors.white,),
height: 50, width: 273,
child: Center(child: Text(
"Tekan Untuk Pilih", style: GoogleFonts
.beVietnamPro(
  fontSize: 12, color: Colors.grey, fontWeight: FontWeight.w300,),
),),),),),),),),),
      SizedBox(
        height: 15,),),),),),),),); }}

```