

## BAB 3

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Koordinasi

Selama menjalani program magang di PT PLN Nusantara Power Muara Karang, penempatan dilakukan di bawah divisi *Assistant Manager System Owner* dengan peran sebagai *Fullstack Developer*. Dalam pelaksanaan kegiatan magang, keterlibatan dilakukan bersama tim kecil yang terdiri dari tiga mahasiswa magang, yakni dua orang dengan posisi *Fullstack Developer* dan satu orang dengan posisi *UI/UX Designer* sekaligus bertanggung jawab terhadap pengelolaan *database*.

Aktivitas sehari-hari dilaksanakan di bawah bimbingan langsung dari Bapak Qasim selaku mentor lapangan, serta mendapatkan arahan dan supervisi dari Bapak Bernandez Noverson Lupy selaku *Assistant Manager System Owner*.

Koordinasi dan komunikasi antar anggota tim dilakukan secara langsung di lingkungan kerja melalui interaksi tatap muka serta memanfaatkan media perpesanan seperti WhatsApp. Metode ini memungkinkan terjadinya komunikasi yang lebih intensif dan efisien, terutama dalam mendiskusikan aspek teknis dan penyelesaian masalah dengan cepat.

Untuk mendukung kolaborasi dalam pengembangan proyek, tim developer menggunakan platform GitHub sebagai media utama dalam pengelolaan kode sumber. Seluruh anggota tim developer melakukan unggah (*push*) kode, peninjauan (*review*), dan penggabungan (*merge*) terhadap perubahan yang dilakukan pada repositori proyek. Penggunaan GitHub ini memberikan transparansi dan struktur kerja yang jelas dalam setiap tahapan pengembangan sistem.

#### 3.2 Tugas yang Dilakukan

Selama pelaksanaan program magang di PT PLN Nusantara Power Muara Karang berikut adalah tugas-tugas yang telah dilaksanakan :

- 1. Implementasi Sistem Login untuk Dua Role Berbeda (*Admin* dan *Super Admin*)**

Mengembangkan sistem autentikasi pengguna yang mendukung dua peran berbeda, yaitu *admin* dan *super admin*. Sistem ini dirancang dengan logika autentikasi yang berbeda untuk masing-masing role. Apabila pengguna

dengan role *admin* memasukkan kredensial yang tidak valid, maka akan ditampilkan notifikasi kesalahan berupa informasi yang dimasukkan tidak sesuai. Sementara itu, apabila pengguna dengan role *super admin* gagal dalam proses login, sistem akan menampilkan halaman tambahan untuk melakukan pergantian kata sandi. Proses pergantian ini mensyaratkan verifikasi melalui NID guna memastikan validitas identitas pengguna sebelum melakukan perubahan.

## 2. Pengembangan Sistem Peminjaman Aset

Sistem peminjaman aset dibangun untuk menampilkan daftar lengkap tipe aset serta aset-aset yang tersedia untuk dipinjam. Setiap aset ditampilkan beserta status terkini (tersedia atau sedang dipinjam), serta riwayat penggunaannya secara masing-masing. Sistem ini dilengkapi dengan fitur *shopping cart* yang memungkinkan pengguna memilih lebih dari satu aset sebelum melanjutkan proses peminjaman. Setelah aset dipilih, pengguna diarahkan ke halaman konfirmasi untuk memastikan aset yang ingin di pinjam sudah benar, lalu pengguna akan di arahkan ke pengisian formulir informasi peminjam, yang kemudian disimpan ke dalam database.

## 3. Pembuatan Halaman Riwayat Peminjaman Aset

Untuk mendukung akuntabilitas dan transparansi, dikembangkan halaman yang menyajikan riwayat lengkap peminjaman aset. Informasi yang ditampilkan mencakup data aset, tanggal peminjaman, identitas peminjam, dan alasan peminjaman tersebut dilakukan. Halaman ini memudahkan pihak terkait dalam melakukan pelacakan penggunaan aset sepanjang waktu.

## 4. Pengembangan Halaman Permintaan Pembelian Barang

Sebagai bagian dari proses pengadaan barang, dibuat halaman khusus yang memungkinkan pengguna untuk mengajukan permintaan pembelian barang melalui formulir digital. Data yang dimasukkan mencakup nama penerima, nama pengirim, identitas penerima seperti department dan NID nya, tanggal permintaan, serta alasan permintaan. Sistem ini menyediakan fitur untuk menghasilkan dokumen permintaan dalam format *PDF*, yang dapat diunduh dan digunakan sebagai dokumen resmi dalam proses pengajuan internal.

## 5. Pembuatan Halaman Pengembalian Aset

Halaman pengembalian aset dirancang untuk mencatat transaksi pengembalian atas barang-barang yang telah dipinjam sebelumnya. Dalam

proses ini, pengguna memilih aset yang ingin dikembalikan, kemudian data akan diverifikasi dan diperbarui secara otomatis agar status aset kembali menjadi tersedia.

### 6. Pengembangan Halaman Riwayat Pengembalian Aset

Sebagai pelengkap sistem pelacakan aset, dibuat halaman yang menyajikan riwayat pengembalian aset, meliputi informasi aset, waktu pengembalian, identitas pengembali, dan nama peminjam aset. Fitur ini berfungsi untuk mendokumentasikan seluruh aktivitas pengembalian dengan sistematis dan transparan.

### 3.3 Uraian Pelaksanaan Magang

Tabel 3.1 merangkum aktivitas kerja yang dilakukan secara bertahap setiap minggu selama masa magang.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Pengenalan Lingkungan Kerja Magang Dan Mengikuti Rangkaian Acara Bulan K3 Yang Wajib Di Ikuti Oleh Seluruh Karyawan Di PT PLN Nusantara Power Unit Pembangkitan Muara Karang .
2	Diskusi Dengan Supervisor dan Tim IT Tentang Masalah Yang Ada Di Lingkungan Kerja Magang .
3	Mempelajari Serta Memahami Ulang Konsep <i>Framework</i> Yang Akan Digunakan .
4	Diskusi dengan mentor dan team IT terkait struktur <i>website</i> dan melakukan <i>set up GitHub</i> untuk proyek yang akan di kerjakan .
5	Menerima design dari <i>UI/UX Designer</i> dan mulai Membuat Logika untuk <i>Popup Login, Forget Password</i> dan Autentikasi User. .
6	Membantu dan mengikuti acara presentasi produk <i>Realwear Navigator 520</i> yang diadakan dalam perusahaan, serta memperbaiki bug pada <i>pop up</i> modal bagian login. .

Minggu Ke -	Pekerjaan yang dilakukan
7	Membantu menyiapkan acara penutupan K3 Nasional yang diadakan oleh perusahaan, serta mengubah halaman login agar tidak menggunakan <i>pop up</i> modal, dan mengadakan diskusi dengan mentor dan tim IT terkait identitas yang akan dimiliki oleh user dan tim IT yang akan menggunakan <i>website</i> , serta mulai mengerjakan halaman aset untuk menampilkan aset.
8	Memperbaiki bug pada halaman aset serta menambahkan fitur <i>shopping cart</i> pada halaman aset yang akan digunakan untuk melakukan peminjaman barang, dan mengadakan diskusi dan presentasi kepada mentor.
9	Memperbaiki bug pada halaman aset di bagian <i>shopping cart</i> dimana aset yang dimasukkan ke dalam cart tidak terbawah ke halaman konfirmasi untuk di pinjam, serta membantu tim IT menyiapkan acara atau rapat yang diadakan di dalam perusahaan dan melakukan laporan mingguan kepada mentor tentang progress yang dikerjakan.
10	Membantu tim IT dalam menyiapkan acara atau rapat, diskusi dengan <i>designer UI/UX</i> tentang tampilan <i>shopping cart</i> serta membuat tampilannya sesuai dengan arahan yang diberikan oleh <i>designer UI/UX</i> , serta memberikan laporan mingguan kepada mentor.
11	Melakukan revisi tampilan <i>shopping cart</i> setelah berdiskusi dengan <i>designer UI/UX</i> dan memperbaiki bug pada halaman aset di bagian <i>shopping cart</i> .
12	Membantu tim IT dalam menyiapkan acara Halal Bihalal yang diadakan dalam perusahaan serta mengikuti rangkaian acara yang diadakan, membantu tim IT merapihkan <i>excel</i> untuk divisi <i>system owner</i> dan membuat halaman baru untuk melakukan input data diri peminjam barang..
13	Membantu tim IT menyiapkan acara atau rapat yang diadakan di dalam perusahaan. Membuat halam baru untuk menampilkan <i>history</i> dari aset yang pernah dipinjam..

Minggu Ke -	Pekerjaan yang dilakukan
14	Membantu tim IT merapikan data <i>Excel</i> divisi <i>System Owner</i> serta menyiapkan acara atau rapat yang diadakan di dalam perusahaan. Melakukan sinkronisasi <i>database</i> untuk table <i>borrowed history</i> yang baru dibuat serta memperbaiki bug dan tampilan pada halaman <i>borrowed history</i> dan melaporkan kepada mentor tentang progress proyek saat itu.
15	Menambahkan fitur search pada halaman <i>borrowed history</i> , lalu menambahkan halaman baru untuk melakukan <i>request</i> pembelian barang baru, serta mendownload laravel-dumpdf agar setiap <i>request</i> pembelian barang yang dilakukan bisa di download sebagai <i>pdf</i> , melaporkan kepada mentor tentang progress proyek saat itu serta mengadakan diskusi dengan designer UI/UX.
16	Membantu tim IT dalam menyiapkan acara donor darah serta mengikuti kegiatan yang diadakan, memperbaiki bug pada halaman <i>request</i> pembelian dimana <i>pdf</i> tidak bisa didownload dan memperbaiki tampilan halamannya. Membuat halaman baru untuk mengembalikan barang yang pernah dipinjam serta melaporkan kepada mentor progress proyek saat itu..
17	Membantu tim IT menyiapkan acara yang diadakan di dalam perusahaan. Membuat halaman baru untuk menampilkan <i>history</i> barang yang pernah dikembalikan serta memberikan laporan mingguan kepada mentor tentang progress proyek saat itu.
18	Membantu tim IT menyiapkan acara seleksi penghargaan karya inovasi PT PLN Nusantara Power. Memperbaiki <i>bug</i> pada halaman <i>Item Return</i> dan <i>Returned Items History</i> serta melakukan laporan mingguan kepada mentor.
19	Memperbaiki <i>bug</i> dan tampilan pada halaman <i>Item Return</i> dan <i>Returned Items History</i> .
20	Membantu tim IT menyiapkan acara yang diadakan dalam perusahaan, serta melakukan <i>debugging</i> untuk fitur yang dikerjakan dan memastikan tampilan sudah sesuai dengan figma yang diberikan oleh <i>designer UI/UX</i> .

Minggu Ke -	Pekerjaan yang dilakukan
21	Mempresentasikan hasil akhir dari proyek yang telah dikerjakan serta melakukan revisi untuk bagian-bagian yang diberikan masukan oleh Supervisor, Mentor dan Tim IT dan membahas untuk melakukan hostingan kepada <i>website</i>

### 3.3.1 Software dan Hardware Yang Digunakan

Selama pelaksanaan kegiatan magang, digunakan beberapa perangkat lunak (*Software*) dan perangkat keras (*Hardware*) yang mendukung proses pengembangan sistem. Adapun perangkat yang digunakan dijelaskan sebagai berikut:

#### a. Perangkat Lunak (*Software*):

1. Visual Studio Code, digunakan sebagai editor utama dalam proses pengembangan sistem berbasis web.
2. XAMPP (MariaDB dan MySQL), digunakan sebagai server lokal untuk menjalankan dan mengelola basis data selama proses pengembangan.
3. Git, digunakan untuk pengelolaan versi kode sumber secara lokal.
4. GitHub, digunakan sebagai platform kolaborasi dan penyimpanan repositori proyek secara daring.

#### b. Perangkat Keras (*Hardware*):

*Hardware* yang digunakan selama pengembangan adalah Laptop ASUS ROG Zephyrus G14 GA401, dengan spesifikasi sebagai berikut:

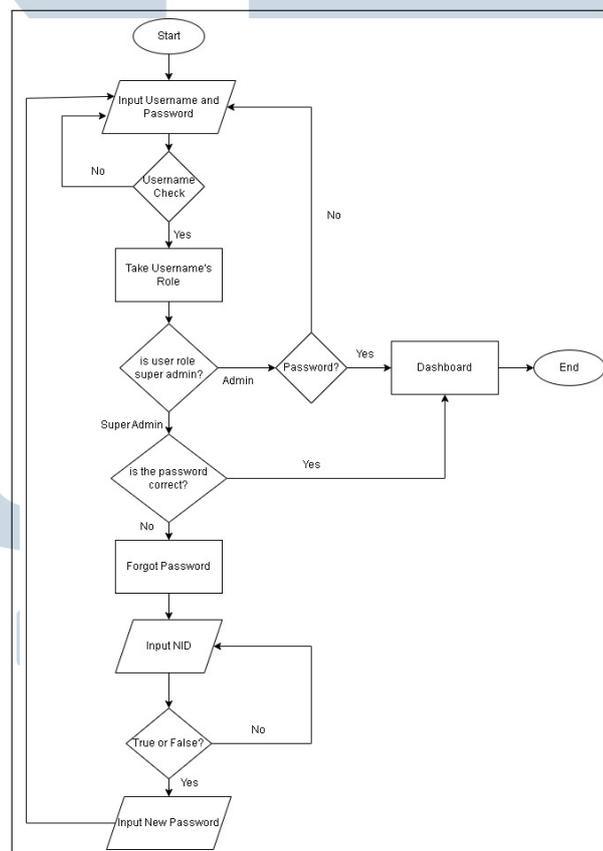
1. Prosesor: AMD Ryzen 9 4900HS
2. Kartu Grafis: NVIDIA GeForce RTX 2060 Max-Q
3. RAM: 16 GB DDR4 3200 MHz
4. Penyimpanan: SSD NVMe 1 TB
5. Layar: 14 inci, resolusi Quad HD (2560 x 1440)
6. Sistem Operasi: Windows 11 Home

### 3.4 Perancangan dan Hasil

#### 3.4.1 Flowchart

*Flowchart* merupakan representasi grafis dari suatu algoritma atau prosedur yang disusun menggunakan simbol-simbol baku seperti persegi panjang, belah ketupat, dan panah untuk menggambarkan alur proses secara sistematis dan logis. *flowchart* berfungsi sebagai alat bantu untuk merancang solusi terhadap permasalahan secara bertahap, serta mempermudah dalam proses perancangan sistem, analisis, *debugging*, dan dokumentasi program. Selain itu, *flowchart* juga berperan dalam memperjelas struktur kontrol suatu aplikasi dan meningkatkan efektivitas komunikasi antar anggota tim dalam pengembangan perangkat lunak [8].

#### A Flowchart Login



Gambar 3.1. Flowchart Login

*Flowchart* yang ada pada Gambar 3.1 menggambarkan alur proses autentikasi pengguna dalam sistem informasi yang telah diterapkan. Proses ini mencakup tahapan login pengguna hingga mekanisme pemulihan kata sandi apabila terjadi kesalahan dalam proses autentikasi.

Alurnya dimulai ketika pengguna memasukkan *username* dan *password* pada halaman login. Selanjutnya, sistem akan melakukan verifikasi untuk memastikan apakah *username* tersebut terdaftar dalam basis data. Apabila *username* yang dimasukkan tidak valid atau tidak ditemukan, maka sistem akan mengarahkan pengguna untuk menginput ulang *username* dan *password* hingga data yang dimasukkan sesuai.

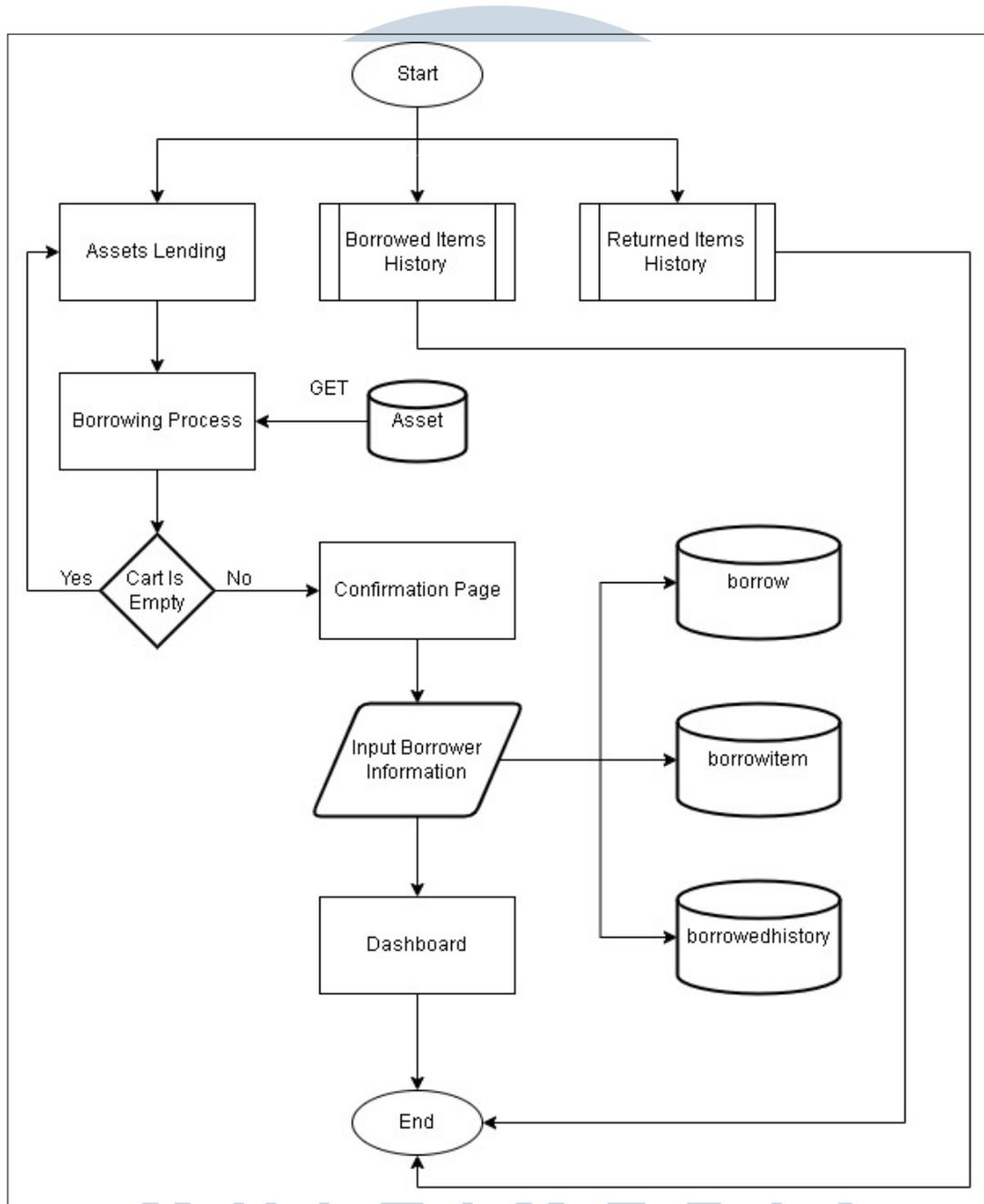
Apabila *username* terverifikasi, sistem akan mengambil informasi mengenai peran pengguna (*role*), yaitu sebagai *Admin* atau *Super Admin*. Kemudian, sistem akan melakukan validasi terhadap *password* yang dimasukkan. Jika *password* benar, maka pengguna akan diarahkan menuju halaman *Dashboard*, dan proses autentikasi dinyatakan berhasil.

Namun, apabila pengguna memiliki peran sebagai *Super Admin* dan *password* yang dimasukkan tidak valid, maka sistem akan menawarkan opsi pemulihan kata sandi melalui fitur "*Forgot Password*". Pada tahap ini, pengguna diminta untuk menginput Nomor Induk Diri (NID) sebagai langkah verifikasi identitas. Setelah NID dimasukkan, sistem akan memvalidasi keabsahan data tersebut.

Jika NID yang dimasukkan valid, pengguna akan diarahkan untuk membuat *password* baru. Setelah proses ini selesai, pengguna dapat kembali ke halaman *login* dan mengakses sistem dengan menggunakan *password* yang telah diperbarui. Sebaliknya, jika NID tidak valid, sistem akan meminta pengguna untuk kembali menginput NID yang sesuai.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

## B Flowchart Asset Lending



Gambar 3.2. Flowchart Asset Lending

*Flowchart* pada Gambar 3.2 menggambarkan alur sistem peminjaman aset yang diterapkan pada sistem. Proses ini mencakup tahapan mulai dari pemilihan aset yang akan dipinjam, input data peminjam, hingga penyimpanan histori

peminjaman ke dalam basis data.

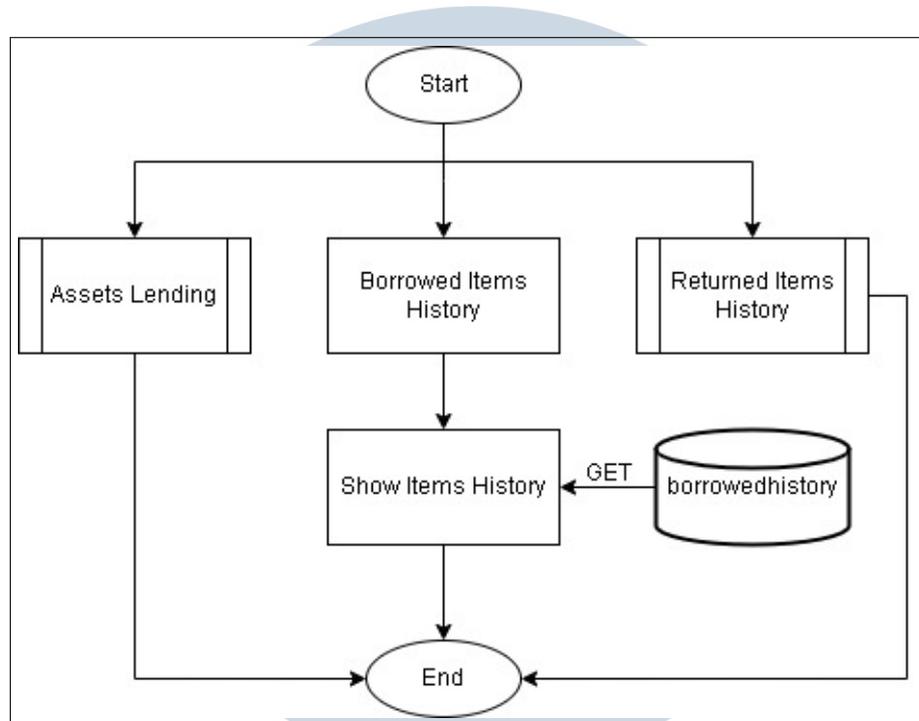
Proses dimulai ketika pengguna memasuki sistem dan dihadapkan pada tiga pilihan utama, yaitu *Assets Lending*, *Borrowed Items History*, dan *Returned Items History*. Pemilihan menu *Assets Lending* akan mengarahkan pengguna ke tahap *Borrowing Process*, yakni proses pemilihan aset yang tersedia dalam sistem. Data aset diperoleh melalui metode *GET*, yang memungkinkan pengguna untuk menambahkan barang ke dalam keranjang peminjaman.

Setelah proses pemilihan aset dilakukan, sistem akan memverifikasi apakah keranjang peminjaman dalam kondisi kosong. Apabila keranjang masih kosong, sistem akan mengarahkan pengguna kembali ke tahap sebelumnya untuk menambahkan aset. Namun, jika keranjang telah berisi *item*, maka pengguna akan diarahkan ke halaman konfirmasi.

Pada halaman ini, pengguna diwajibkan untuk mengisi formulir informasi peminjam yang meliputi data diri serta detail lainnya yang relevan. Data yang dimasukkan akan dikirimkan ke server menggunakan metode *POST* dan selanjutnya disimpan ke dalam tiga entitas basis data, yaitu tabel *borrow*, *borrowitem*, dan *borrowedhistory*. Setelah seluruh data berhasil disimpan, pengguna akan diarahkan kembali ke *dashboard* utama sistem sebagai tanda bahwa proses telah selesai.



### C Flowchart Borrowed History



Gambar 3.3. Flowchart Borrowed History

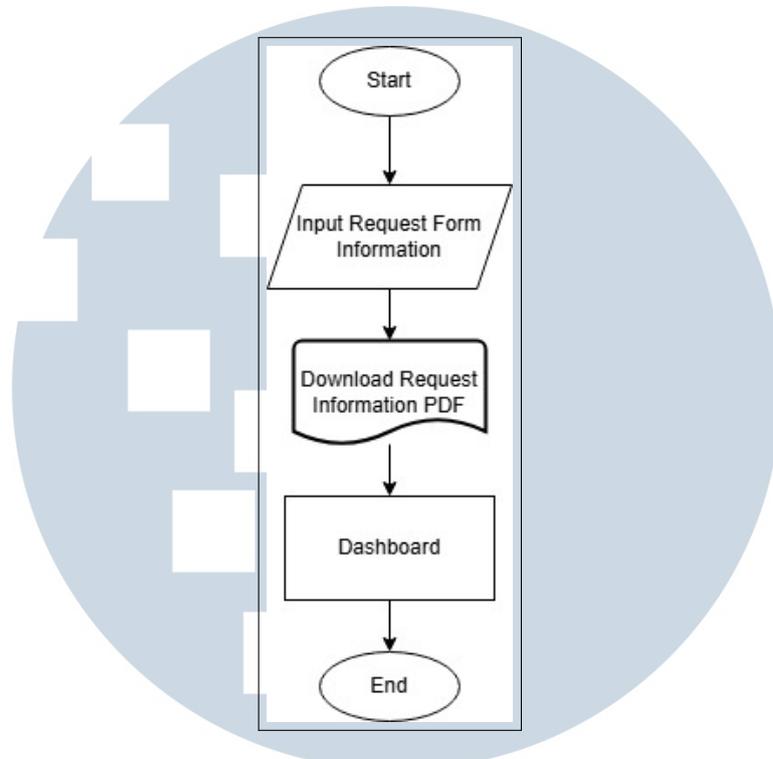
Gambar flowchart pada Gambar 3.3 menggambarkan alur proses penelusuran riwayat peminjaman barang (*Borrowed Items History*) pada sistem informasi manajemen aset.

Proses dimulai ketika pengguna mengakses sistem dan diberikan tiga opsi utama: *Assets Lending*, *Borrowed Items History*, dan *Returned Items History*. Apabila pengguna memilih menu *Borrowed Items History*, maka sistem akan menavigasi pengguna menuju proses berikutnya, yaitu *Show Items History*.

Pada tahap ini, sistem akan melakukan pengambilan data histori peminjaman barang dengan metode *GET* dari basis data yang bernama *borrowedhistory*. Basis data ini menyimpan catatan seluruh transaksi peminjaman yang telah terjadi sebelumnya, mencakup informasi seperti identitas peminjam, tanggal peminjaman, daftar aset yang dipinjam, dan status pengembalian.

Setelah data berhasil diambil dari basis data, sistem akan menampilkan informasi tersebut kepada pengguna dalam bentuk daftar riwayat peminjaman. Tahapan ini menandai berakhirnya proses, dan sistem siap menerima interaksi lanjutan atau kembali ke halaman awal (*Start*).

## D Flowchart Request

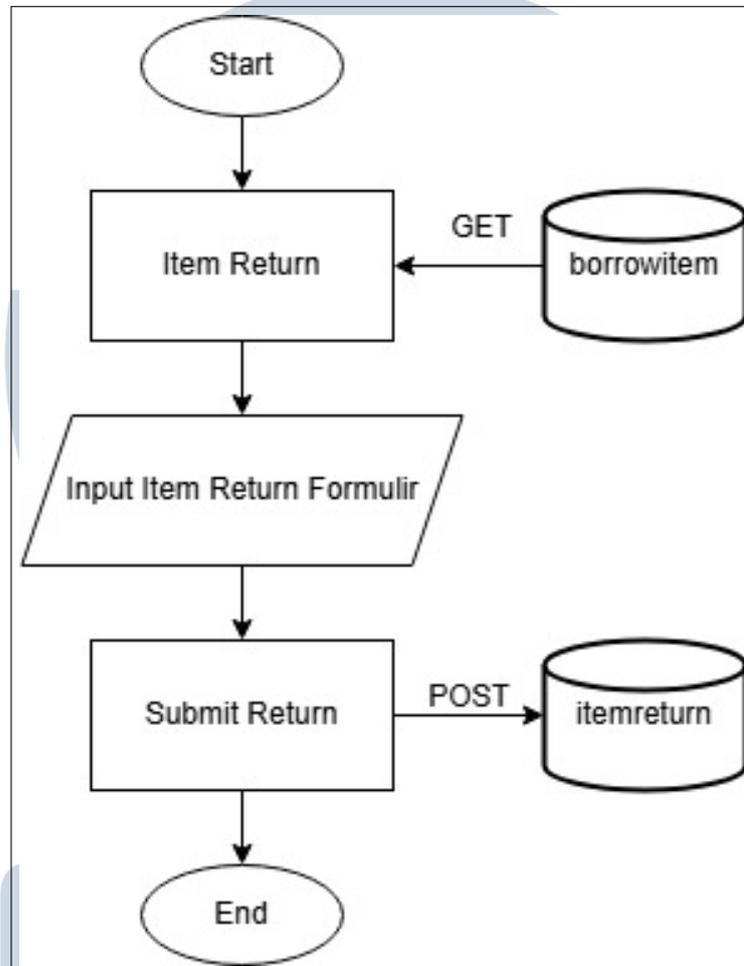


Gambar 3.4. Flowchart Request

*Flowchart* yang ditunjukkan pada Gambar 3.4 menggambarkan alur kerja proses permintaan untuk pembelian barang. Proses akan dimulai saat pengguna memasuki halaman *Request* pada *website*, setelah itu pengguna akan di minta untuk melakukan *input* data yang dibutuhkan untuk melakukan proses pembelian barang baru seperti nama pengirim, nama penerima, department dan NID penerima, tanggal request dilakukan, dan tipe barang yang ingin di beli.

Setelah pengguna melakukan submit *request* maka pengguna akan diberikan pilihan untuk mengunduh *pdf* yang berisi template yang dibutuhkan untuk melakukan request pembelian barang baru atau bisa langsung kembali ke *dashboard*, setelah itu proses akan selesai.

## E Flowchart Item Return



Gambar 3.5. Flowchart Item Return

Flowchart yang ditunjukkan dalam Gambar 3.5 adalah proses yang terjadi saat pengguna ingin mengembalikan barang

Langkah pertama yang dilakukan oleh sistem adalah menampilkan halaman atau modul *Item Return*. Pada tahap ini, sistem melakukan permintaan data dari basis data dengan menggunakan metode *GET*. Permintaan ini diarahkan ke entitas *borrowitem*, yang merepresentasikan data barang-barang yang sedang dipinjam oleh pengguna. Melalui proses *GET* ini, informasi tentang barang yang dapat dikembalikan ditampilkan kepada pengguna.

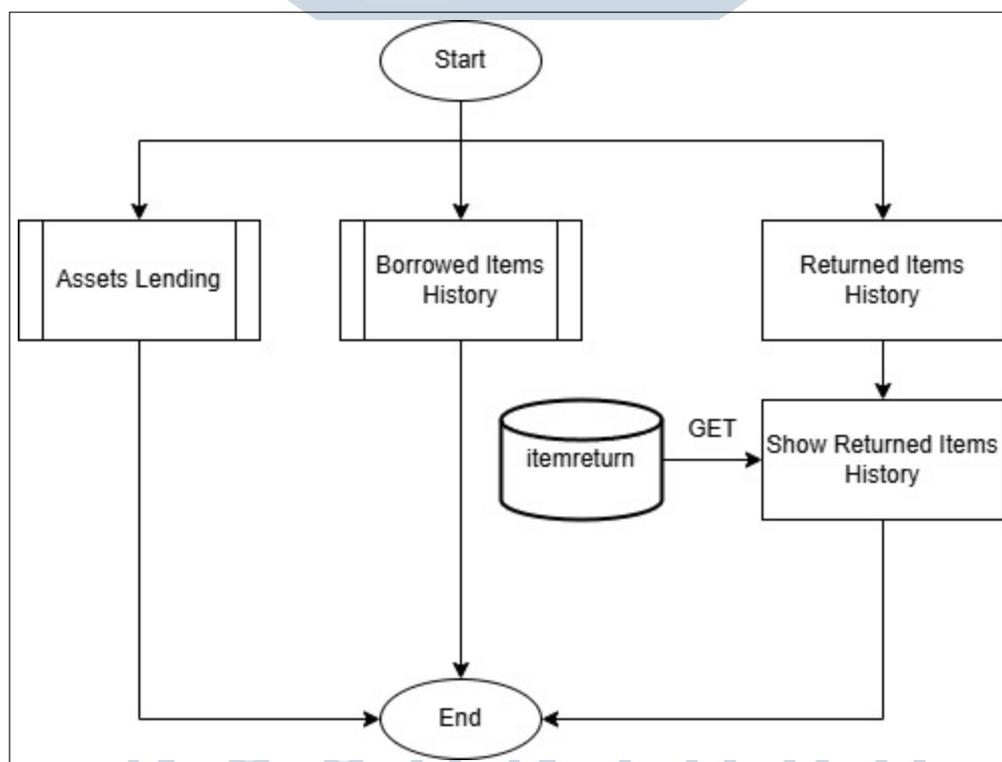
Setelah data ditampilkan, pengguna diminta untuk mengisi formulir pengembalian barang, yang ditunjukkan dalam simbol proses berbentuk jajar

genjang. Pengisian formulir ini mencakup informasi-informasi penting seperti identitas barang, tanggal pengembalian, dan mungkin alasan pengembalian atau kondisi barang saat dikembalikan. Tahap ini mencerminkan adanya interaksi langsung dari pengguna dalam menyediakan data.

Setelah formulir diisi, pengguna menekan tombol *Submit Return* untuk mengirimkan data tersebut ke sistem. Proses ini digambarkan dengan simbol proses biasa dan mengindikasikan bahwa data hasil pengisian formulir dikirim ke server menggunakan metode *POST*. Pengiriman ini diarahkan ke entitas basis data bernama *itemreturn*, yang berfungsi sebagai tempat penyimpanan data pengembalian barang.

Proses berakhir ketika data telah berhasil disimpan ke dalam basis data, yang ditandai dengan simbol "End". Ini menunjukkan bahwa sistem telah menyelesaikan proses bisnis pengembalian barang secara utuh dan sesuai prosedur.

#### F Flowchart Returned Items History



Gambar 3.6. Flowchart Returned Items History

*Flowchart* yang di tunjukkan di Gambar 3.6 menggambarkan alur proses untuk melihat history dari barang yang sudah dikembalikan dalam sebuah sistem manajemen aset.

Setelah memasuki sistem, pengguna disajikan tiga pilihan utama. Pilihan ketiga adalah melihat riwayat barang yang telah dikembalikan, ditunjukkan oleh bagian "*Returned Items History*". Ketika pengguna memilih menu ini, sistem melakukan permintaan data dari basis data bernama "*itemreturn*" menggunakan metode *HTTP GET*. Permintaan data ini bertujuan untuk mengambil informasi mengenai barang-barang yang sudah dikembalikan ke sistem.

Setelah data berhasil diambil dari basis data, sistem menampilkan informasi tersebut kepada pengguna dalam bentuk antarmuka "*Show Returned Items History*". Tampilan ini memberikan informasi lengkap terkait barang yang telah dikembalikan, seperti nama barang, tanggal pengembalian, serta pengguna yang melakukan pengembalian.

### **3.4.2 Tampilan Halaman dan Potongan Kode Website**

#### **A Tampilan dan Potongan Kode Halaman Login**

##### **A.1 Tampilan Halaman Login**

Pada modal *Login* pengguna akan di minta untuk memasukkan Nama dan *Password* agar bisa mengakses halaman utama *website*, dan sistem akan melakukan pengecekan untuk Nama yang di input mempunyai *role Admin* atau *Super Admin*.

UIN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

Sign in to PLN IT Asset Management

PLN  
Nusantara Power

Nama:

Wrong Password.

Password:

Login

Gambar 3.7. Tampilan Login Admin Gagal

Pada Gambar 3.7 adalah contoh saat pengguna menginput *password* yang salah tetapi Nama yang di input memiliki *role Admin*.

UMMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

Sign in to PLN IT Asset Management

PLN  
Nisantara Power

Nama:

Wrong Password.

Password:

Login

Forgot Password (Super Admin)

Gambar 3.8. Tampilan Login Super Admin Gagal

Super Admin NID Verification

NID:

Verify Cancel

Gambar 3.9. Tampilan Verifikasi NID

Pada Gambar 3.8 adalah contoh tampilan saat pengguna menginput *password* salah tetapi Nama yang di input memiliki *role Super Admin* dan pada gambar 3.9 adalah tampilan *pop up* verifikasi NID yang akan muncul jika pengguna dengan *role Super Admin* tersebut ingin mengubah password yang lama.

The image shows a modal window titled "Reset New Password" with a close button (X) in the top right corner. Below the title, there is a label "New Password:" followed by a single-line text input field. At the bottom right of the modal, there are two buttons: a green "Reset Password" button and a grey "Cancel" button.

Gambar 3.10. Tampilan Reset Password

Pada Gambar 3.10 adalah contoh tampilan dari *popup* saat pengguna dengan *role Super Admin* ingin mengubah *password* nya.

The image shows a modal window titled "Sign in to PLN IT Asset Management" with a close button (X) in the top right corner. Below the title is the PLN Nisantara Power logo. There are two input fields: "Nama:" and "Password:". Below the input fields is a green "Login" button. At the bottom of the modal, there is a light green message box that says "Password Successfully Reseted."

Gambar 3.11. Tampilan Reset Password

Lalu pada Gambar 3.11 adalah contoh tampilannya saat pengguna dengan *role Super Admin* berhasil mengubah *password* lama nya, dan setelah melakukan *login* maka akan bisa mengakses halaman utama *website*.

## A.2 Potongan Kode Login Page

```
// Jika salah password tapi user adalah super admin
if (strtolower($user->Role) === 'super admin') {
    Session::put('superadmin_name', $user->Name);
    return back()->withErrors(['super_admin_wrong_password' => true]);
}
```

Gambar 3.12. Potongan Kode Check Role User

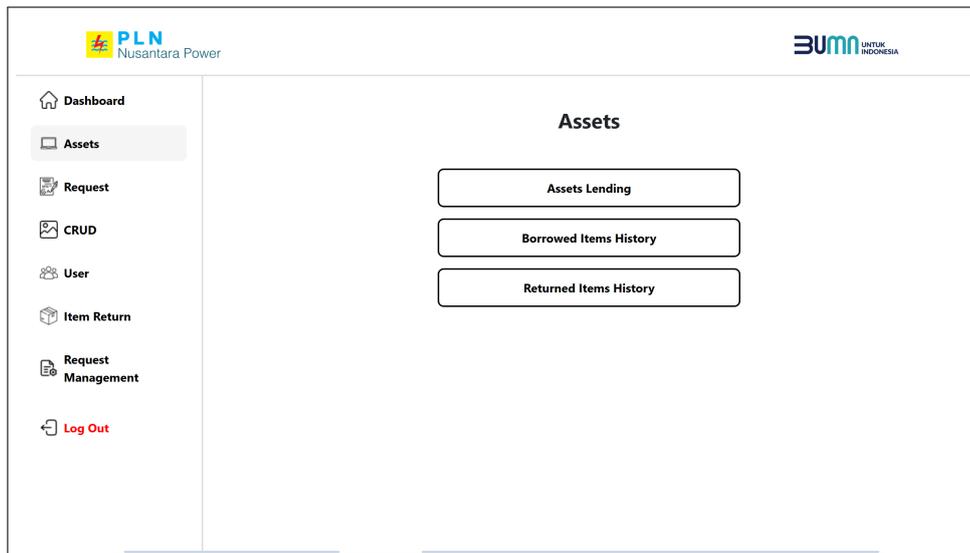
Potongan kode pada Gambar 3.12 digunakan untuk menangani kondisi ketika pengguna yang memiliki peran sebagai *super admin* salah memasukkan kata sandi. Sistem akan memeriksa apakah peran pengguna adalah *super admin* dengan mengubah nilai *role* menjadi huruf kecil agar sama. Jika benar, maka nama pengguna akan disimpan ke dalam *session* dengan kunci *superadmin\_name*. Setelah itu, sistem akan mengarahkan kembali ke halaman sebelumnya dengan membawa pesan kesalahan bahwa super admin salah memasukkan kata sandi, sehingga dapat ditindaklanjuti dengan proses verifikasi NID untuk mereset kata sandi.

## B Tampilan dan Potongan Kode Assets & Lending Page

### B.1 Tampilan Halaman Assets & Lending

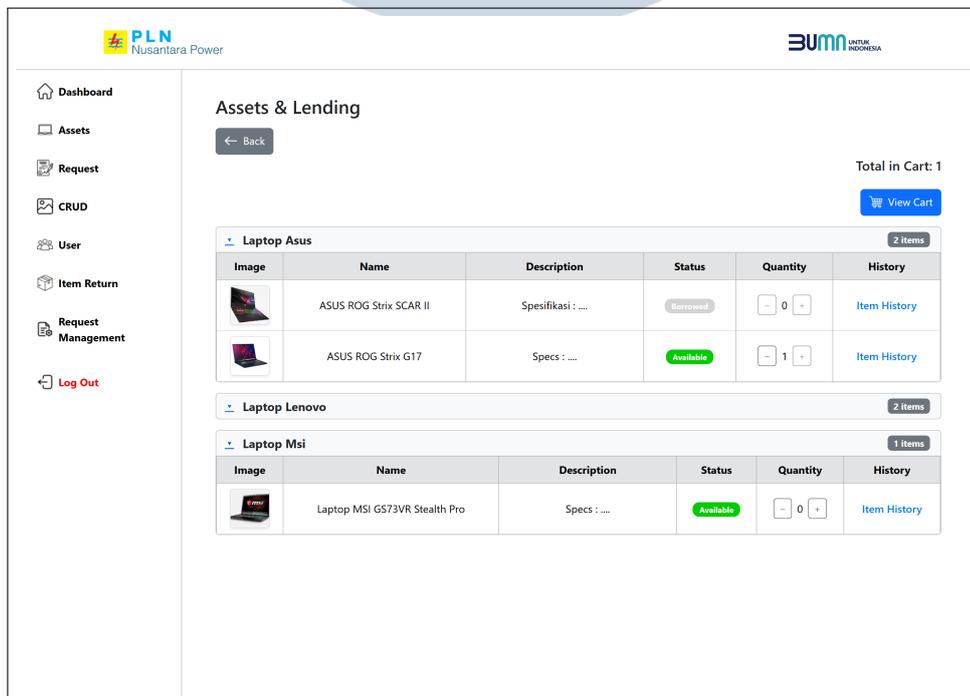
*Assets Lending* adalah salah satu bagian yang ada pada halaman *Assets* yang bisa di navigasi melalui *sidebar* yang ada di kiri bagian *website*.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 3.13. Tampilan Assets

Gambar 3.13 adalah contoh tampilan *website* saat pengguna melakukan navigasi ke *sidebar Assets*.

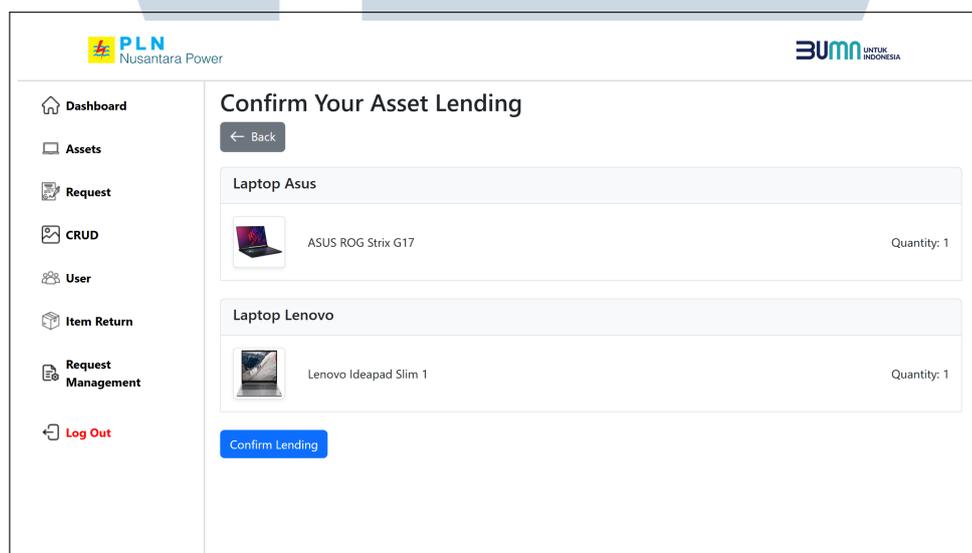


Gambar 3.14. Tampilan Assets Lending

Gambar 3.14 menampilkan halaman *Assets & Lending* yang digunakan

untuk memantau dan meminjam aset. Setiap jenis aset dipisahkan menggunakan *dropdown* berdasarkan mereknya seperti Asus, Lenovo, dan Msi. Di dalam setiap *dropdown* terdapat daftar laptop dengan informasi gambar, nama, deskripsi, status ketersediaan, jumlah yang dapat dipinjam, serta tautan untuk melihat riwayat peminjaman masing-masing aset.

Jika suatu aset sedang dipinjam, maka statusnya akan ditandai sebagai "Borrowed" dan tidak dapat ditambahkan ke dalam keranjang. Sebaliknya, jika tersedia, statusnya akan menjadi "Available" dan pengguna hanya dapat memilih jumlah yang ingin ditambahkan ke dalam keranjang sebanyak satu untuk setiap aset. Total item yang telah dipilih akan ditampilkan di bagian kanan atas halaman beserta tombol untuk melihat isi keranjang. Pada bagian atas kiri halaman juga ada tombol *Back* yang dapat digunakan untuk kembali ke halaman sebelumnya.



Gambar 3.15. Tampilan View Cart

Gambar 3.15 menampilkan halaman konfirmasi peminjaman aset. Pada halaman ini, pengguna dapat melihat daftar aset yang telah dimasukkan ke dalam keranjang (*cart*) sebelum dilakukan proses peminjaman akhir. Setiap aset dikelompokkan berdasarkan tipe atau mereknya, seperti "Laptop Asus" dan "Laptop Lenovo". Masing-masing kelompok menampilkan informasi nama aset, gambar, dan jumlah (*quantity*) yang akan dipinjam.

Jika aset berasal dari tipe atau kategori yang berbeda, maka akan ditampilkan secara terpisah agar lebih mudah dikenali. Di bagian bawah terdapat tombol "Confirm Lending" yang berfungsi untuk mengonfirmasi dan memproses

peminjaman aset-aset tersebut. Dibagian atas kiri halaman juga ada tombol *Back* yang dapat digunakan untuk melakukan navigasi kembali ke halaman sebelumnya.

The screenshot shows a web interface for asset borrowing. On the left is a sidebar with navigation links: Dashboard, Assets, Request, CRUD, User, Item Return, Request Management, and Log Out. The main content area is titled 'Borrower Information' and contains a 'Back' button at the top left. Below it are input fields for 'Full Name', 'Department', and 'NID'. There are two date pickers: 'Borrow Date' and 'Due Date', both with a format of 'dd/mm/yyyy'. Below the date pickers is the 'Asset Reasons' section, which is pre-filled with 'ASUS ROG Strix G17'. At the bottom of the form is a green 'Submit Borrow Request' button.

Gambar 3.16. Tampilan Formulir Borrower Information

Pada Gambar 3.16 menampilkan halaman formulir peminjaman aset. Pengguna diminta untuk mengisi informasi data diri peminjam seperti nama lengkap, departemen, NID, tanggal mulai peminjaman, dan tanggal pengembalian.

Di bagian bawah, terdapat kolom untuk mengisi alasan peminjaman aset yang ditampilkan, sebagai contoh yaitu ASUS ROG Strix G17 dan jika aset yang dipinjam lebih dari satu maka di bagian *Asset Reasonnya* akan muncul lebih dari satu kotak yang dapat digunakan untuk mengisi alasan dari peminjaman setiap aset. Setelah semua data diisi, pengguna dapat menekan tombol "Submit Borrow Request" untuk mengajukan permintaan peminjaman aset. Terdapat juga tombol *Back* di bagian atas untuk kembali ke halaman sebelumnya.

## B.2 Potongan Kode Assets & Lending

```
function syncCartToServer() {
  fetch('/cart/save', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
      'X-CSRF-TOKEN': '{{ csrf_token() }}'
    },
    body: JSON.stringify({
      cart
    })
  })
  .then(res => res.json())
  .then(data => console.log('Cart synced', data))
  .catch(err => console.error('Sync error', err));
}
```

Gambar 3.17. Potongan Code Function syncCartToServer

Fungsi *syncCartToServer()* yang ada pada Gambar 3.17 digunakan untuk mengirim data keranjang (*cart*) dari sisi *client* ke *server* menggunakan metode *fetch* dengan request *POST* ke *endpoint* */cart/save*. Data keranjang dikirim dalam format JSON, dan disertai token CSRF untuk keamanan. Setelah data berhasil dikirim, respon dari *server* akan ditampilkan di *console*. Jika terjadi kesalahan, pesan *error* akan ditampilkan di *console* browser. Fungsi ini bertujuan agar isi keranjang pengguna dapat disimpan sementara di *server*, khususnya dalam *session* Laravel.

```
public function saveCart(Request $request)
{
  $cart = $request->input('cart', []);
  session()->put('cart', $cart);

  return response()->json(['message' => 'Cart saved to session', 'cart' => $cart]);
}
```

Gambar 3.18. Potongan Code Function saveCart

Pada Gambar 3.18 Fungsi *saveCart* adalah fungsi pada *controller* Laravel yang digunakan untuk menerima dan menyimpan data keranjang (*cart*) dari sisi

*client* ke dalam *session server*. Fungsi ini menerima permintaan HTTP (*request*) yang berisi data keranjang dalam format JSON, yang dikirim oleh fungsi JavaScript *syncCartToServer()* yang ada pada Gambar 3.17 menggunakan metode *fetch* dengan method *POST*.

Data *cart* kemudian diambil dari *request* menggunakan `$request->input('cart', [])`, dan disimpan ke dalam *session* dengan perintah `session()->put('cart', $cart)`. Setelah berhasil disimpan, fungsi ini mengembalikan respon dalam bentuk JSON yang berisi pesan bahwa data berhasil disimpan dan menampilkan isi keranjangnya. Fungsi ini memungkinkan data keranjang tetap tersimpan di server meskipun halaman berpindah, sehingga pengguna bisa melanjutkan proses seperti konfirmasi atau pengisian form peminjaman dengan data yang tetap konsisten.

```
// Fungsi update tampilan quantity dan tombol
function updateQuantity(id) {
    const quantityElement = document.getElementById(`quantity-${id}`);
    const increaseButton = document.querySelector(`.increase-btn[data-id="${id}"]`);
    const decreaseButton = document.querySelector(`.decrease-btn[data-id="${id}"]`);

    quantityElement.textContent = cart[id] || 0;

    if (!cart[id]) {
        if (increaseButton) increaseButton.disabled = false;
        if (decreaseButton) decreaseButton.disabled = true;
    } else {
        if (increaseButton) increaseButton.disabled = true;
        if (decreaseButton) decreaseButton.disabled = false;
    }

    updateCartSummary();
    syncCartToServer();
}
```

Gambar 3.19. Potongan Code Function *updateQuantity*

Fungsi *updateQuantity(id)* pada Gambar 3.19 bertujuan untuk memperbarui tampilan jumlah suatu aset dalam keranjang (*cart*) berdasarkan ID aset yang diberikan. Fungsi ini juga mengatur status tombol “+” (tambah) dan “-” (kurang) sesuai dengan kondisi isi keranjang.

fungsi ini mengambil elemen tampilan jumlah (*quantityElement*) dan tombol tambah (*increaseButton*) serta tombol kurang (*decreaseButton*) berdasarkan ID aset. Kemudian, nilai jumlah yang ditampilkan akan diperbarui sesuai isi *cart*: jika belum ada, akan ditampilkan 0.

Jika aset tersebut belum dimasukkan ke dalam *cart*, maka tombol “+” akan

diaktifkan agar pengguna bisa menambahkannya, sedangkan tombol “-” akan dinonaktifkan karena belum ada yang bisa dikurangi. Sebaliknya, jika aset sudah ada dalam cart, maka tombol “+” akan dinonaktifkan untuk mencegah penambahan lebih dari satu, sementara tombol “-” diaktifkan agar pengguna bisa mengurangi atau menghapusnya dari keranjang.

fungsi ini juga akan memanggil *updateCartSummary()* Gambar 3.20 untuk memperbarui total item yang ada di keranjang dan *syncCartToServer()* untuk menyimpan data cart ke server agar tetap sinkron saat berpindah halaman atau melakukan proses lainnya.

```
// Update total item di keranjang
function updateCartSummary() {
  const total = Object.values(cart).reduce((sum, qty) => sum + qty, 0);
  document.getElementById('cart-summary').textContent = `Total in Cart: ${total}`;
}
```

Gambar 3.20. Potongan Code Function *updateCartSummary*



```

cart[id] = 1;
updateQuantity(id);

// Kirim ke server via AJAX
fetch('/cart/add/' + id, {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
    'X-CSRF-TOKEN': '{{ csrf_token() }}'
  },
  body: JSON.stringify({
    quantity: 1
  })
})
.then(response => response.json())
.then(data => {
  console.log('Cart updated on server:', data);
})
.catch(error => {
  console.error('Error updating cart:', error);
});
syncCartToServer();

```

Gambar 3.21. Potongan Code Untuk Menyimpan Isi Cart

Potongan kode pada Gambar 3.21 ini digunakan saat pengguna menekan tombol “+” untuk menambahkan item ke dalam keranjang. Pertama, item dengan ID tertentu diset ke jumlah 1 di objek *cart*, lalu fungsi *updateQuantity(id)* dipanggil untuk memperbarui tampilan jumlah pada halaman dan menonaktifkan tombol “+” agar tidak bisa ditekan lagi. Setelah itu, data keranjang dikirim ke server melalui AJAX dengan *endpoint POST /cart/add/id*, bersama data kuantitas (*quantity: 1*) dalam format JSON. Bila berhasil, respon akan ditampilkan di konsol. Setelah semua proses selesai, fungsi *syncCartToServer()* dipanggil untuk menyimpan seluruh isi cart ke dalam session di server.

```
public function showFinalForm()
{
    $cart = session('cart', []);
    $assetIds = array_keys($cart);
    $assets = Asset::whereIn('AssetID', $assetIds)->get();
    return view('borrow_form', compact('assets'));
}
```

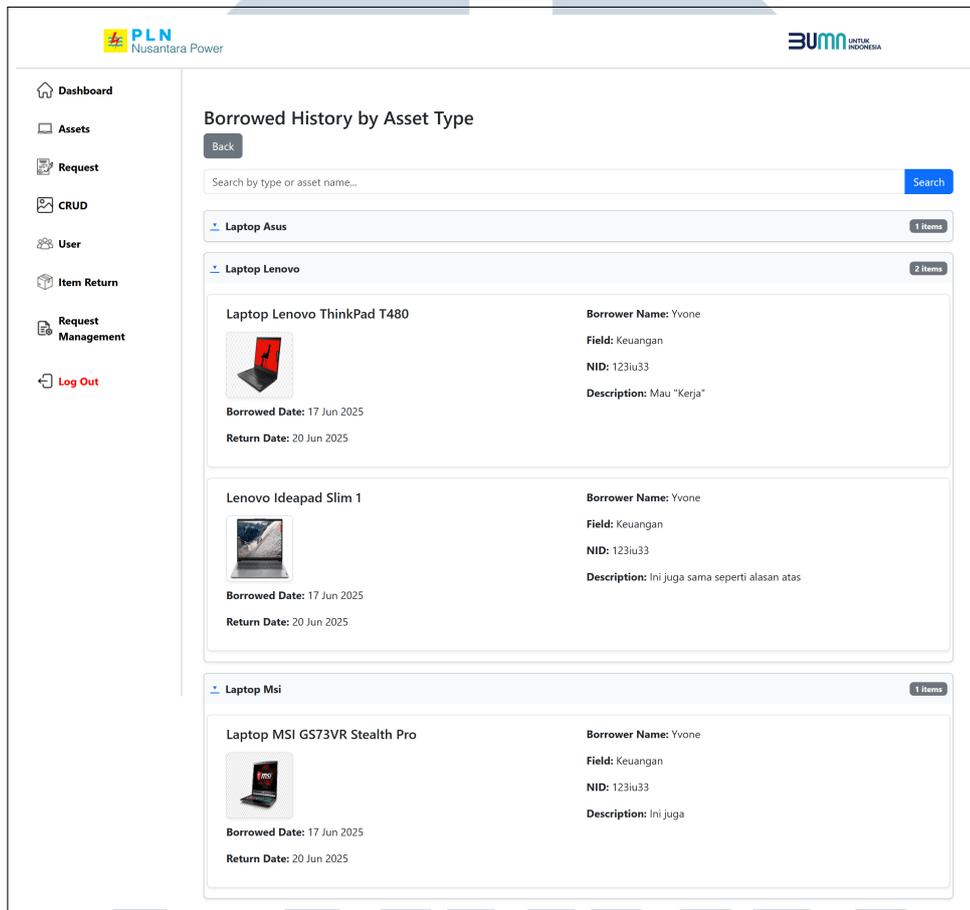
Gambar 3.22. Potongan Code Function showFinalForm

Pada Gambar 3.22 Fungsi *showFinalForm()* bertugas untuk menampilkan halaman form peminjaman. Pertama, fungsi ini mengambil data keranjang (*cart*) dari *session*, jika tidak ada maka akan digunakan array kosong sebagai default. Kemudian, dari keranjang tersebut diambil daftar ID aset menggunakan *array\_keys()*. ID-ID ini kemudian digunakan untuk mengambil data lengkap aset dari database menggunakan query *Asset::whereIn('AssetID', \$assetIds)->get()*. Terakhir, data aset yang berhasil diambil akan dikirim ke *view Blade* bernama *borrow\_form*, agar bisa ditampilkan pada halaman form peminjaman yang akan diisi oleh pengguna.

UMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

## C Tampilan dan Potongan Kode Borrowed Items History

### C.1 Tampilan Halaman Borrowed Items History



Gambar 3.23. Tampilan Halaman Borrowed Items History

Pada Gambar 3.23 Halaman tersebut menampilkan daftar riwayat peminjaman aset yang telah dikelompokkan berdasarkan tipe aset. Setiap tipe aset seperti "Laptop Asus", "Laptop Lenovo", dan "Laptop Msi" ditampilkan dalam bentuk dropdown, yang ketika dibuka akan menampilkan daftar aset dari tipe tersebut yang pernah dipinjam. Di sebelah kanan nama tipe aset terdapat informasi jumlah item yang termasuk dalam kategori tersebut.

Setiap item aset yang ditampilkan berisi informasi lengkap, termasuk nama aset, gambar aset, tanggal peminjaman, tanggal pengembalian, nama peminjam, bidang atau divisi, nomor identitas, dan deskripsi yang berisi alasan atau keterangan peminjaman. Informasi ini disusun secara rapi dalam bentuk kartu untuk

memudahkan pembacaan dan identifikasi masing-masing aset. Tampilan ini memberikan gambaran jelas mengenai siapa yang meminjam aset, kapan aset dipinjam dan dikembalikan, serta alasan peminjaman.

Di bagian atas halaman terdapat tombol kembali dan kolom pencarian. Kolom pencarian ini dapat digunakan untuk mencari riwayat peminjaman berdasarkan nama aset atau tipe aset, sehingga mempermudah pengguna dalam menemukan data yang diinginkan.

## C.2 Potongan Kode Borrowed Items History

```
public function index()
{
    $borrowedItems = BorrowedHistory::with(['asset', 'borrow'])->orderBy('LendingDate', 'desc')->get();

    return view('borrowed_history', compact('borrowedItems'));
}
```

Gambar 3.24. Potongan Code Function index

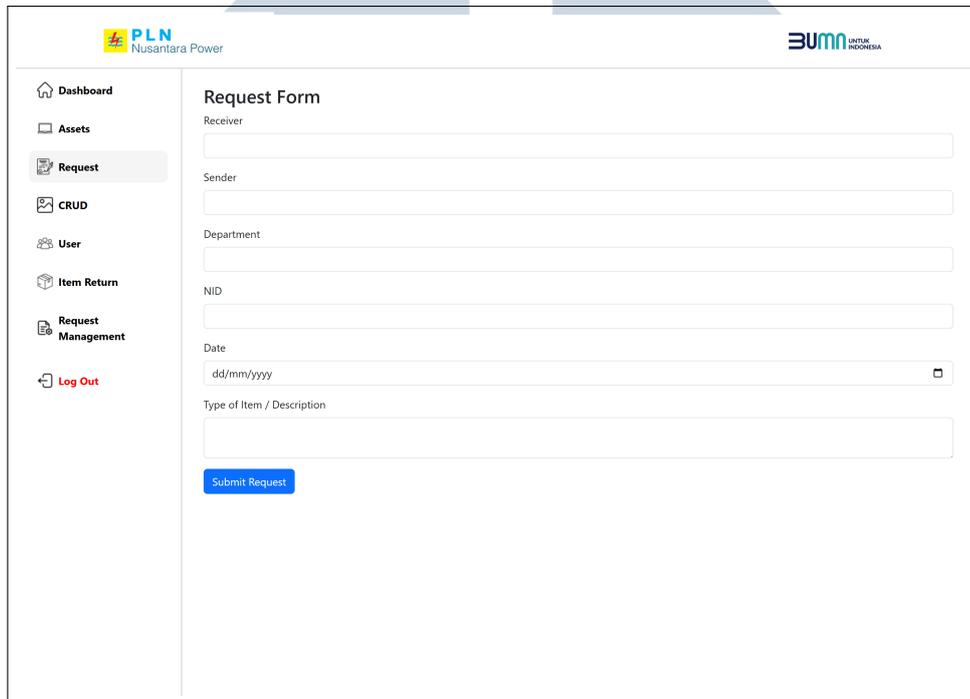
Fungsi *index()* pada Gambar 3.24 ini digunakan untuk menampilkan halaman daftar riwayat peminjaman aset. Fungsi ini mengambil semua data dari tabel *borrowedhistory* dengan memuat relasi *asset* dan *borrow* secara *eager loading*. Artinya, setiap data riwayat peminjaman akan otomatis disertai data aset yang dipinjam (*asset*) dan data peminjamnya (*borrow*).

Data tersebut diurutkan berdasarkan *LendingDate* (tanggal peminjaman) dari yang terbaru ke yang lama menggunakan *orderBy('LendingDate', 'desc')*. Setelah data berhasil diambil, fungsi ini mengembalikan view bernama *borrowed\_history* dan mengirimkan data *borrowedItems* ke dalam view tersebut agar bisa ditampilkan di halaman antarmuka pengguna.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

## D Tampilan dan Potongan Kode Halaman Request

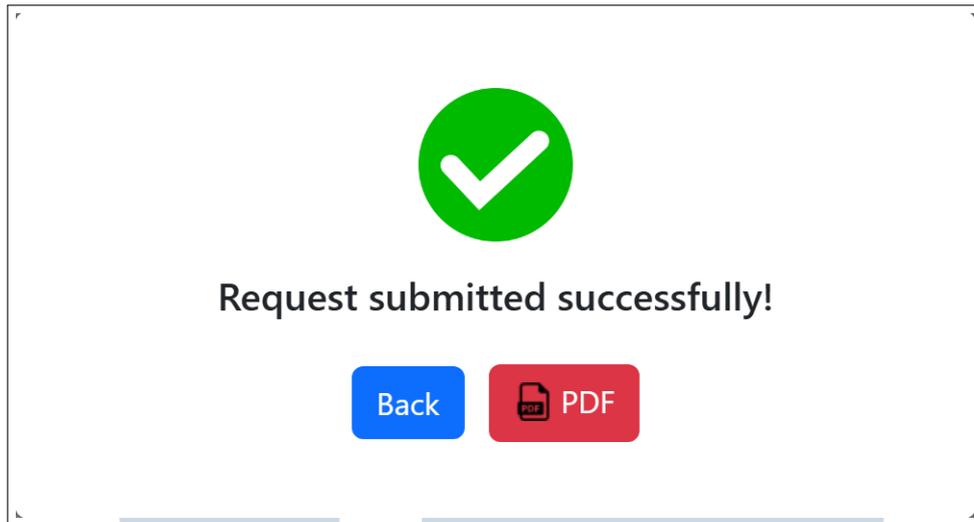
### D.1 Tampilan Halaman Request



The screenshot shows a web application interface for submitting a request. The page has a header with the PLN Nisantara Power logo on the left and the BUMA logo on the right. A sidebar on the left contains navigation links: Dashboard, Assets, Request (highlighted), CRUD, User, Item Return, Request Management, and Log Out. The main content area is titled "Request Form" and contains several input fields: Receiver, Sender, Department, NID, Date (with a date picker), and Type of Item / Description. A blue "Submit Request" button is located at the bottom of the form.

Gambar 3.25. Tampilan Halaman Request

Halaman pada Gambar 3.25 menampilkan *form* permintaan peminjaman barang yang terdiri dari beberapa isian penting. Pengguna diminta untuk mengisi nama penerima permintaan pada kolom "Receiver", nama pengirim atau pemohon pada kolom "Sender", serta unit kerja atau divisi pada kolom "Department". Selanjutnya, pengguna juga diminta untuk mengisi nomor identitas pada kolom "NID", memilih tanggal permintaan melalui kolom tanggal yang telah dilengkapi dengan date picker, dan menjelaskan jenis barang yang diminta atau keperluan peminjaman pada kolom "Type of Item / Description". Setelah seluruh data diisi dengan benar, pengguna dapat menekan tombol "Submit Request" yang tersedia di bagian bawah untuk mengajukan permintaan. Tampilan *form* ini dibuat sederhana dan terstruktur agar mudah dipahami dan diisi oleh pengguna.



Gambar 3.26. Tampilan Popup Request

Gambar 3.26 menunjukkan saat pengguna sudah selesai menginput data yang di butuhkan untuk melakukan request pembelian barang seperti Nama Pengirim, Nama Penerima, Departemen Penerima, NID Penerima, Tanggal Permintaan Dibuat, dan Tipe item serta alasan untuk permintaan pembelian tersebut dibuat.



Jalan Pluit Karang Ayu Utara, Jakarta 14450  
Telp: 021 - 6600054  
Fax: 021 - 6692806

---

---

### Tanda Terima Barang

Telah diterima barang dengan rincian:

**Item** : Laptop Asus - Spek apa saja buat "kerja"  
**Dari** : Cuan Zefanya  
**Kepada** : Yvone  
**Tanggal Permohonan** : 17 June 2025

Demikian surat ini kami buat. Atas perhatian dan kerja samanya kami ucapkan terima kasih.

Jakarta, Tuesday, 17 June 2025

Yang Menyerahkan

Penerima

Cuan Zefanya

Yvone

Gambar 3.27. Tampilan Contoh Request PDF

Pada Gambar 3.27 adalah contoh tampilan dari *PDF* jika pengguna memilih untuk mendownload *PDF* dari *request* yang telah dibuat, pada *PDF* tersebut terdapat isi dari informasi yang telah di isi sebelumnya seperti nama pengirim, nama penerima, departemen penerima, NID penerima, Tanggal permohonan dibuat, dan tipe item serta alasan untuk permintaan pembelian tersebut dibuat. lalu akan ada tempat untuk tanda tangan oleh pengirim barang dan penerima barang.

## D.2 Code Halaman Request

```
public function store(Request $request)
{
    $validated = $request->validate([
        'Receiver' => 'required|string|max:100',
        'Sender' => 'required|string|max:100',
        'Field' => 'required|string|max:30',
        'NID' => 'required|string|max: 50',
        'RequestDate' => 'required|date',
        'RequestDesc' => 'nullable|string',
    ]);

    $newRequest = RequestModel::create($validated);

    // Simpan data request baru ke session untuk modal & PDF
    return redirect()->route('request')
        ->with('success', 'Request submitted successfully!')
        ->with('requestData', $newRequest);
}
```

Gambar 3.28. Potongan kode Function store

Potongan kode pada Gambar 3.28 tersebut merupakan fungsi *store* yang bertugas untuk menangani data permintaan yang dikirimkan melalui *form*. Pertama-tama, data yang dikirim akan divalidasi untuk memastikan bahwa semua input yang wajib diisi sudah sesuai dengan aturan, seperti nama penerima, nama pengirim, bidang, NID, dan tanggal permintaan. Validasi dilakukan untuk memastikan nilai-nilai tersebut bertipe string dan berada dalam batas panjang karakter tertentu, sedangkan deskripsi permintaan bersifat opsional.

Setelah data berhasil divalidasi, data tersebut akan disimpan ke dalam *database* menggunakan model *Request*. Kemudian, pengguna akan diarahkan kembali ke halaman permintaan dengan membawa pesan sukses dan data permintaan yang baru saja dibuat disimpan ke dalam *session*.

```
public function generatePdf($id)
{
    $request = RequestModel::findOrFail($id);

    $pdf = Pdf::loadView('request_pdf', compact('request'));
    return $pdf->download('request_' . $request->RequestID . '.pdf');
}
```

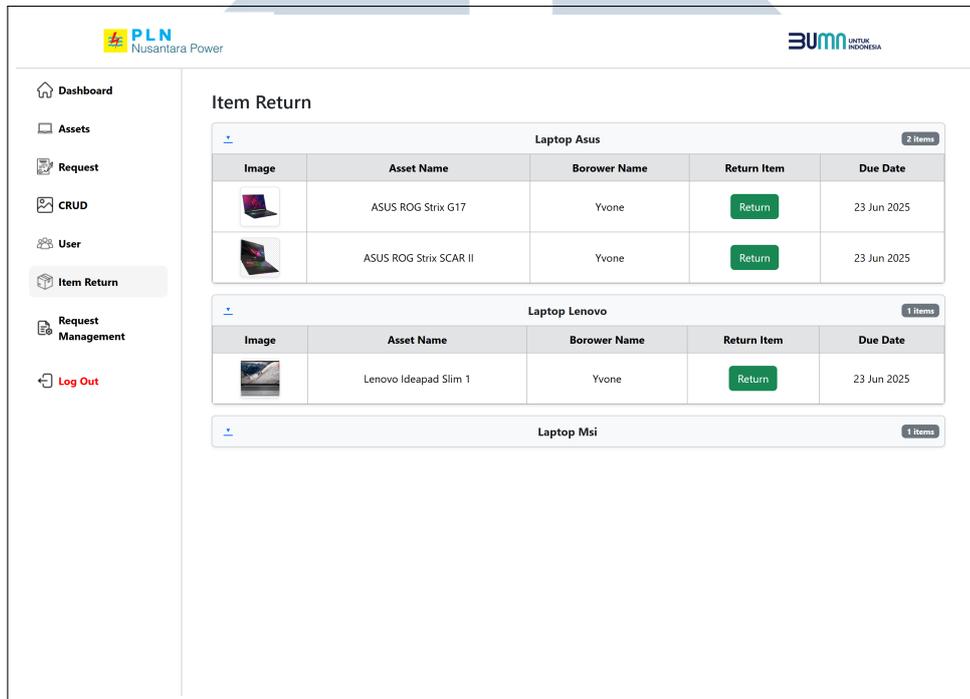
Gambar 3.29. Potongan kode Function generatePdf

Pada Gambar 3.29 menunjukkan bahwa Potongan kode tersebut adalah fungsi *generatePdf* yang digunakan untuk menghasilkan file *PDF* dari data permintaan berdasarkan ID tertentu. Pertama, sistem akan mencari data permintaan di *database* menggunakan ID yang diberikan. Jika data ditemukan, maka data tersebut akan dikirim ke view bernama *request\_pdf* yang akan digunakan sebagai template untuk isi *PDF*. Setelah itu, file *PDF* akan dibuat menggunakan *library PDF* dan diunduh secara otomatis oleh pengguna dengan nama file yang berformat *request\_[ID].pdf*, di mana [ID] adalah ID dari data permintaan tersebut.

UMMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

## E Tampilan dan Potongan Kode Halaman Item Return

### E.1 Tampilan Halaman Item Return



Laptop Asus <span>2 Items</span>				
Image	Asset Name	Borrower Name	Return Item	Due Date
	ASUS ROG Strix G17	Yvone	<button>Return</button>	23 Jun 2025
	ASUS ROG Strix SCAR II	Yvone	<button>Return</button>	23 Jun 2025

Laptop Lenovo <span>1 Items</span>				
Image	Asset Name	Borrower Name	Return Item	Due Date
	Lenovo Ideapad Slim 1	Yvone	<button>Return</button>	23 Jun 2025

Laptop Msi <span>1 Items</span>				
---------------------------------	--	--	--	--

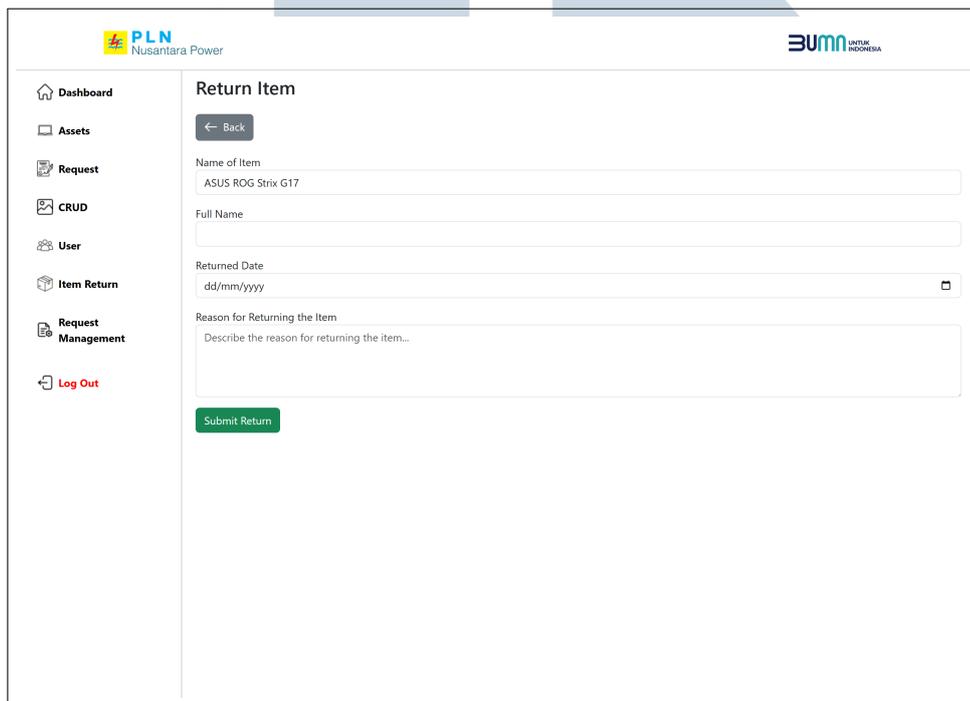
Gambar 3.30. Tampilan Halaman Item Return

Pada halaman *Item Return* di Gambar 3.30, sistem menampilkan daftar barang yang sedang dalam status peminjaman dan perlu dikembalikan oleh pengguna. Halaman ini dirancang untuk memudahkan proses pengelolaan aset dengan menyajikan informasi secara terstruktur dan rapi.

Tampilan dibagi berdasarkan jenis aset, seperti Laptop Asus, Laptop Lenovo, dan Laptop MSI. Masing-masing kelompok aset ditampilkan dalam bentuk *dropdown* yang dapat dibuka dan ditutup sesuai kebutuhan pengguna. Di setiap bagian atas *dropdown*, terdapat keterangan jumlah item yang sedang dipinjam dalam kategori tersebut, sehingga pengguna dapat langsung mengetahui berapa banyak aset yang sedang berada di luar.

Setiap aset yang sedang dipinjam akan ditampilkan beserta informasi lengkap, meliputi nama aset, nama peminjam, serta tanggal jatuh tempo pengembalian. Selain itu, tersedia tombol khusus yang memungkinkan pengguna untuk melakukan proses pengembalian barang melalui sistem.

Fitur ini memberikan kemudahan bagi pengguna dalam memantau aset yang telah dipinjam dan memastikan proses pengembaliannya berjalan dengan tertib dan tepat waktu. Seluruh data yang disajikan membantu mempercepat proses verifikasi dan pencatatan aset dalam lingkungan kerja.



The screenshot shows a web interface for 'Return Item' within the PLN Nusantera Power system. The page has a header with the PLN logo and 'Nusantera Power' on the left, and the BUMI logo with 'UNTUK INDONESIA' on the right. A sidebar on the left contains navigation links: Dashboard, Assets, Request, CRUD, User, Item Return, Request Management, and Log Out. The main content area is titled 'Return Item' and features a 'Back' button. Below this are several input fields: 'Name of Item' (containing 'ASUS ROG Strix G17'), 'Full Name', 'Returned Date' (with a date picker icon and the format 'dd/mm/yyyy'), and a text area for 'Reason for Returning the Item' with the placeholder 'Describe the reason for returning the item...'. A green 'Submit Return' button is positioned at the bottom of the form.

Gambar 3.31. Tampilan Halaman Formulir Item Return

Pada Halaman di Gambar 3.31 tersebut merupakan halaman formulir pengembalian barang dalam sistem manajemen aset. Halaman ini digunakan oleh pengguna untuk mengisi informasi saat akan mengembalikan barang yang sebelumnya telah dipinjam.

Tampilan halaman ini sangat sederhana dan fokus pada pengisian data penting terkait proses pengembalian. Di bagian atas, terdapat nama barang yang akan dikembalikan agar pengguna dapat memastikan barang yang sedang mereka proses. Di bawahnya, pengguna diminta untuk mengisi nama lengkap, tanggal pengembalian, serta alasan mengapa barang tersebut dikembalikan.

Setelah seluruh data diisi dengan lengkap, pengguna dapat menekan tombol "Submit Return" untuk menyelesaikan proses pengembalian barang secara sistematis. Dengan adanya halaman ini, proses pencatatan pengembalian barang menjadi lebih tertib dan terdokumentasi dengan baik dalam sistem.

## E.2 Code Halaman Item Return

```
public function index()
{
    // Ambil semua BorrowItem yang belum dikembalikan
    $borrowedItems = BorrowItem::with(['asset', 'borrow'])
        ->where('is_returned', 0)
        ->get();

    return view('item_return', compact('borrowedItems'));
}
```

Gambar 3.32. Potongan Kode Function index Item Return

Potongan kode tersebut pada Gambar 3.32 merupakan fungsi *index()* yang digunakan untuk menampilkan halaman daftar barang yang masih belum dikembalikan oleh peminjam. Fungsi ini mengambil seluruh data dari tabel *borrowitem* yang memiliki relasi dengan *asset* (data barang) dan *borrow* (data peminjaman), kemudian memfilter hanya item yang memiliki status *is\_returned* bernilai 0, yang berarti barang tersebut masih dalam status dipinjam. Setelah data terkumpul, data tersebut dikirimkan ke view bernama *item\_return* melalui variabel *borrowedItems* untuk ditampilkan kepada pengguna.

UMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

```

public function store(Request $request)
{
    $request->validate([
        'borrow_item_id' => 'required|exists:borrowitem,BorrowItemID',
        'full_name' => 'required|string|max:100',
        'asset_desc' => 'required|string',
        'returned_date' => 'required|date',
    ]);

    DB::transaction(function () use ($request) {
        \App\Models\ItemReturn::create([
            'ReturnID' => \App\Models\ItemReturn::max('ReturnID') + 1,
            'BorrowItemID' => $request->borrow_item_id,
            'FullName' => $request->full_name,
            'ReturnedDescription' => $request->asset_desc,
            'ReturnedDate' => $request->returned_date,
        ]);

        // Update flag is_returned di borrowitem supaya statusnya updated
        $borrowItem = BorrowItem::findOrFail($request->borrow_item_id);
        $borrowItem->is_returned = 1;
        $borrowItem->save();
    });

    return redirect()->route('item.return')->with('success', 'Item has been returned.');
```

Gambar 3.33. Potongan Kode Function store Item Return

Potongan kode pada Gambar 3.33 merupakan fungsi *store()* yang digunakan untuk menyimpan data pengembalian barang yang sebelumnya dipinjam.

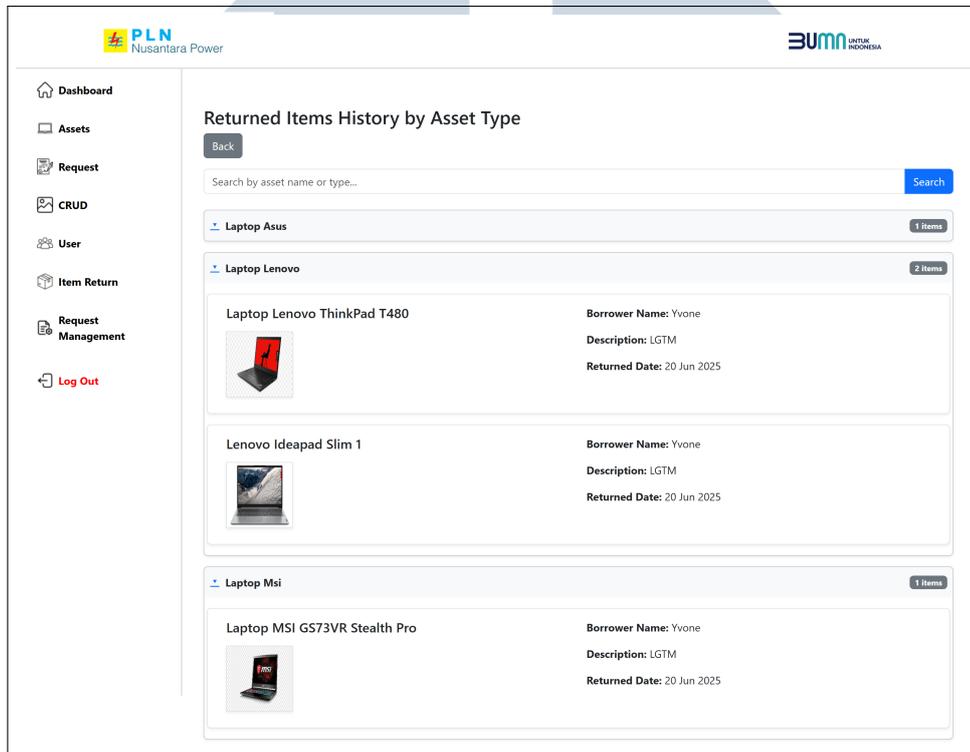
Fungsi ini memvalidasi data yang dikirimkan dari form pengembalian, seperti ID barang yang dipinjam, nama lengkap pengembali, deskripsi kondisi barang saat dikembalikan, dan tanggal pengembalian. Validasi ini memastikan bahwa semua data yang dibutuhkan sudah benar dan lengkap.

Setelah validasi berhasil, proses penyimpanan dilakukan di dalam transaksi database untuk menjaga konsistensi data. Di dalam transaksi tersebut, sistem akan membuat data baru di tabel *ItemReturn* dengan *ReturnID* yang ditentukan berdasarkan nilai tertinggi sebelumnya ditambah satu. Data yang disimpan meliputi ID barang yang dikembalikan, nama pengembali, deskripsi, dan tanggal pengembalian.

Setelah itu, status barang di tabel *borrowitem* akan diperbarui dengan mengubah nilai *is\_returned* menjadi 1, yang menandakan bahwa barang tersebut sudah dikembalikan. Jika seluruh proses berhasil, pengguna akan diarahkan kembali ke halaman daftar pengembalian barang dengan notifikasi bahwa pengembalian berhasil dilakukan.

## F Tampilan dan Potongan Kode Halaman Returned Items History

### F.1 Tampilan Halaman Returned Items History



Gambar 3.34. Tampilan Halaman Returned Items History

Halaman pada Gambar 3.34 merupakan tampilan riwayat pengembalian barang yang telah dilakukan oleh pengguna. Halaman ini digunakan untuk menampilkan daftar aset yang sudah dikembalikan berdasarkan jenis asetnya.

Setiap tipe aset dikelompokkan dan ditampilkan dalam bentuk *dropdown* yang dapat diperluas atau disembunyikan. Pada setiap bagian *dropdown*, pengguna dapat melihat jumlah *item* yang termasuk dalam kategori aset tersebut, sehingga memudahkan dalam melakukan pencarian atau pelacakan.

Untuk setiap barang yang telah dikembalikan, ditampilkan informasi berupa nama aset, gambar aset, nama peminjam, alasan atau deskripsi pengembalian, serta tanggal kapan barang tersebut dikembalikan. Dengan informasi ini, pengguna dapat mengetahui histori pengembalian barang secara lengkap dan terstruktur.

## F.2 Code Halaman Returned Items History

```
public function history()
{
    $returnedItems = \App\Models\ItemReturn::with(['borrowItem.asset', 'borrowItem.borrow'])->get();
    return view('item_return_history', compact('returnedItems'));
}
```

Gambar 3.35. Potongan Kode Returned Items History

Potongan kode pada Gambar 3.35 adalah fungsi *history()* yang digunakan untuk menampilkan halaman riwayat pengembalian barang. Fungsi ini mengambil semua data dari model *ItemReturn*, yaitu data barang-barang yang telah dikembalikan. Saat pengambilan data, sistem juga memuat relasi ke tabel *borrowitem*, lalu dari sana ikut memuat relasi ke tabel *asset* (data barang) dan *borrow* (data peminjamannya).

Seluruh data yang telah dikumpulkan ini kemudian dikirim ke *view* bernama *item\_return\_history* dalam variabel *returnedItems*, yang nantinya akan ditampilkan sebagai daftar histori pengembalian barang secara lengkap.

## 3.5 Pengujian Sistem

### 3.5.1 Pengujian *Black Box Testing* Pada Sistem

#### A Fitur *Login*

Tabel 3.2 digunakan untuk merangkung hasil pengujian black box untuk fitur login yang dikerjakan.

Tabel 3.2. Pengujian Blackbox *Login*

No.	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Hasil
1	<i>Username</i> atau <i>password</i> tidak diisi tetapi klik <i>button login</i>	<i>Username</i> atau <i>password</i> kosong	Harap masukkan isi pada kolom berikut	<i>Please fill out this field</i>	Berhasil

No.	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Hasil
2	Mengetikkan username dengan role Admin, tetapi tidak memasukkan password atau password salah	<i>username</i> terisi tetapi <i>password</i> kosong atau salah	<i>Wrong Password</i>	<i>Wrong Password</i>	Berhasil
3	Mengetikkan username dengan role Super Admin, tetapi tidak memasukkan password	<i>username</i> terisi tetapi <i>password</i> kosong atau salah	<i>Wrong Password</i> dan muncul <i>button</i> untuk mengganti password	<i>Button</i> untuk mengubah password muncul	Berhasil

## B Fitur Asset Lending

Tabel 3.3 digunakan untuk merangkum hasil pengujian black box untuk fitur yang dikerjakan.

Tabel 3.3. Pengujian Blackbox Asset Lending

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Hasil
1	Pengguna mencoba meminjam tanpa memasukkan aset ke dalam <i>shopping cart</i>	Klik tombol <i>View Cart</i> saat keranjang kosong	Tampil pesan "Your cart is empty"	Tampil pesan "Your cart is empty"	Berhasil

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Hasil
2	Pengguna mengakses halaman <i>View History</i> untuk melihat riwayat aset yang pernah dipinjam	Klik tombol <i>View History</i>	Tampil daftar riwayat peminjaman untuk aset tersebut	Tampil daftar riwayat peminjaman untuk aset tersebut	Berhasil
3	Pengguna menambahkan aset ke dalam keranjang dan melanjutkan ke proses peminjaman	Pilih aset → tambahkan ke <i>shopping cart</i> → klik <i>Confirm Lending</i> → isi form peminjam	Tampil halaman konfirmasi dan form input data peminjam	Tampil halaman konfirmasi dan form input data peminjam	Berhasil

### C Fitur *Borrowed Items History*

Tabel 3.4 digunakan untuk merangkum hasil pengujian black box untuk fitur yang dikerjakan.

Tabel 3.4. Pengujian *Blackbox Borrowed Items History*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Hasil
1	Pengguna mengklik dropdown pada tipe aset tertentu untuk menampilkan seluruh aset yang tersedia	Klik dropdown pada tipe aset (misal: Laptop)	Seluruh aset pada tipe tersebut tampil, termasuk yang pernah dipinjam	Seluruh aset tampil dengan benar	Berhasil

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Hasil
2	Pengguna menggunakan search bar untuk mencari aset berdasarkan nama barang atau tipe aset	Ketik kata kunci pada search bar (misal: "Asus" atau "Laptop")	Aset yang sesuai dengan nama atau tipe aset muncul	Aset yang sesuai berhasil ditampilkan	Berhasil

#### D Fitur *Returned Items History*

Tabel 3.5 digunakan untuk merangkung hasil pengujian black box untuk fitur yang dikerjakan.

Tabel 3.5. Pengujian *Blackbox Returned Items History*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Hasil
1	Pengguna mengklik dropdown pada tipe aset tertentu untuk menampilkan seluruh aset yang tersedia	Klik dropdown pada tipe aset (misal: Laptop)	Seluruh aset pada tipe tersebut tampil, termasuk yang pernah dipinjam	Seluruh aset tampil dengan benar	Berhasil
2	Pengguna menggunakan search bar untuk mencari aset berdasarkan nama barang atau tipe aset	Ketik kata kunci pada search bar (misal: "Asus" atau "Laptop")	Aset yang sesuai dengan nama atau tipe aset muncul	Aset yang sesuai berhasil ditampilkan	Berhasil

## E Fitur *Request Form*

Tabel 3.6 digunakan untuk merangkung hasil pengujian black box untuk fitur yang dikerjakan.

Tabel 3.6. Pengujian *Blackbox* Fitur *Request*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Hasil
1	Pengguna mengirim form permintaan barang tanpa mengisi salah satu kolom isian (form tidak lengkap)	Biarkan salah satu field kosong (misal: NID atau RequestDesc) lalu klik tombol submit	Form tidak dapat disubmit, muncul validasi atau pesan kesalahan	Form tidak dapat disubmit, muncul pesan error bawaan browser	Berhasil
2	Pengguna mengklik tombol untuk mengunduh permintaan dalam bentuk PDF	Klik tombol <i>Download PDF</i> setelah request berhasil disimpan	File PDF berhasil diunduh	File PDF berhasil diunduh sesuai isi form	Berhasil

## F Fitur *Return Item*

Tabel 3.7 digunakan untuk merangkung hasil pengujian black box untuk fitur yang dikerjakan.

Tabel 3.7. Pengujian *Blackbox* Fitur *Item Return*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Hasil
1	Sistem menampilkan daftar item yang sedang dipinjam berdasarkan tipe aset, masing-masing dalam dropdown	Klik dropdown pada setiap tipe aset (misal: Laptop, Printer)	Isi aset dari tiap tipe tampil di bawahnya, disertai tombol <i>Return</i>	Daftar aset tampil dan tombol <i>Return</i> muncul untuk tiap item	Berhasil
2	Pengguna mengklik tombol <i>Return</i> untuk memulai pengembalian aset tertentu	Klik tombol <i>Return</i> pada salah satu aset	Form pengembalian tampil (Return Item Form)	Form pengembalian tampil dengan field yang harus diisi	Berhasil
3	Pengguna mencoba submit form Return Item tanpa mengisi semua field wajib	Biarkan salah satu kolom (misal: Return Date atau Reason) kosong lalu klik submit	Form tidak dapat disubmit, dan muncul pesan "Please fill out this field"	Muncul pesan "Please fill out this field"	Berhasil

### 3.6 Kendala dan Solusi yang Ditemukan

#### 3.6.1 Kendala yang Ditemukan

Selama proses pengembangan sistem, ditemukan beberapa kendala teknis yang mempengaruhi kelancaran implementasi yaitu :

1. Pertama, sering terjadi kesalahan pada layanan XAMPP, khususnya pada komponen MySQL, dengan pesan error "MySQL shutdown unexpectedly". Kendala ini menyebabkan database tidak dapat dijalankan secara normal dan menghambat proses pengujian sistem.
2. Kedua, keterbatasan teknis juga ditemukan saat menggunakan pustaka *dompdf* untuk kebutuhan pencetakan dokumen ke dalam format *PDF*. *Dompdf* tidak sepenuhnya mendukung fitur-fitur layout modern berbasis CSS, seperti penggunaan *flexbox*, *grid*, dan beberapa properti dari *positioning*. Akibatnya, struktur tampilan yang dihasilkan dalam file *PDF* sering kali tidak sesuai dengan desain semula yang diterapkan pada tampilan *website*.

### 3.6.2 Solusi yang Ditemukan

Setelah menemukan kendala yang di hadapi selama proses implementasi, ditemukan juga solusi yang dapat menyelesaikan kendala yang dihadapi sebelumnya, yaitu :

1. Untuk mengatasi permasalahan pada MySQL, solusi yang diambil adalah dengan memanfaatkan data cadangan (*backup*) yang tersimpan di dalam direktori '*backup*' pada folder instalasi MySQL. File penting seperti '*ibdata1*' diduplikasi dari folder *backup*, kemudian dipindahkan ke direktori utama MySQL untuk menggantikan file yang rusak. Langkah ini berhasil memulihkan database dan memungkinkan XAMPP untuk kembali berjalan normal.
2. Sementara itu, untuk mengatasi keterbatasan *dompdf* dalam menampilkan layout modern, pendekatan yang digunakan adalah dengan menyusun kembali struktur HTML menggunakan elemen *table*. *Layout* berbasis tabel dipilih karena lebih kompatibel dengan engine rendering *dompdf*, sehingga hasil keluaran *PDF* dapat tetap terbaca dan terstruktur dengan baik.