

BAB III

PELAKSANAAN KERJA

3.1 Bidang Kerja

Praktikan bekerja di divisi pengembangan sebagai programmer bagian Front-End dan Back-End. Divisi ini bertanggung jawab untuk memasukkan hasil analisis ke dalam kode program dan memastikan bahwa aplikasi atau website berjalan dan bekerja dengan baik. Divisi pengembangan juga melakukan pengujian pada aplikasi atau website untuk menemukan dan memperbaiki *bug* dan *error*.

Setelah menjadi karyawan magang di divisi pengembangan, mentor memberi praktikan proyek untuk membuat aplikasi pengelola data barang logistik event berbasis web yang menggunakan *framework Next.js*, *Tailwind CSS*, dan *Supabase* sebagai *database*. Aplikasi berbasis web tersebut diminta untuk memiliki fitur tertentu. Ini termasuk kemampuan untuk menyimpan data barang yang dibeli atau disewa, memantau detail produk, dan menemukan data melalui *kode QR*.

Praktikan diminta untuk mempelajari *framework* dan platform pendukung perancangan aplikasi yang akan digunakan untuk menyelesaikan proyek pengelolaan data barang logistik berbasis web.

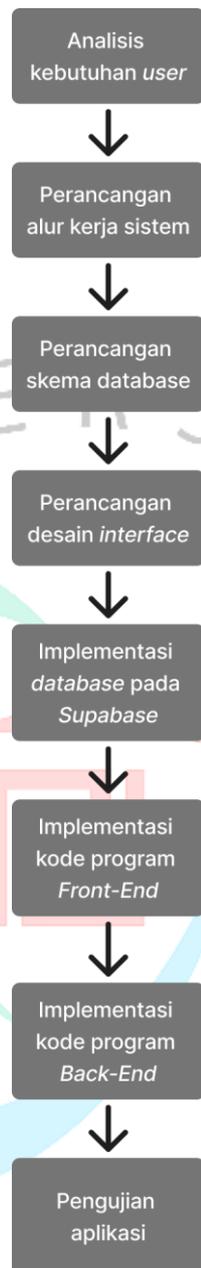
3.2 Pelaksanaan Kerja

Agar proyek aplikasi berbasis website berhasil, ada beberapa tahap yang harus dilakukan secara terstruktur dan sistematis. Praktikan tampilkan tahapan pelaksanaan kerja secara detail melalui diagram.

Tabel 3.1 Tahap Pelaksanaan Kerja

Kegiatan	Jadwal Kegiatan (Per Minggu)											
	Agustus				September				Oktober			
	1	2	3	4	1	2	3	4	1	2	3	4
Analisis Kebutuhan <i>User</i>	■											
Perancangan Alur Kerja Sistem <i>Website</i> Aplikasi		■										
Perancangan Skema <i>Database</i>			■	■								
Perancangan Desain UI <i>Website</i> Aplikasi					■							
Implementasi <i>Database</i> pada <i>Supabase</i>						■						
Implementasi Kode Program <i>Front-End</i>							■					
Implementasi Kode Program <i>Back-End</i>								■	■			
Pengujian <i>website</i> aplikasi									■			
Pemaparan hasil kerja										■		

Berdasarkan tabel 3.1 di atas menunjukkan proses pelaksanaan kerja profesi praktikan di perusahaan Brainlens rekayasa konsultan dalam waktu 3 bulan. Dalam pengembangan *website* aplikasi ini, praktikan menggunakan beberapa tahapan agar sistem dapat berjalan dengan baik saat digunakan. Tahapan tersebut yakni fase analisis, fase perancangan, fase implementasi, dan fase pengujian.



Gambar 3.1 Tahapan Perancangan Sistem Aplikasi

Pada tahap perancangan sistem aplikasi di atas, terlihat seluruh tahapan perancangan aplikasi berbasis website. Untuk menyelesaikan proyek tersebut, para praktikan bekerja sama dengan rekan kerja mereka untuk membagi tugas dan membahas bagian apa yang harus mereka lakukan. Hasil diskusi mereka menentukan bahwa praktikan akan bekerja pada bagian analisis, perancangan database, dan implementasi kode program *Back-End*.

3.2.1 Analisis Kebutuhan User

Pada tahap ini, praktikan menganalisis fungsi dan fitur aplikasi yang akan dibuat agar sesuai dengan permintaan pengguna. Hasil analisis praktikan tentang pembuatan aplikasi adalah sebagai berikut.

1) Identifikasi *Stakeholder*

Aplikasi pengelola data barang logistik ini akan digunakan oleh panitia *event* khususnya di divisi logistik, tim logistik akan menggunakan aplikasi ini sebagai tempat penyimpanan data barang yang akan dibeli ataupun sewa serta bisa mengelolanya, tujuannya adalah untuk memudahkan tim logistik menyimpan dan mengelola data barang dengan lebih efisien.

2) Spesifikasi Kebutuhan User

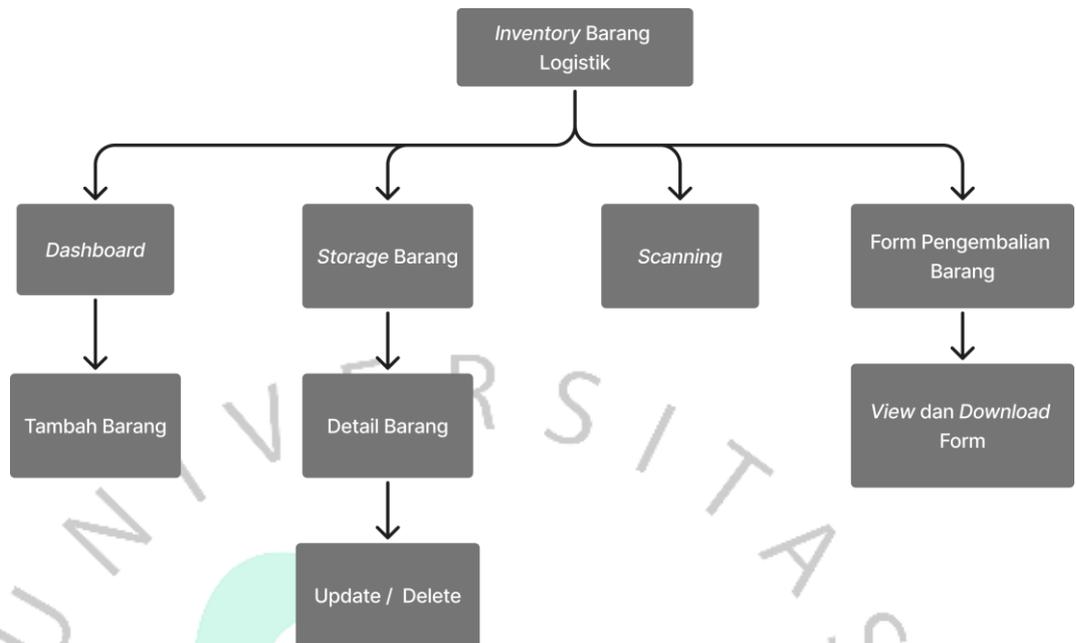
User diharuskan untuk dapat menambahkan, mengedit, dan menghapus data barang, serta dapat melihat informasi barang secara rinci. User juga dapat melakukan *scanning* pada QR Code yang telah dibuat dan ditempelkan pada barang untuk mempermudah pencarian barang di aplikasi.

3) Spesifikasi Kebutuhan User Interface

Pada saat membangun aplikasi berbasis *website*, praktikan diharapkan dapat membuat antarmuka yang mudah digunakan dan *mobile friendly* untuk memudahkan user dalam menggunakan aplikasi tersebut di lapangan.

3.2.2 Perancangan Sistem

Setelah melakukan analisis kebutuhan untuk pengembangan aplikasi berbasis *website*, praktikan merancang fitur serta sistem dari aplikasi tersebut sesuai dengan hasil analisis yang telah dilakukan. Perancangan ini akan menjadi dasar untuk pengembangan sistem berikutnya.

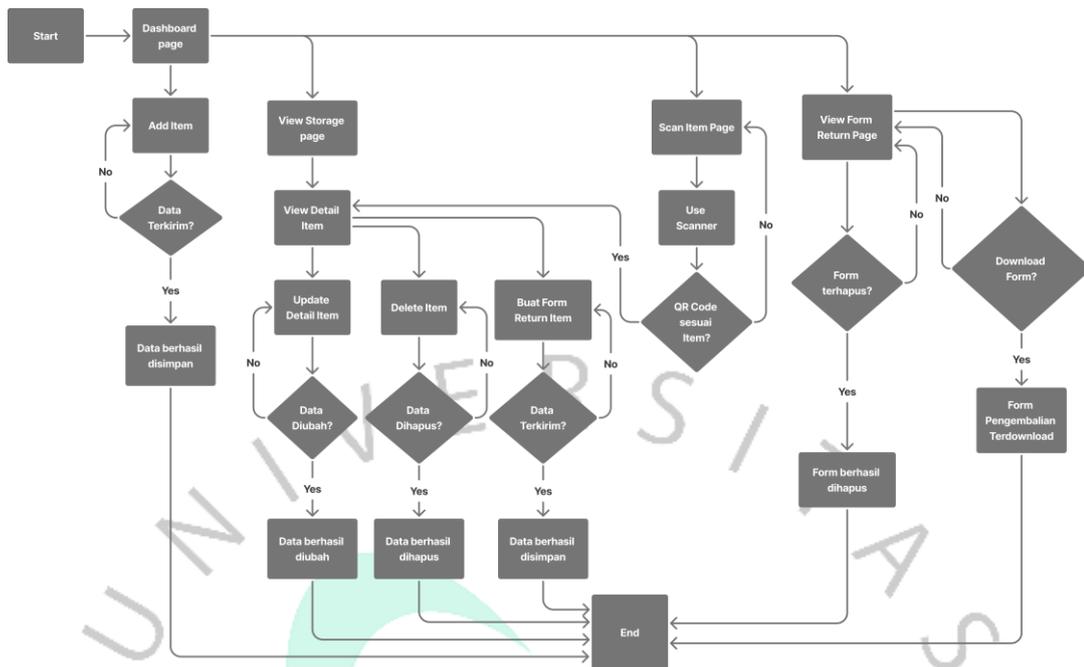


Gambar 3.2 Struktur Aplikasi

Gambar di atas merupakan gambaran struktur aplikasi yang terdapat pada aplikasi pengelolaan data barang logistik. Pada struktur aplikasi ini, terdapat beberapa halaman yang dirancang untuk dikembangkan menjadi desain *flowchart*, desain *Mockup*, dan perancangan *database*. *Flowchart* adalah diagram yang digunakan untuk menggambarkan alur atau proses dari suatu aplikasi secara grafis. Desain *Mockup* merupakan representasi visual yang menampilkan tampilan dan fungsi utama antarmuka dari aplikasi tersebut. Perancangan *database* adalah proses untuk menetapkan struktur, tabel, hubungan, dan elemen data lainnya untuk memenuhi kebutuhan aplikasi tersebut.

3.2.2.1 *Flowchart*

Setelah praktikan menganalisis semua kebutuhan untuk membuat aplikasi ini, langkah selanjutnya dalam proses perancangan adalah membuat *flowchart* untuk menunjukkan alur atau proses kerja aplikasi secara visual.



Gambar 3.3 Desain *Flowchart* Aplikasi

Gambar di atas merupakan desain *flowchart* aplikasi yang akan digunakan untuk pengembangan aplikasi. Sebelum tahapan *Mockup* dan implementasi, *flowchart* berfungsi sebagai dasar alur aplikasi yang akan bekerja. Berdasarkan *flowchart* tersebut, terdapat beberapa aktivitas yang dapat dikerjakan oleh *user*, ketika *user* ingin menambah sebuah barang maka *user* akan diarahkan untuk mengisi data terkait barang tersebut, apabila seluruh data telah diisi dengan lengkap dan benar, maka data akan dikirim ke *database* lalu ditampilkan ke halaman *storage*. Selanjutnya ketika *user* ingin mengubah data barang yang telah tersimpan, *user* akan memilih data yang akan diubah pada halaman *storage*, lalu diarahkan ke halaman detail barang, pada halaman tersebut, *user* dapat mengubah data barang ataupun menghapus data barang yang dipilih sesuai keinginannya. Lalu ketika *user* ingin melakukan pencarian data barang melalui fitur *scan item*, pada halaman *scan item* *user* diharuskan untuk menyetujui perizinan kamera pada perangkat pribadi, lalu melakukan *scanning* pada *QR Code* yang ada pada barang yang ingin diketahui datanya. Ketika berhasil melakukan *scanning*, *user* akan diarahkan ke halaman detail barang

tersebut. Terakhir ketika *user* ingin membuat formulir pengembalian barang sewa, pada halaman detail barang ketika *user* menekan tombol form pengembalian, maka *user* akan diarahkan ke halaman pengisian data untuk mengisi data tanggal pengembalian barang, lalu apabila data telah diisi dengan lengkap dan benar, maka data akan dikirim ke *database*. Ketika *user* ingin melihat hasil pembuatan formulir pengembalian barang, pada halaman form pengembalian, *user* dapat melihat data barang yang sebelumnya telah dikirim, lalu *user* dapat mengunduh ataupun menghapus data pada form tersebut.

3.2.2.2 Sketsa Database

Setelah melakukan analisis kebutuhan dan desain flowchart, praktikan melanjutkan ke fase perancangan dengan membuat sketsa database. Ini memberikan gambaran awal tentang struktur data sebelum memulai implementasi database.

Tabel 3.1 Sketsa Tabel Storage

Nama	Tipe Data	Length	Keterangan
id_barang	int	-	Klasifikasi data barang
nama_pic	varchar	50	Nama penanggung jawab barang
divisi	varchar	50	Divisi penanggung jawab
no_contact	varchar	20	Nomor kontak penanggung jawab
kategori_barang	varchar	50	Kategori barang
barang	varchar	100	Nama barang
status_barang	varchar	15	Status barang
tgl_beli	date	-	Tanggal pembelian/penyewaan barang
vendor	varchar	50	Nama vendor sewa
catatan	text	100	Catatan untuk barang
jumlah_barang	int	10	Jumlah barang
foto_url	text	-	Menampilkan gambar dari bucket di <i>supabase</i>

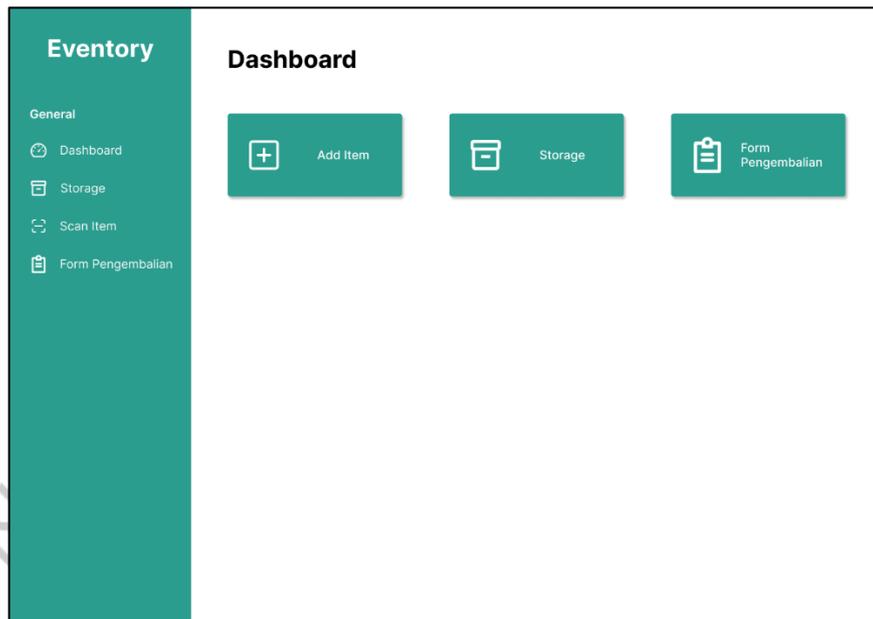
Tabel 3.2 Sketsa Tabel *Return*

Nama	Tipe Data	Length	Keterangan
id_barang	int	-	Menyimpan data barang sewa
nama_pic	varchar	50	Nama penanggung jawab
barang_sewa	varchar	100	Nama barang sewa
tgl_kembali	date	-	Tanggal pengembalian barang
vendor	varchar	50	Nama vendor terkait
no_contact	varchar	20	Nomor kontak penanggung jawab

Kedua tabel di atas merupakan gambaran awal serta struktur dari tabel yang akan dibuat ke dalam *database*. Pada *database*, terdapat dua tabel yaitu tabel *Storage* dan *Return*, tabel *Storage* bertujuan untuk menyimpan data-data yang masuk ketika *user* melakukan *input* data barang, sedangkan tabel *Return* bertujuan sebagai tempat penyimpanan data barang sementara untuk keperluan formulir pengembalian barang.

3.2.2.3 Desain Mockup

Setelah menyelesaikan perancangan *database*, praktikan melakukan desain mockup untuk menunjukkan alur kerja dan tampilan aplikasi. Ini membantu mereka memahami gambaran sistem sebelum tahap implementasi kode program dimulai.



Gambar 3.4 Mockup *dashboard*

Gambar di atas adalah desain dari halaman *dashboard*, pada tampilan halaman *dashboard* ini akan menampilkan *button* navigasi ke halaman *add item*, *storage*, dan form pengembalian. Halaman *dashboard* juga merupakan halaman awal pada saat user membuka aplikasi berbasis website tersebut.

Eventory

General

- Dashboard
- Storage
- Scan Item
- Form Pengembalian

Add Item

Data Pengaju

Nama PIC: Divisi: Nomor Telepon/WhatsApp:

Data Barang

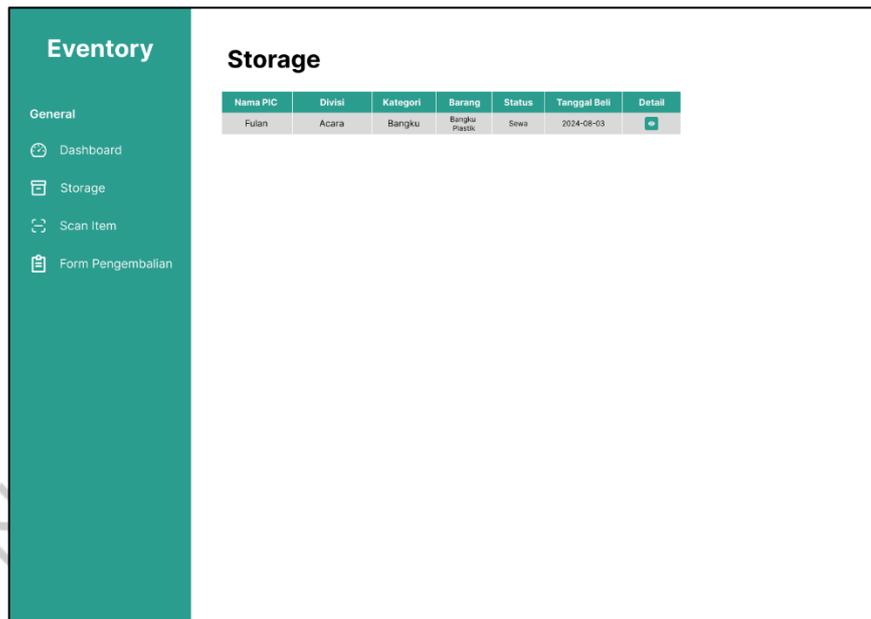
Kategori Barang: Nama Barang: Status Barang: Tanggal Pengembalian Barang:

Jumlah Barang: Nama Vendor: Catatan Barang:

Foto Bukti Barang

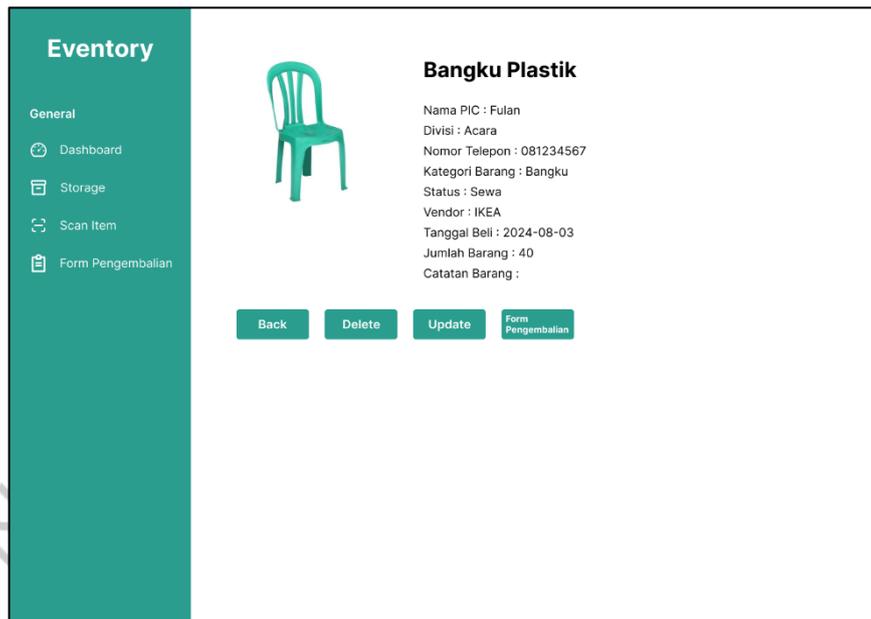
Gambar 3.5 Mockup Add Item

Gambar di atas merupakan desain dari halaman *add item* setelah user menekan *button* untuk menuju ke halaman *add item*. Pada halaman ini, terdapat beberapa data yang perlu diisi oleh user terkait informasi barang seperti data pengaju dan data barang yang nantinya akan disimpan di dalam *storage* nanti.



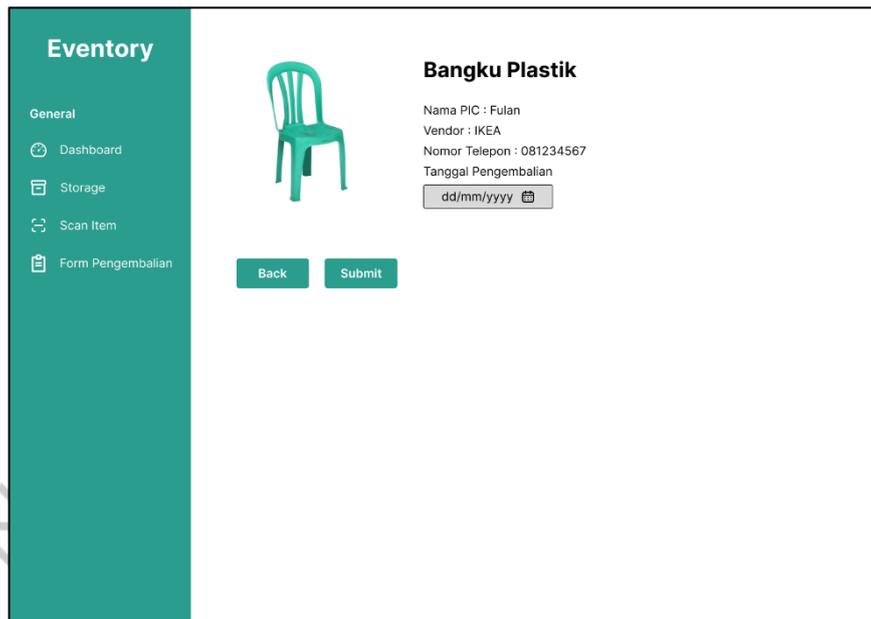
Gambar 3.6 Mockup Storage Utama

Gambar di atas merupakan desain mockup dari tampilan halaman utama *storage*, pada tampilan halaman utama *storage*, terdapat tabel yang pada masing-masing kolom berisikan beberapa informasi mengenai data pengaju dan data barang, seperti nama pengaju atau PIC barang, divisi, kategori barang, nama barang, status barang, tanggal beli atau sewa, dan detail. Pada kolom detail terdapat *button* untuk mengarahkan user ke halaman detail dari data barang yang dipilih. Halaman utama *storage* ini bertujuan untuk menampilkan data-data yang berhasil tersimpan setelah *user* menambahkan data barang.



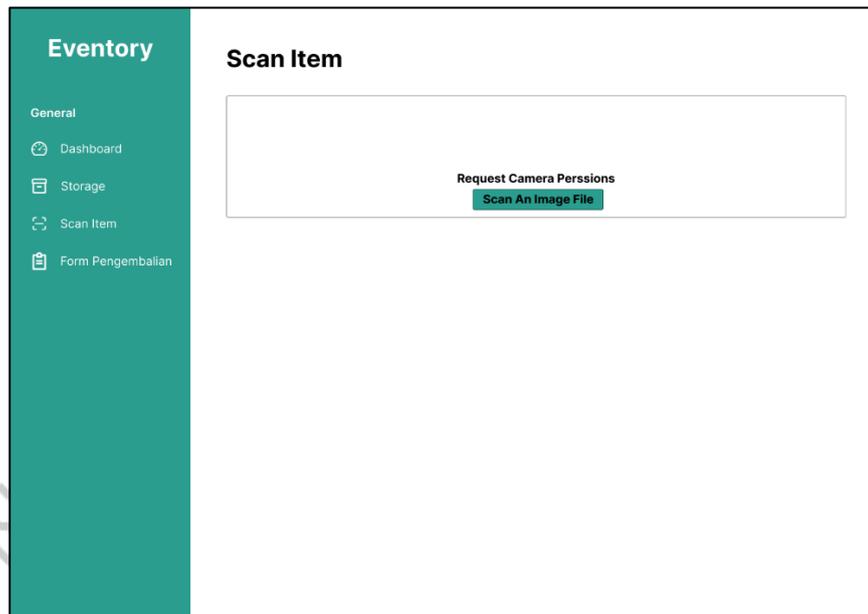
Gambar 3.7 Mockup Detail Storage

Gambar di atas merupakan desain mockup dari tampilan halaman detail *storage*, pada desain tampilan tersebut terdapat seluruh informasi terkait barang yang telah disimpan. Pada halaman tersebut juga terdapat fitur untuk menghapus dan mengubah data barang tersebut, selain itu terdapat *button* form pengembalian untuk mengarahkan *user* ke halaman pengisian form pengembalian.



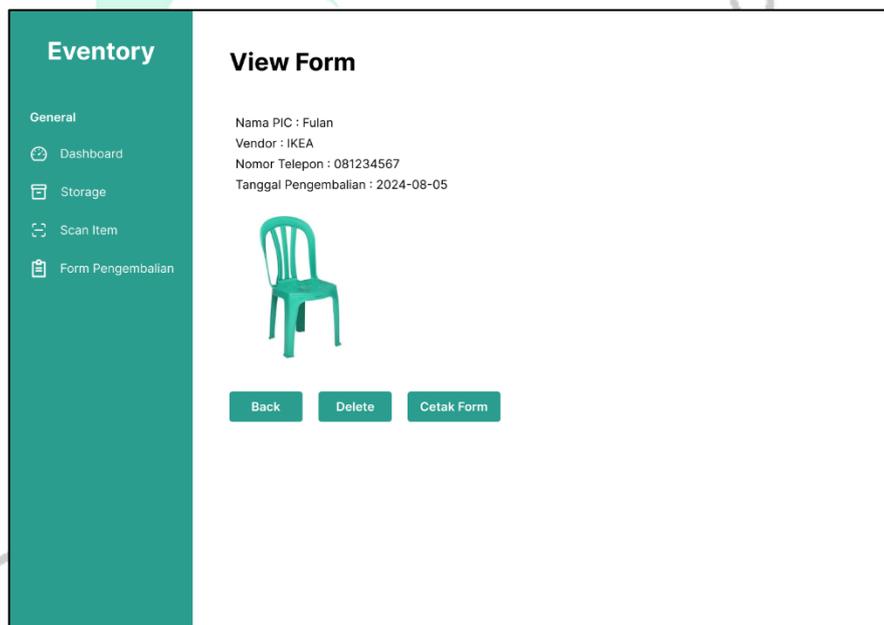
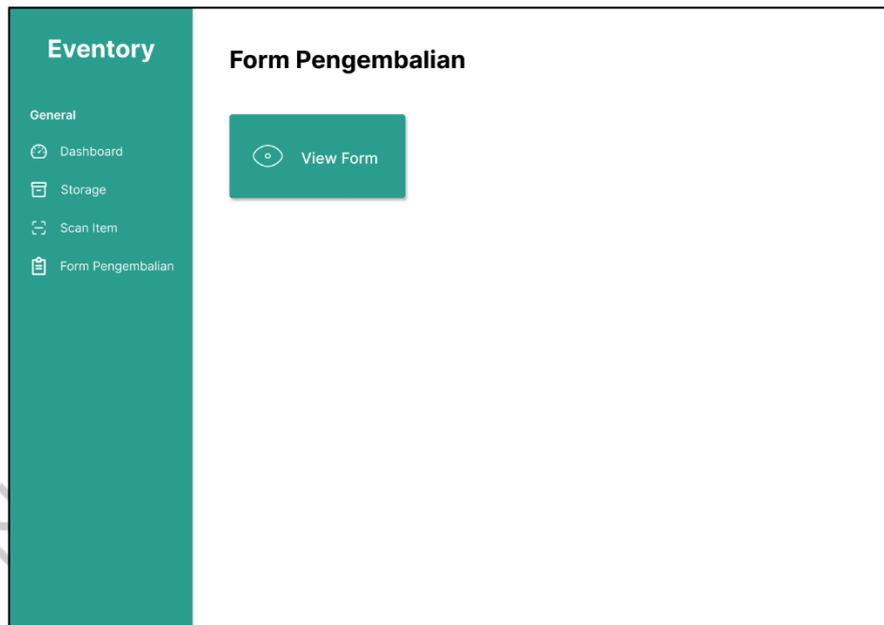
Gambar 3.8 Mockup Input Form Pengembalian

Gambar di atas merupakan desain tampilan dari halaman pengisian form pengembalian, ketika *user* menekan *button* form pengembalian pada halaman detail *storage*, *user* akan diarahkan ke halaman ini untuk mengisi tanggal pengembalian yang telah ditentukan pada saat peminjaman barang sebagai catatan lalu disimpan ke dalam *database* pada tabel *return*.



Gambar 3.9 Mockup Scan Item

Gambar di atas merupakan tampilan halaman *scan item*, halaman tersebut bertujuan untuk melakukan *scanning* pada *QR Code* yang ada pada barang, ketika berhasil melakukan *scanning*, sistem akan mengarahkan *user* ke halaman detail barang yang sesuai dengan *QR Code*.

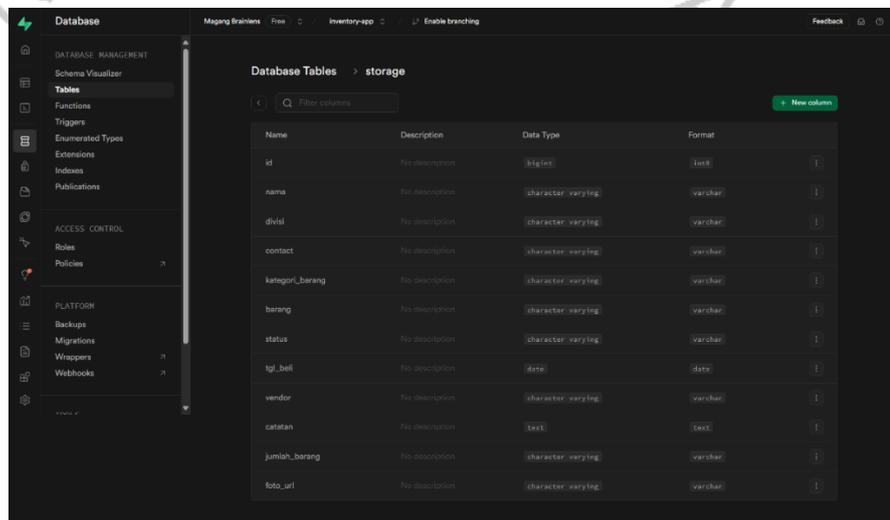


Gambar 3.10 Mockup Form Pengembalian

Gambar di atas merupakan mockup dari halaman *form* pengembalian, pada halaman tersebut terdapat informasi mengenai barang sewa yang akan dibuatkan menjadi sebuah formulir pengembalian barang, fitur untuk menghapus data pada formulir, dan fitur untuk mencetak data tersebut ke dalam format *pdf*.

3.2.2.4 Realisasi Database

Pada tahap perancangan ini, setelah praktikan melakukan desain mockup dari tampilan halaman yang nanti akan dikembangkan ke dalam aplikasi berbasis website ini, selanjutnya praktikan melakukan realisasi *database* ke dalam platform *database Supabase*. Platform ini biasa dipakai oleh perusahaan untuk membuat sebuah *website* ataupun aplikasi karena mendukung *framework React* yaitu *Next.js*. Supabase ini merupakan alternatif dari *Firestore* yang menggunakan sistem manajemen basis data PostgreSQL.



The screenshot shows the Supabase database management interface. On the left, there is a sidebar with navigation options: DATABASE MANAGEMENT (Schema Visualizer, Tables, Functions, Triggers, Enumerated Types, Extensions, Indexes, Publications), ACCESS CONTROL (Roles, Policies), and PLATFORM (Backups, Migrations, Wrappers, Webhooks). The main area displays the 'Database Tables > storage' table structure. A search bar 'Filter columns' and a '+ New column' button are at the top right of the table view.

Name	Description	Data Type	Format
id	No description	bigint	int64
nama	No description	character varying	varchar
divisi	No description	character varying	varchar
contact	No description	character varying	varchar
kategori_barang	No description	character varying	varchar
barang	No description	character varying	varchar
status	No description	character varying	varchar
tgl_beli	No description	date	date
vendor	No description	character varying	varchar
catatan	No description	text	text
jumlah_barang	No description	character varying	varchar
foto_url	No description	character varying	varchar

Gambar 3.11 Realisasi Database Tabel Storage

The screenshot shows a database management interface with a sidebar on the left containing categories like 'DATABASE MANAGEMENT', 'ACCESS CONTROL', and 'PLATFORM'. The main area displays 'Database Tables > return' with a table listing columns and their data types.

Name	Description	Data Type	Format
id	No description	uuid	uuid
nama	No description	character_varying	varchar
barang	No description	character_varying	varchar
tgl_kembali	No description	date	date
vendor	No description	character_varying	varchar
contact	No description	character_varying	varchar

Gambar 3.12 Realisasi Database Tabel Return

Gambar di atas merupakan realisasi tabel *database* yang dibuat praktikan dengan sistem manajemen *PostgreSQL*. Pada umumnya sistem manajemen *PostgreSQL* lebih cepat dan efisien untuk jumlah data yang cukup besar dan *query* yang rumit, hal itu yang membuatnya ideal untuk aplikasi yang berskala besar, dibutuhkan cepat dan ketahanan performa yang tinggi. (Ardian *et al.*, 2020). Selain itu, *PostgreSQL* mendukung pembuatan fungsi, prosedur tersimpan, dan pemacu (*trigger*), yang memungkinkan sebagian logika bisnis diproses langsung di server. Ini membuat aplikasi yang dihubungkan dengan *database* lebih cepat, terutama untuk aplikasi *thin-client* yang membutuhkan reaksi cepat. (Siti Munawaroh., 2005).

Pada gambar di atas terdapat dua tabel *database* yang telah dibuat oleh praktikan dengan fungsi yang berbeda masing-masing, tabel *storage* digunakan untuk menyimpan data atau informasi terkait barang beli ataupun sewa, sedangkan untuk tabel *return* digunakan untuk menyimpan data hasil *input* ketika *user* ingin melakukan pembuatan form pengembalian barang sewa.

3.2.3 Implementasi

Setelah tahap perancangan selesai, tahap implmentasi adalah tahap melaksanakan rancangan. Pada titik ini, praktikan akan memberikan penjelasan tentang framework yang digunakan, spesifikasi kebutuhan perangkat lunak dan perangkat keras, dan tampilan hasil realisasi aplikasi.

3.2.3.1 Spesifikasi Kebutuhan Perangkat Lunak

Praktikan menggunakan beberapa perangkat lunak untuk membantu mereka membangun aplikasi berbasis *website* ini. Berikut adalah spesifikasi perangkat lunak yang dibutuhkan untuk membangun aplikasi ini.

Tabel 3.3 Spesifikasi Kebutuhan Perangkat Lunak

No	Perangkat Lunak	Kebutuhan Perangkat Lunak
1	<i>Operation System</i>	Windows 11
2	<i>IDE</i>	Visual Studio Code
3	<i>Front-End Framework</i>	Next.js
4	<i>Back-End Framework</i>	Next.js
5	<i>Database</i>	Supabase
6	<i>Scripting Language</i>	Javascript
7	<i>CSS Framework</i>	Tailwind

3.2.3.2 Spesifikasi Kebutuhan Perangkat Keras

Praktikan menggunakan beberapa perangkat keras untuk membantu mereka membangun aplikasi berbasis *website* ini. Berikut adalah spesifikasi perangkat keras yang dibutuhkan untuk membangun aplikasi ini.

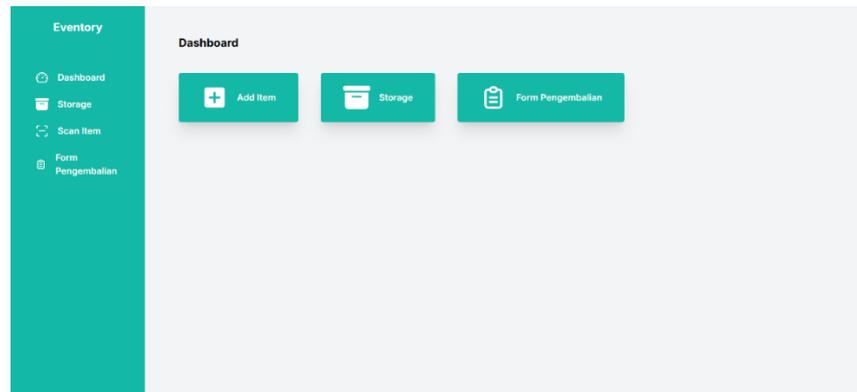
Tabel 3.4 Spesifikasi Kebutuhan Perangkat Keras

No	Perangkat Keras	Kebutuhan Perangkat Keras
1	Processor	Intel Core i5 1035G1
2	VGA	Intel UHD Graphics
3	Memory	8 GB

3.2.3.3 Tampilan dan Realisasi Aplikasi

Pembangunan aplikasi berbasis *website* ini dibangun dengan menggunakan *framework* yang biasa digunakan oleh perusahaan, *framework* yang digunakan praktikan untuk membangun aplikasi ini adalah *Next.js*. *Next.js* adalah *framework React* yang dikembangkan untuk mempermudah pembangunan aplikasi berbasis *website* dengan fitur *server-side rendering* (SSR), *static site generation* (SSG), dan *automatic code splitting* yang membuat pengembangan aplikasi web menjadi lebih mudah dan cepat dengan performa yang optimal. (Ronny *et al.*, 2022).

Praktikan menggunakan *framework* tersebut dikarenakan unggul dalam fitur modern untuk pengembangan aplikasi *front-end* dan *back-end* secara serentak. Selain itu, pada bagian *front-end* praktikan menggunakan *framework CSS Tailwind* untuk mempermudah praktikan dalam mengelola *style* halaman aplikasi web tersebut. *Tailwind* juga sudah termasuk kedalam fitur yang dimiliki *Next.js* dalam pengembangan aplikasi web di bidang *front-end*. Untuk pengelolaan *database* praktikan menggunakan platform *Supabase*, platform ini adalah salah satu platform pengelolaan *database* yang biasa digunakan oleh perusahaan. Praktikan menggunakan *Supabase* dikarenakan mendukung penggunaan *framework Next.js*. Integrasi yang mudah serta fitur *bucket storage* untuk pengelolaan file seperti gambar ataupun dokumen juga menjadi alasan mengapa praktikan menggunakan platform *Supabase* untuk pengelolaan *database* pada pembangunan aplikasi yang sedang dilakukan.

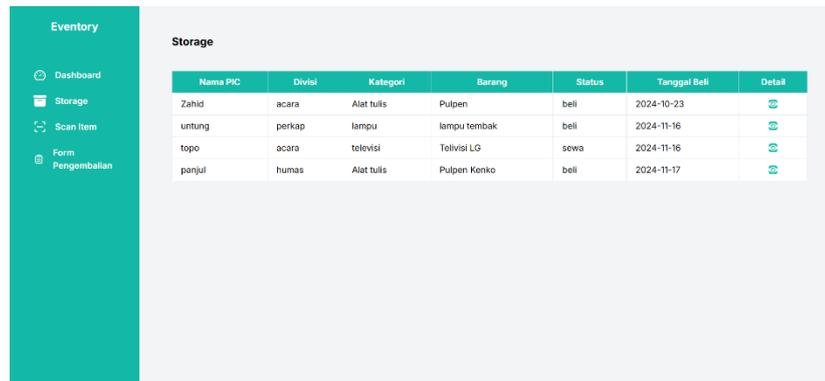


Gambar 3.13 Realisasi Halaman *Dashboard*

Gambar di atas merupakan hasil realisasi dari halaman *dashboard* yang telah disesuaikan dengan rancangan sistem dan desain sebelumnya. Pada gambar halaman *dashboard* tersebut, terdapat *button* yang masing-masing memiliki fungsi salah satunya *button add item* yang akan mengarahkan *user* langsung kepada halaman *add item*. Hal ini dilakukan agar *user* dapat dengan langsung menambahkan data barang yang ingin disimpan secara cepat.

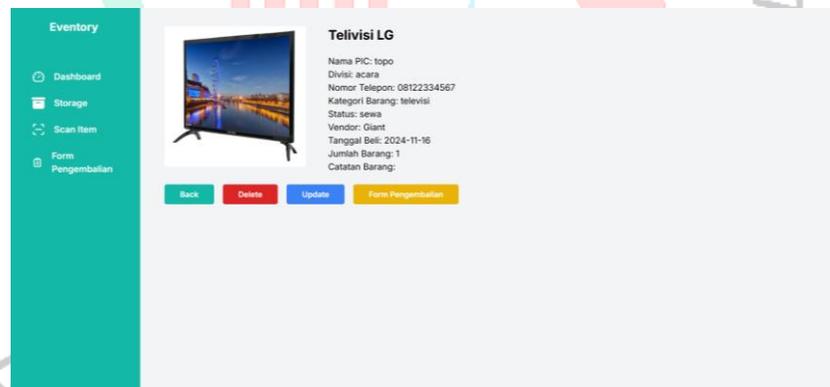
Gambar 3.14 Realisasi Halaman *Add Item*

Gambar di atas merupakan hasil realisasi dari halaman *Add Item*. Pada halaman tersebut, terdapat beberapa data yang harus diisi dengan benar oleh *user* agar data tersebut berhasil disimpan ke *database* dan dapat dilihat pada halaman *storage*.



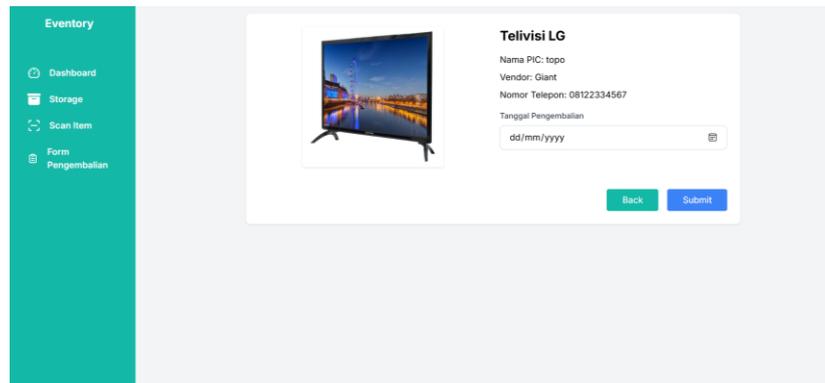
Gambar 3.15 Realisasi Halaman Storage

Gambar di atas merupakan realisasi dari halaman utama *Storage*. Sesuai dengan rancangan desain sebelumnya, pada halaman ini terdapat tabel yang berisikan beberapa data barang yang berhasil disimpan ke dalam *database* pada tahap add item sebelumnya. Selain itu, pada kolom detail terdapat *button* untuk mengarahkan *user* ke halaman data barang tersebut yang lebih lengkap.



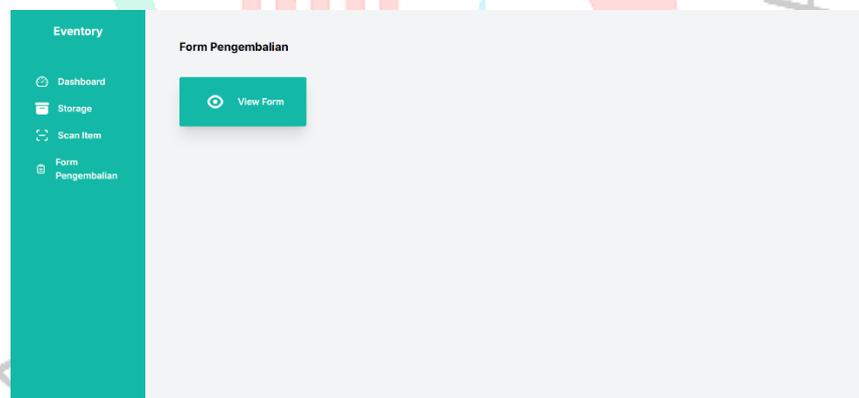
Gambar 3.16 Realisasi Halaman Detail Barang

Gambar di atas adalah realisasi dari halaman detail barang. Masih pada halaman *storage*, halaman detail barang ini menyediakan informasi lengkap terkait barang yang dipilih, selain itu, *user* juga dapat mengubah ataupun menghapus data. Halaman ini bertujuan untuk memperlihatkan kepada *user* data barang yang berhasil disimpan, dan mengelola lebih lanjut apakah data telah sesuai atau tidak.



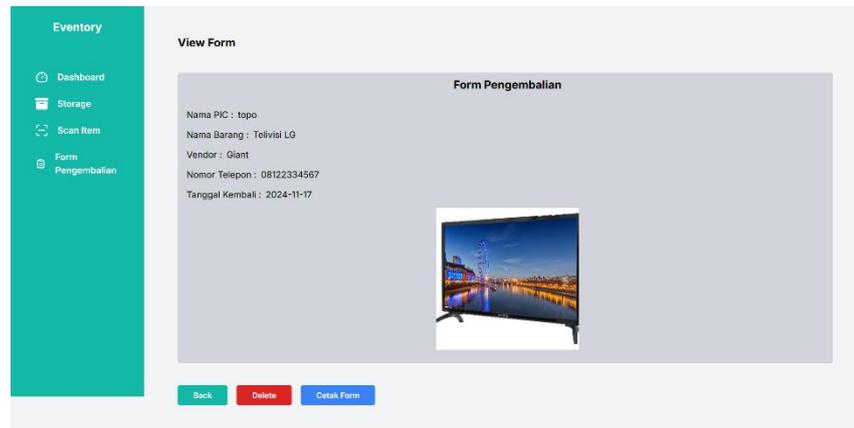
Gambar 3.17 Realisasi Halaman Input Form Pengembalian

Gambar di atas merupakan realisasi dari halaman input form pengembalian pada halaman storage. Halaman ini berfungsi untuk pembuatan formulir pengembalian yang berisi beberapa data barang. Setelah data diisi dengan benar, maka data akan disimpan ke dalam tabel *return* pada *database*. Halaman ini akan muncul ketika *user* menekan *button* form pengembalian pada halaman detail barang.



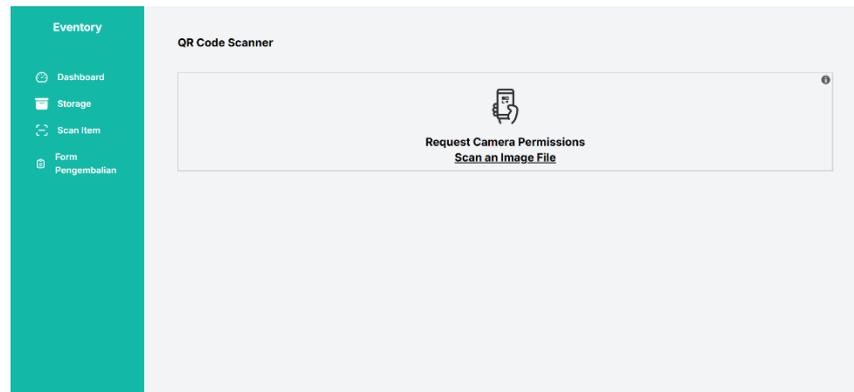
Gambar 3.18 Realisasi Halaman Utama Form Pengembalian

Gambar di atas merupakan realisasi dari halaman utama form pengembalian. Halaman ini berisi *button* untuk mengarahkan *user* ke halaman *preview* formulir pengembalian barang yang telah berhasil disimpan.

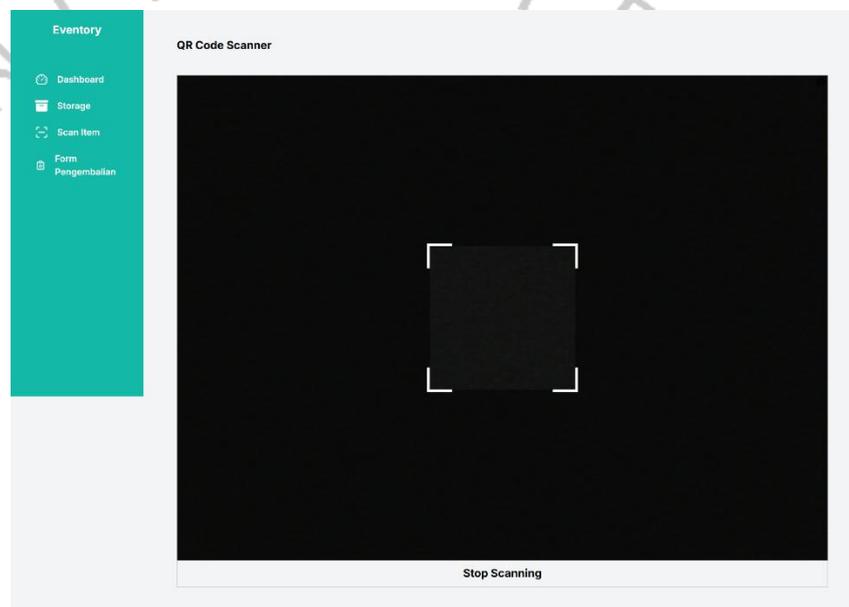


Gambar 3.19 Realisasi Halaman Preview Form Pengembalian

Gambar di atas merupakan realisasi dari halaman *preview* form pengembalian. Halaman ini berfungsi untuk memperlihatkan data barang sewa hingga waktu pengembaliannya yang dibentuk layaknya formulir, data yang terdapat pada formulir tersebut antara lain adalah nama PIC, nama barang, nama vendor, nomor telepon PIC, tanggal pengembalian barang, dan foto barang. Data tersebut adalah data yang berhasil disimpan ke dalam *database* tabel *return*, dan diperlihatkan di halaman *preview* form pengembalian ini. Selain itu terdapat fitur untuk mencetak form ke dalam format *.pdf* dan fitur untuk menghapus *preview* form apabila *user* telah mencetak form pengembalian tersebut.



Gambar 3.20 Realisasi Halaman Scan Item



Gambar 3.21 Realisasi Fitur Kamera Untuk Scan Item

Gambar di atas merupakan realisasi dari halaman *scan item*. Pada halaman ini, terdapat fitur *scan item* yang berfungsi untuk melakukan *scanning* pada QR Code barang. Fitur scanning bisa dilakukan secara langsung lewat kamera device atau lewat file foto QR Code.

3.3 Kendala yang Dihadapi

Praktikan telah bekerja dengan baik sebagai karyawan magang di divisi developer PT Brainlens Rekayasa Konsultan. Namun, terdapat beberapa kendala selama bekerja pada pembangunan aplikasi berbasis website. Berikut adalah hambatan yang dihadapi oleh praktikan:

- 1) Implementasi bahasa pemrograman yang terbatas, terutama penggunaan *framework* *Next.js* dan platform pengelola *database* *Supabase*, menyebabkan beberapa tantangan dan perlu waktu yang lebih.
- 2) Kurangnya rekan kerja magang menyebabkan pembangunan aplikasi tersebut memakan waktu cukup lama dan terbatasnya inovasi baru yang didapatkan dari rekan kerja.
- 3) Sulitnya berkomunikasi secara tatap muka ataupun *virtual* dengan mentor
 - yang menyebabkan inovasi baru atau pengalaman dunia kerja yang didapatkan dari mentor tidak banyak.

3.4 Cara mengatasi Kendala

Selama praktikan menjalani kegiatan magang tentunya akan ada beberapa kendala yang harus dihadapi oleh praktikan untuk menambah kemampuan dan pengalaman di dunia kerja nanti. Berikut adalah cara mengatasi beberapa kendala yang telah dijabarkan sebelumnya.

- 1) Praktikan melakukan eksplorasi tentang penggunaan *framework* dan platform yang berkaitan lewat *internet*, serta berkonsultasi dengan mentor agar pembangunan aplikasi ini dapat berjalan dengan baik dan sesuai rencana.
- 2) Praktikan terus menerapkan diskusi secara langsung dengan rekan kerja ketika masing-masing memiliki waktu luang, diskusi tersebut membahas terkait progres kerja, membagi jadwal pekerjaan, serta saling membagi pengetahuan yang berkaitan dengan pembangunan aplikasi ini.
- 3) Setiap adanya kesempatan berdiskusi dengan mentor, praktikan dan rekan kerja selalu memaksimalkan pertanyaan yang diajukan kepada mentor agar pengetahuan yang didapat untuk melaksanakan pekerjaan juga maksimal.

3.5 Pembelajaran yang Diperoleh dari Kerja Profesi

Selama praktikan bekerja di perusahaan PT Brainlens Rekayasa Konsultan, praktikan mendapatkan beberapa pembelajaran dari pembangunan aplikasi berbasis *website* ini. Berikut adalah pembelajaran yang diperoleh praktikan dari kerja profesi.

- 1) Praktikan mendapatkan pembelajaran tentang bagaimana membangun aplikasi pengelolaan data logistik berbasis *website*.
- 2) Praktikan mendapatkan pembelajaran terkait penggunaan *framework Next.js*, *Tailwind CSS*, dan platform pengelolaan *database Supabase* yang biasa digunakan perusahaan untuk mengembangkan sebuah aplikasi atau *website*.
- 3) Praktikan mendapatkan wawasan baru tentang apa saja tantangan yang akan ditemui di dunia kerja nanti terutama bagi seorang yang bekerja sebagai *developer*.