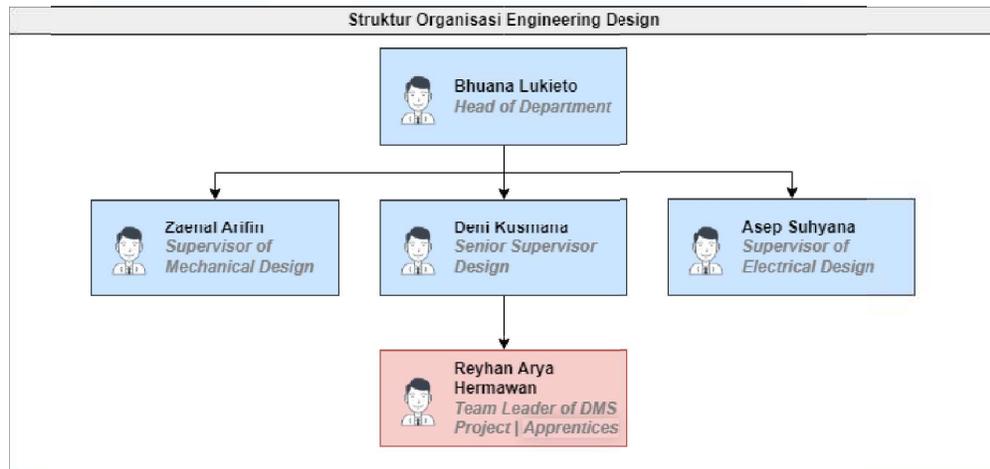


BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi



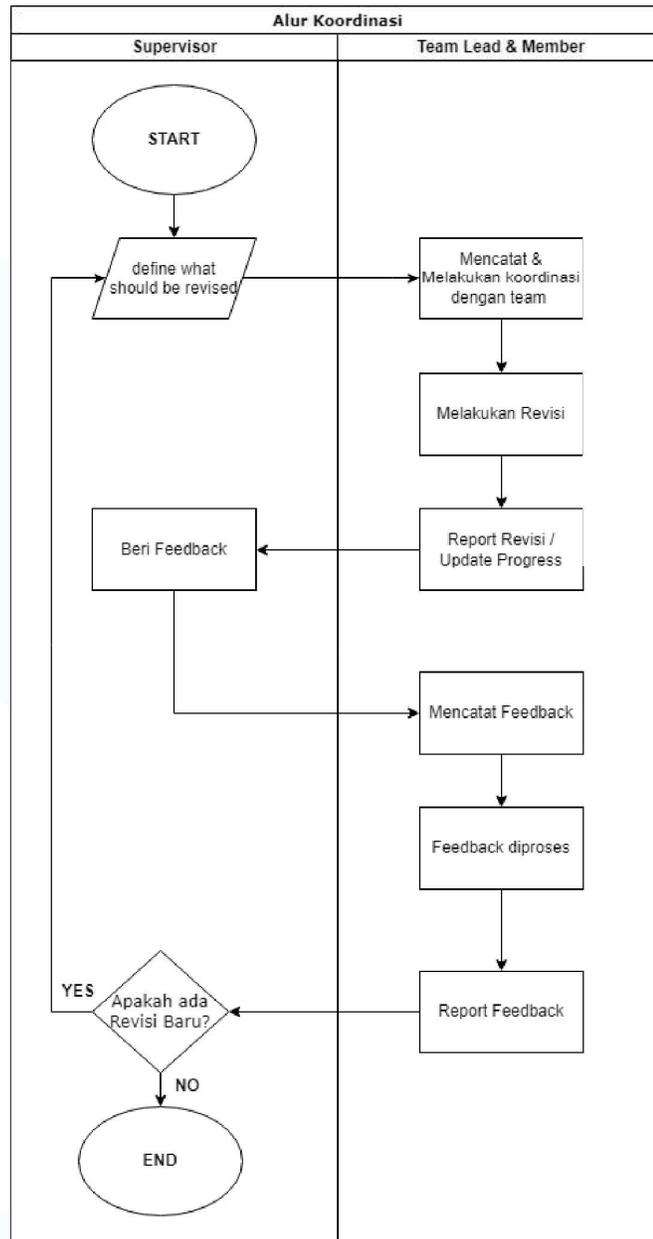
Gambar 3. 1 Kedudukan peserta magang dalam struktur organisasi departemen Engineering Design

Awalnya dalam pelaksanaan kerja magang, peserta magang berada dibawah naungan talent management saat melakukan *briefing*, *onboarding* dan persiapan penerimaan peserta magang. Tetapi setelah proyek akan berjalan, keseharian peserta magang sebagai *Front-end developer* berada di kantor *Engineering Design*.

Dalam program praktik kerja magang ini, peserta magang yang tergabung dalam tim MIT Design berada dibawah naungan supervisi Engineering Design PT. Bridgestone Tire Indonesia. Tim berada langsung dibawah pengawasan *Senior Supervisor* dari departemen *Engineering Design* yaitu Bapak Deni Kusmana. Tim MIT Design menentukan salah satu dari anggotanya untuk menjadi team leader sebagai perwakilan tim selama periode magang berlangsung. Pengerjaan dan koordinasi dengan tim MIT Design dilakukan setiap hari di kantor *Engineering Design*. Apabila salah seorang anggota tim mengambil hak *WFH*-nya, maka koordinasi dilakukan melalui platform *chat*, *call* serta *video call* pada media sosial. Koordinasi dengan mentor atau Senior Supervisor dilakukan dengan mengadakan

meeting setiap satu minggu sekali untuk melaporkan progress pengerjaan proyek. Meeting progress ini juga dilakukan untuk mendapatkan pendapat mengenai kelebihan dan kekurangan dari proyek terkait.

Tim MIT Design juga melakukan koordinasi dengan peserta magang dari bagian *Administrative standard* untuk mengerjakan proyek DMS. Proyek DMS dikerjakan oleh tiga orang peserta magang yakni, *Reyhan Arya Hermawan* sebagai *Team Leader* sekaligus *Front-end Developer* dari tim MIT Design, Wanda Taupik Ramdan sebagai *Back-end Developer* dari tim MIT Design, dan Khoirunisa sebagai *Administrator* untuk melakukan *indexing* dan pengelolaan dokumen dari bagian *Administrative standard*. Tim MIT Design juga melakukan diskusi dan konsultasi dengan peserta magang dari bagian LSCM terkait proyek DMS.



Gambar 3. 2 Alur Koordinasi dalam program kerja magang

3.2 Tugas dan Uraian Kerja Magang

Selama periode magang berlangsung, peserta kerja magang diminta untuk mengerjakan sebuah proyek berupa pembuatan DMS (Document Management Systems) berbasis *website*. Berikut adalah rincian mengenai aktivitas yang dilakukan peserta magang dalam tim proyek DMS:

1. Melakukan pengenalan terhadap departemen Engineering Design oleh mentor kepada peserta magang.
2. Melakukan project kickstart meeting.
3. Membuat project Charter mengenai proyek DMS.
4. Mendefinisikan project scope & objectives.
5. Melakukan setup environment untuk keperluan pengerjaan proyek DMS.
6. Menentukan strategi perancangan.
7. Melakukan QC pada perpustakaan di departemen Engineering Design.
8. Mempersiapkan Presentasi Tahap Perencanaan Proyek.
9. Menganalisis Fitur yang akan dirancang
10. Menganalisis Finalisasi Fitur untuk Rilis Awal
11. Melakukan Use Case Analysis
12. Menyusun Activity Diagram
13. Dokumen SRS.
14. Analysis of the Needs of Data and Document Traffic
15. ERD Analysis and Design (DB Schema)
16. User Interface Design and Wireframing
17. Website Design
18. Website Test (Directly Jump to UAT)
19. Website Deployment
20. Trial, Training, and Monitoring
21. Completion & Hand Over Report

Kegiatan pada poin 1 akan berfokus kepada pengenalan lingkungan departemen engineering design, diperkenalkan juga karyawan dan mentor-mentor yang akan membimbing peserta magang. Kegiatan pada poin 2 dan 3 akan berfokus dalam pemberian informasi dari mentor kepada peserta magang mengenai proyek seperti apa yang akan dilakukan dan dibuat. Aktivitas pada poin 4 hingga poin 8 akan berfokus pada fase perencanaan awal sistem dimulai dengan mendefinisikan *project scope*, menentukan metode perancangan *software* yang akan digunakan, mempersiapkan *environment* untuk perancangan sampai melakukan *quality check*

terhadap dokumen *hardcopy* yang dibutuhkan untuk dijadikan *softcopy*. Kegiatan pada poin 9 sampai poin 16 pada periode kerja magang akan berfokus dalam fase analisis sistem dimulai dari menentukan fitur prioritas, menyusun analisis *use case*, hingga melakukan *indexing* terhadap dokumen terkait. Aktivitas pada poin 17 hingga poin 19 akan berfokus dalam penyusunan skema *database* dan mulai melakukan pembuatan *wireframe* dan *user interface*. Kegiatan pada poin 20 sampai poin 23 akan berfokus pada fase perancangan mulai dari pengerjaan *front-end* dan *back-end* secara *parallel* sampai melakukan *testing* dan *deployment*. Aktivitas pada poin 24 hingga 25 akan berfokus pada fase integrasi dimana peserta magang melakukan *trial* dan *training* kepada calon pengguna aplikasi yang dibuat serta melakukan *monitoring* terhadap aplikasi yang telah dibuat, selain itu peserta magang juga melakukan *handover project* kepada pihak terkait.

Tabel 3. 1 Aktivitas Kerja Magang

DMS Project			
No.	Kegiatan	Waktu Pelaksanaan	
		Tanggal Mulai	Tanggal Selesai
Project Initiate			
1.	Engineering Design Department Introduction	20 Februari 2023	24 Februari 2023
2.	Project Kickstart Meeting	24 Februari 2023	24 Februari 2023
3.	Project Charter	27 Februari 2023	1 Maret 2023
Initial System Planning Phase			
4.	Defining Project Scope and Objective	3 Maret 2023	7 Maret 2023
5.	Setup Environment of Design	13 Maret 2023	15 Maret 2023
6.	Determining the Strategy of Design	15 Maret 2023	17 Maret 2023
7.	Preparation for the Project Planning Phase Presentation	8 Maret 2023	9 Maret 2023
System Analysis Phase & Administrative Standard			
8.	Feature Design Analysis	21 Maret 2023	22 Maret 2023
9.	Feature Finalization Analysis for Initial Release	22 Maret 2023	23 Maret 2023
10.	Use Case Analysis	23 Maret 2023	24 Maret 2023
11.	Compiling Activity Diagrams	24 Maret 2023	28 Maret 2023
System Robustness Design Phase			

12.	ERD Analysis and Design (DB Schema)	5 April 2023	7 April 2023
13.	User Interface Design and Wireframing	9 Maret 2023	11 April 2023
Design Phase			
14.	Website Design	11 April 2023	9 Juni 2023
15.	Website Test (Directly Jump to UAT)	12 Juni 2023	16 Juni 2023
16.	Website Deployment	12 Juni 2023	16 Juni 2023
Integration Phase			
17.	Trial, Training, and Monitoring	19 Juni 2023	21 Juni 2023
18.	Completion & Hand Over Report	22 Juni 2023	23 Juni 2023

3.2.1 Pengenalan Lingkungan Departemen *Engineering Design*

Dalam periode pelaksanaan pengenalan lingkungan departemen *engineering design* ini, peserta magang diminta untuk mengenal seluruh lingkungan kerja dimulai dari area *Head Office* sampai ke lingkungan pabrik. Peserta magang diajak untuk mengikuti berbagai kegiatan yang bertujuan untuk mengenal lebih dalam mengenai produk-produk PT. Bridgestone Tire Indonesia, mengenal SOP yang berlaku pada perusahaan terutama pada departemen *engineering design*, *safety induction* untuk keselamatan kerja mulai dari pakaian yang harus dipakai sampai melakukan *KYT Report (Kiken Yochi Training)*, dimana hal ini dapat membekali peserta magang mengenai pengetahuan dan keterampilan dalam melakukan aktifitas keselamatan dan mengantisipasi kecelakaan kerja.

Pada periode ini juga dilakukan juga pembukaan oleh *HR-GA Director* serta manajemen dan jajarannya untuk menyambut dan melakukan tanda tangan kontrak Bersama peserta magang. Peserta magang ditemukan dengan mentor masing-masing sesuai dengan divisinya Dilakukan juga *briefing* kepada peserta magang mengenai gambaran besar hal yang akan dilakukan saat periode proyek dimulai dan proyek apa yang akan dikerjakan. Dalam pelaksanaan praktik kerja magang, peserta magang mendapatkan proyek *DMS* dimana dilakukan

pembuatan aplikasi berbasis *website* sebagai wadah untuk menyimpan dokumen fisik yang sudah di *scan*.

3.2.2 Melakukan *Kickstart Meeting*

Dalam pelaksanaannya pada periode ini, dilakukan pertemuan dengan berbagai mentor untuk membahas mengenai proyek apa yang akan dibuat oleh peserta magang. Peserta magang mendapatkan proyek DMS dimana pemegang diberikan kesempatan untuk membuat aplikasi berbasis *website* sebagai wadah untuk menyimpan dokumen fisik yang sudah di *scan* (*Softcopy*). Proyek DMS dikerjakan oleh suatu tim yang terdiri dari 3 peserta magang dengan bidang masing-masing yakni bidang *front-end*, *back-end* dan *administrator*. Pada periode ini, tim yang melakukan pengerjaan pada proyek DMS diberikan referensi mengenai program DMS itu sendiri oleh mentor.

3.2.3 Membuat *Project Charter*

Pembuatan project charter dibutuhkan sebagai pendefinisian awal proyek agar gambaran proyek terlihat secara jelas karena *project charter* berisi elemen-elemen unik yang mendefinisikan suatu proyek. Tabel berikut adalah *project charter* dari proyek DMS.

Tabel 3. 2 Tabel Project Charter

Project Description <ul style="list-style-type: none"> • Penjelasan singkat 	Design of a website-based Document Management System (DMS) as a means of stored and standardized all documents belonging to the Engineering Design Department.
Project Justification <ul style="list-style-type: none"> • Latar belakang dan mengapa proyek ini penting 	<ul style="list-style-type: none"> • Difficulties in managed and maintained engineering hardcopy documents. • Need ease of collaboration in terms of data engineering at PT. Bridgestone Tire Indonesia. • To increase productivity from the performance of Engineering Department employees.
Success Criteria <ul style="list-style-type: none"> • Bagaimana cara mengukur keberhasilannya? 	<ul style="list-style-type: none"> • DMS Website is available and socialized to users to be ready for implementation • DMS Website can store all engineering softcopy documents
Deliverables/Outputs <ul style="list-style-type: none"> • Item apa yang dihasilkan? 	<ul style="list-style-type: none"> • Document Management System (DMS) Website • Standardized Engineering Department Library
Milestones <ul style="list-style-type: none"> • Apa saja tahap penting yang akan dicapai? • Jadwal program 	<ol style="list-style-type: none"> 1. Project Initiate 2. Initial System Planning Phase 3. System Analysis Phase & Administrative Standard 4. System Robustness Design Phase

	<ul style="list-style-type: none"> 5. Design Phase 6. Integration Phase
Key Assumptions <ul style="list-style-type: none"> • Asumsi yang dibuat agar proyek dapat berjalan dengan lancar 	<ul style="list-style-type: none"> • Mendapatkan dukungan dari manajemen • Anggota tim capable dalam menjalankan tugasnya • Proyek memiliki requirement yang jelas
Key Team Members <ul style="list-style-type: none"> • Anggota tim 	<ul style="list-style-type: none"> • Reyhan Arya Hermawan (Front-end) • Wanda Taupik Ramdan (Back-end) • Khoirunisa (Administrator)
Resources Requirement <ul style="list-style-type: none"> • Sumber daya manusia • Dana • Infrastruktur, dll 	<ul style="list-style-type: none"> • Manpower Resources: DMS Team, Mentors
Risks <ul style="list-style-type: none"> • Resiko program • Hal yang mungkin tidak sesuai rencana? • Bagaimana mitigasinya? • Resiko jika program tidak dijalankan 	<ul style="list-style-type: none"> • The project is not finished perfectly • The course of project exceeded the estimated schedule because there are many custom feature requests on web systems to be developed • Risk of data work being lost and hampered due to device damage • Leak of document data.

3.2.4 Mendefinisikan *Project Scope* dan Tujuan

Project scope didefinisikan untuk mengetahui cakupan proyek. Cakupan proyek pada proyek DMS hanya pada departemen *engineering design* saja. Sedangkan tujuan dari proyek DMS ini adalah mengembangkan dan merancang sistem informasi yang efisien dalam pengelolaan arsip dokumen yang disimpan di Perpustakaan departemen *engineering*.

3.2.5 Menyiapkan Environment Untuk Perancangan

Dilakukan persiapan *environment* seperti *software* dan *framework* untuk melakukan perancangan *website* agar mendukung proses perancangannya. *Software* yang digunakan dalam proses perancangan oleh peserta magang yakni Visual Studio Code, Figma, Canva dan Github. *Framework* yang digunakan dalam proses perancangan adalah Bootstrap untuk desain tampilan antarmuka pengguna dan Laravel untuk perancangan *Back-end*.

3.2.6 Menentukan Strategi

Adapun strategi yang dimaksud pada uraian kali ini adalah pendekatan pengelolaan dan perancangan perangkat lunak. Pendekatan Agile digunakan dalam perancangan *website* DMS. Pendekatan Agile digunakan karena bersifat

fleksibel dan adaptif terhadap perubahan kebutuhan, serta efisien.[10] Melihat dari periode waktu magang yang terbatas, metode agile adalah metode yang tepat untuk perancangan proyek website DMS.

3.2.8 Melakukan Presentasi Mengenai Project yang Akan Dibuat

Dilakukan presentasi mengenai proyek yang akan dibuat. Sebelum proses perancangan, presentasi dilakukan untuk mendapatkan *approval* dari mentor. Diajukan fitur-fitur yang akan dibuat, color pallete, dan ide-ide yang dapat membuat website menjadi lebih menarik.

3.2.9 Menganalisa Fitur-Fitur yang Akan Dirancang

Dilakukan analisis terhadap fitur-fitur yang akan dikembangkan pada *website* DMS. Setelah dianalisis, berikut adalah beberapa fitur-fitur utama untuk perancangan *website* DMS untuk *role User*:

- Fitur *Tree View (Tree Branch)*
Tree Branch merupakan fitur tampilan pohon yang merepresentasikan visual hierarki data dalam bentuk struktur berbentuk pohon. Dalam *tree view*, data diorganisasi dalam bentuk cabang-cabang yang saling terhubung, mirip dengan cabang dan daun pada pohon sebenarnya.
- Fitur *Favourites*
Fitur ini dibuat untuk menampilkan *file* yang sudah dijadikan favorit oleh user.
- Fitur User Information
Fitur ini merupakan informasi mengenai pengguna yang sedang *online* pada *website* DMS.
- Fitur Search:
Fitur ini yang memungkinkan pengguna untuk mencari dan menemukan dokumen yang spesifik dalam *website*.

Adapun beberapa fitur penting untuk perancangan *website* DMS untuk *role Admin*:

- Fitur Add & Edit User
Fitur *Add user* memungkinkan admin untuk menambah pengguna lain. Fitur *Edit user* digunakan untuk mengubah akses dokumen yang dimiliki user.
- Fitur Add Document
Fitur yang memungkinkan admin untuk menambah dokumen lain.
- Fitur List User
Fitur yang memungkinkan admin untuk melihat *list* pengguna.
- Fitur History
Fitur yang memungkinkan admin untuk melihat aktivitas pengguna, khususnya saat pengguna melakukan *view & download* dokumen.
- Fitur Rename Document
Fitur yang memungkinkan admin untuk melakukan perubahan nama dokumen.
- Fitur *Tree View (Tree Branch)*
- Fitur User Information
- Fitur Search

3.2.10 Melakukan Finalisasi Pada Analisa Fitur

Setelah menganalisis fitur-fitur yang akan dikembangkan, diputuskan juga fitur final untuk tahap selanjutnya. Berikut adalah fitur final pada role User:

- Fitur *Tree View (Tree Branch)*
- Fitur User Information
- Fitur Search

Adapun beberapa fitur final pada *role Admin*:

- Fitur Add & Edit User
- Fitur Add Document
- Fitur List User
- Fitur History

- Fitur Rename Document

- Fitur Password Document

Fitur ini digunakan untuk memberikan keamanan berupa password pada dokumen yang diunduh oleh pengguna.

- Fitur *Tree View (Tree Branch)*

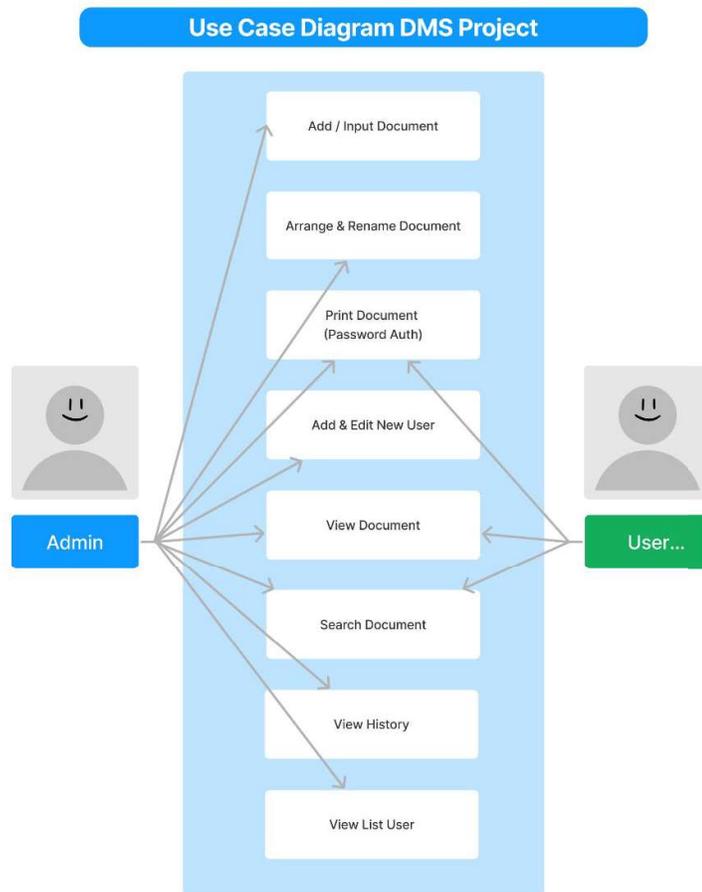
- Fitur User Information

- Fitur Search

3.2.11 Menyusun Use Case

Use case (kasus penggunaan) adalah sebuah teknik yang digunakan dalam analisis dan perancangan sistem yang berfokus pada interaksi antara aktor (pengguna) dan sistem. *Use case* digunakan untuk mendokumentasikan fungsionalitas sistem dari perspektif pengguna, dengan menggambarkan bagaimana aktor menggunakan sistem untuk mencapai tujuan tertentu.

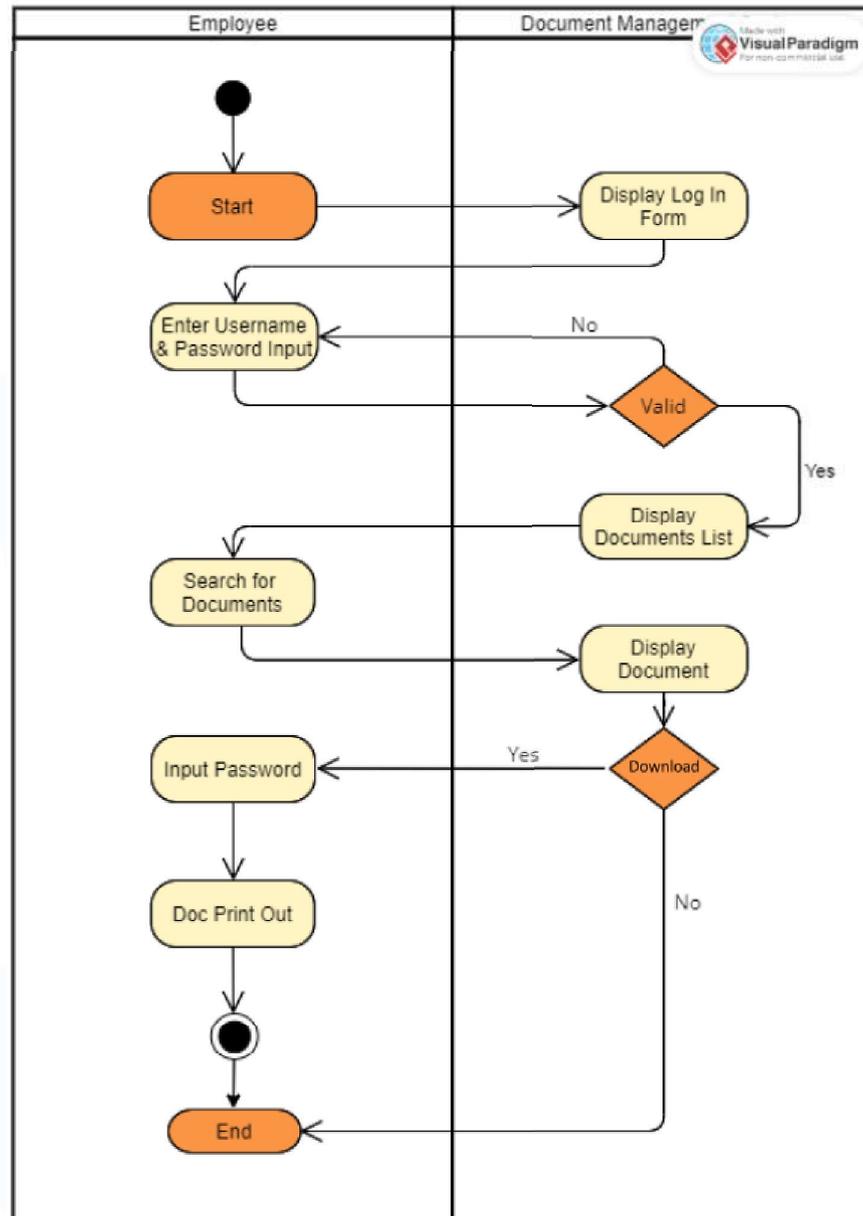




Gambar 3. 3 Use Case Diagram dari proyek DMS

3.2.12 Menyusun Activity Diagrams.

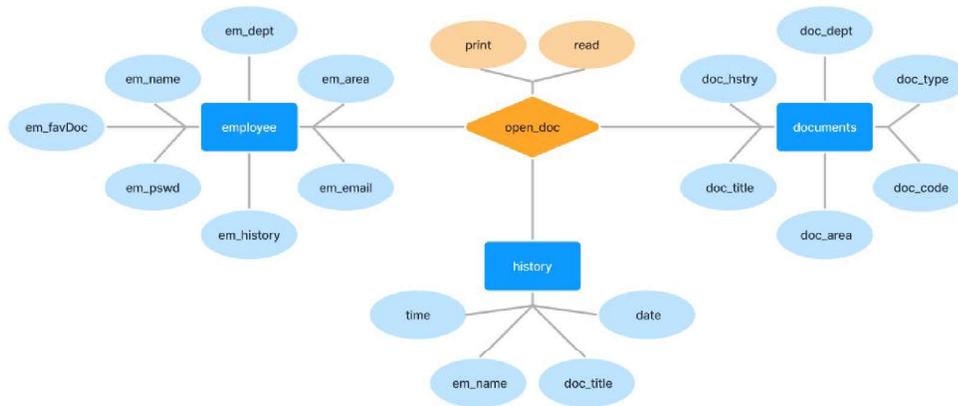
Dalam pelaksanaan kerja magang, activity diagram dibuat untuk menggambarkan serangkaian aktivitas atau tindakan yang terjadi dalam suatu proses atau sistem. Activity diagram membantu dalam memvisualisasikan bagaimana aktivitas-aktivitas saling berhubungan.



Gambar 3. 4 Activity Diagram User dari proyek DMS

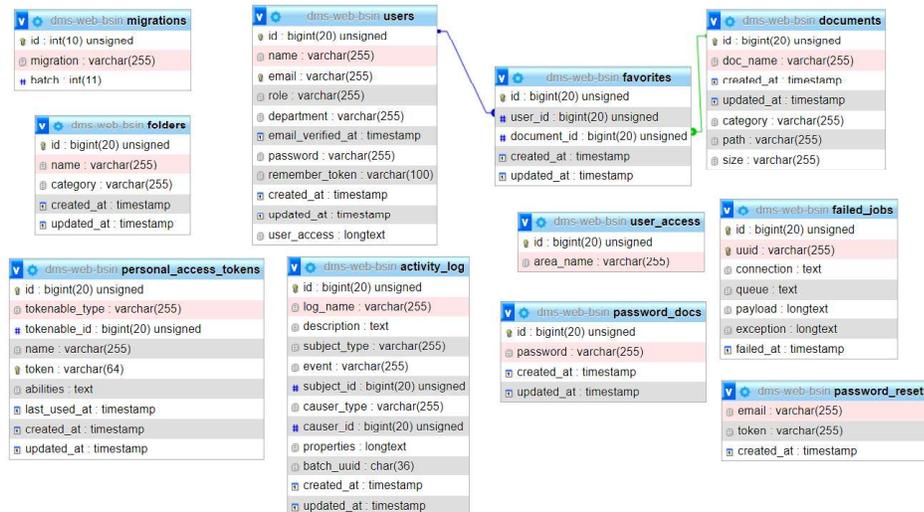
3.2.14 Menganalisa Desain *ERD* dan Skema *Database*

Dilakukan analisis terhadap desain *ERD* dan skema *Database* dari *website* DMS. *ERD* merupakan sebuah diagram yang digunakan untuk menggambarkan hubungan antara entitas (objek) dalam suatu sistem atau basis data.



Gambar 3. 5 Entity Relationship Diagram dari proyek DMS

Skema *database* adalah struktur logis yang mendefinisikan bagaimana data akan disimpan, diatur, dan dihubungkan dalam suatu basis data. Skema *database* memberikan gambaran tentang tabel, kolom, kunci, hubungan, dan batasan data lainnya yang ada dalam basis data. Adapun skema *database* yang dibuat untuk *website* DMS.



Gambar 3. 6 Skema Database dari proyek DMS

3.2.15 Merancang *Wireframe* dan *User Interface*

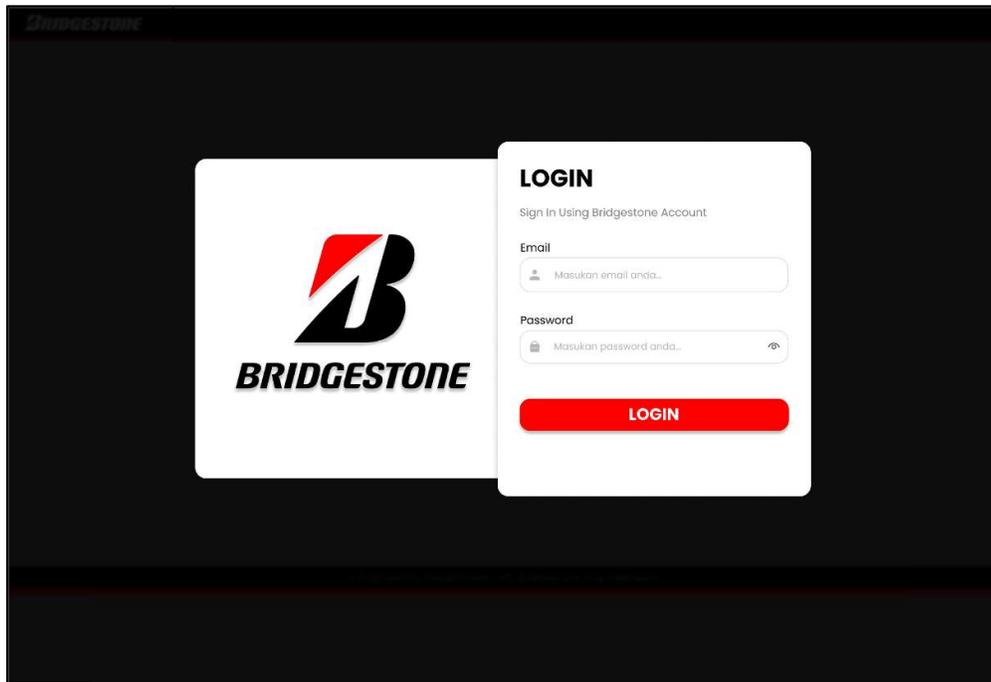
Dalam pelaksanaannya, dilakukan perancangan *wireframe* & *user interface* pada aktivitas kerja magang ini. *Wireframe* dibuat untuk menggambarkan struktur dan tata letak elemen-elemen penting pada halaman

atau aplikasi. Setelah itu, desain antarmuka pengguna (*UI*) dirancang dengan mempertimbangkan palet warna yang sesuai, font yang mudah dibaca, dan ikon atau elemen grafis yang relevan. Desain *UI* juga memperhatikan kegunaan dan responsivitas, sehingga pengguna dapat mengakses dan berinteraksi dengan mudah. Selama proses pembuatan *wireframe* dan desain *UI*, kolaborasi dengan tim pengembang lainnya dilakukan untuk mendapatkan *feedback* yang berharga. Kritik konstruktif diterima dengan baik untuk melakukan perbaikan yang diperlukan. Prinsip-prinsip desain responsif diperhatikan agar tampilan dan pengalaman pengguna tetap konsisten dan optimal di berbagai ukuran layar.

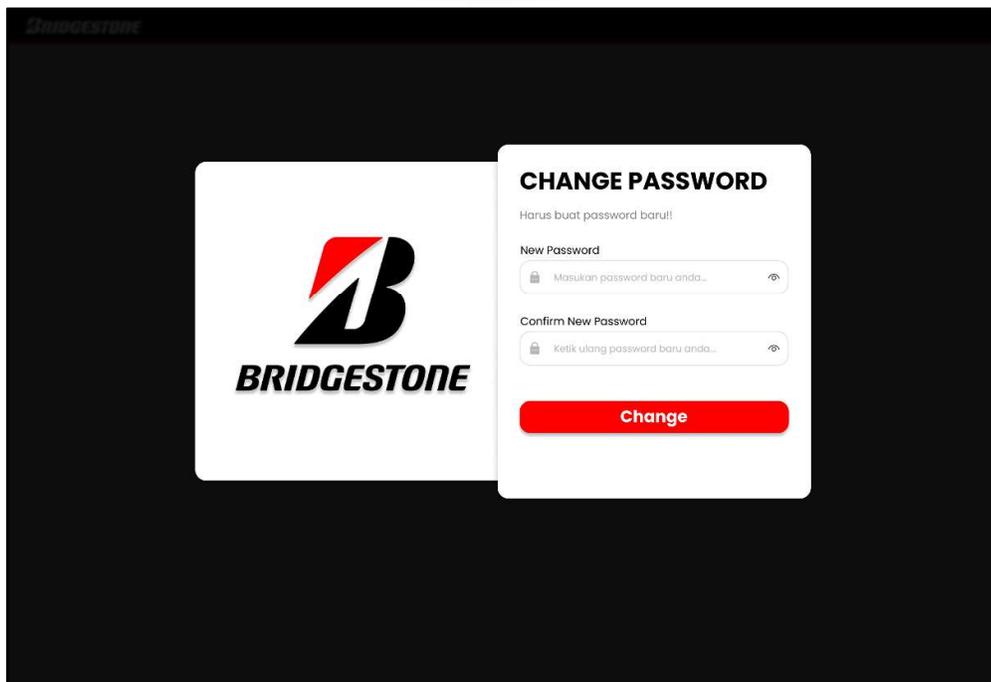


Gambar 3. 7 Collor Pallette dari web DMS

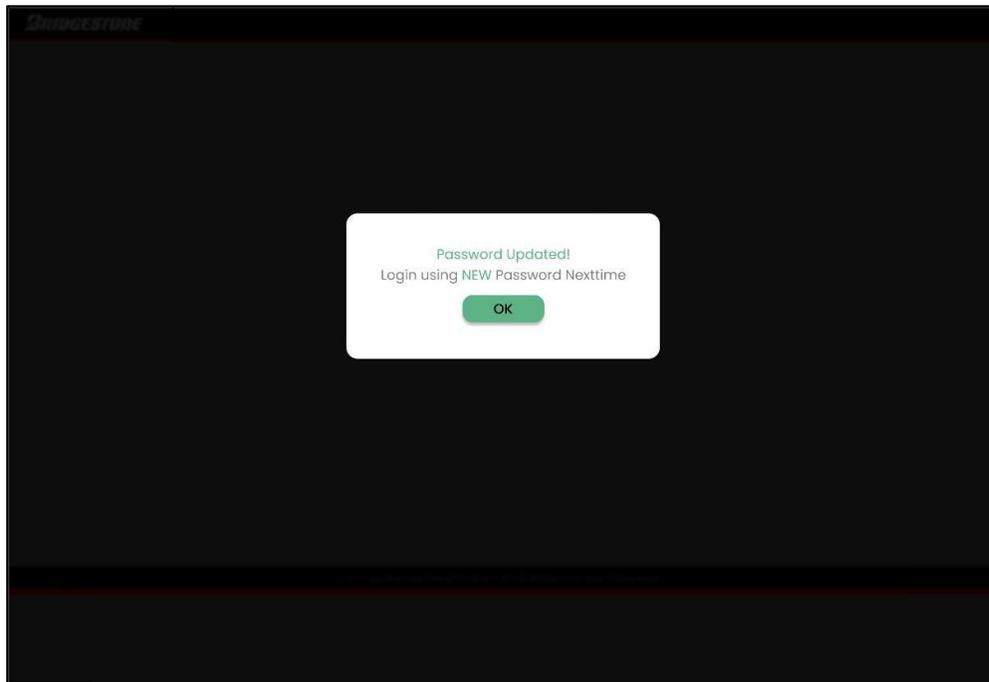
Dilakukan perancangan *UI* awal menggunakan software figma. Tampilan akan dibuat sesuai permintaan mentor, yakni seperti referensi yang diperlihatkan oleh mentor pada aktivitas [3.2.2](#) dimana tampilan memiliki *sidebar* untuk navigasinya. Tampilan antarmuka pengguna (*UI*) dibagi menjadi dua *role* yang berbeda yaitu *user* dan admin. Gambar 3.8 sampai gambar 3.18 merupakan desain antarmuka (*UI*) awal dari *role user*.



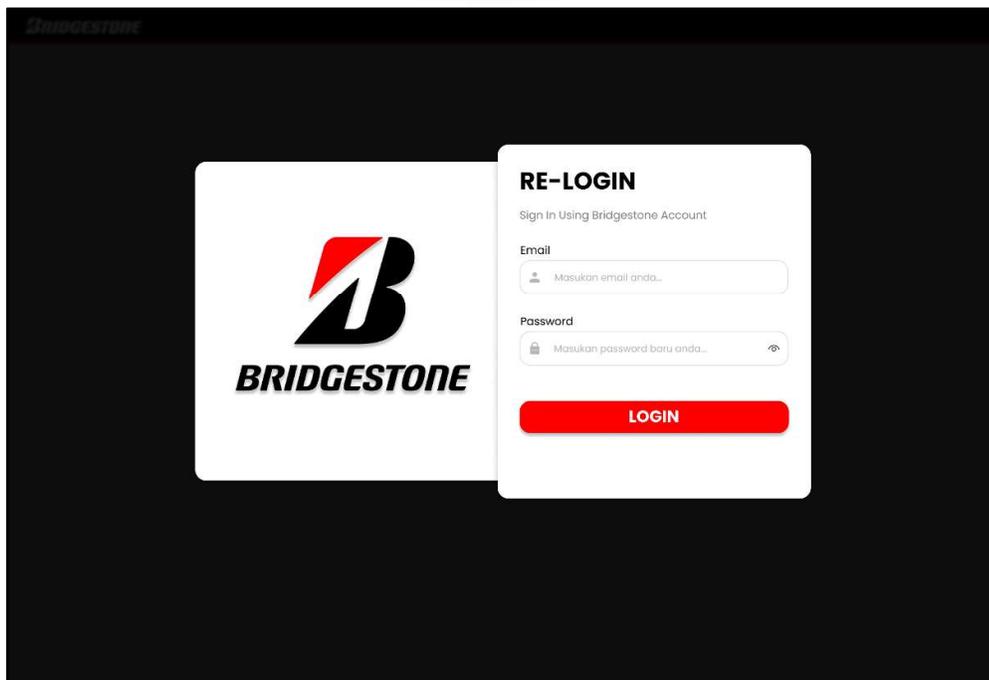
Gambar 3. 8 Login Page



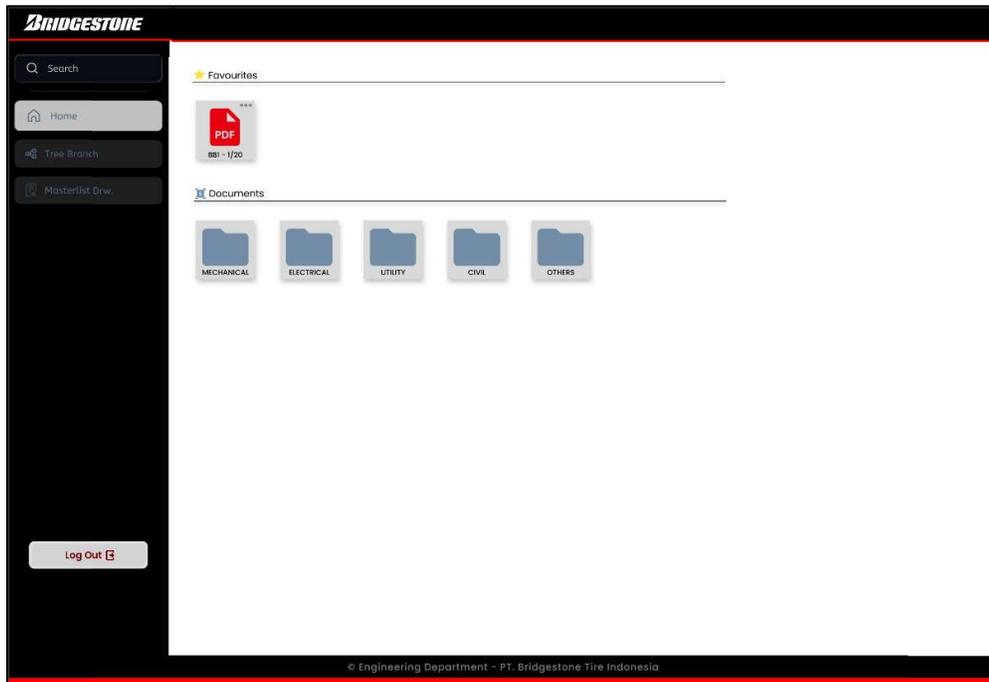
Gambar 3. 9 Change Password Page



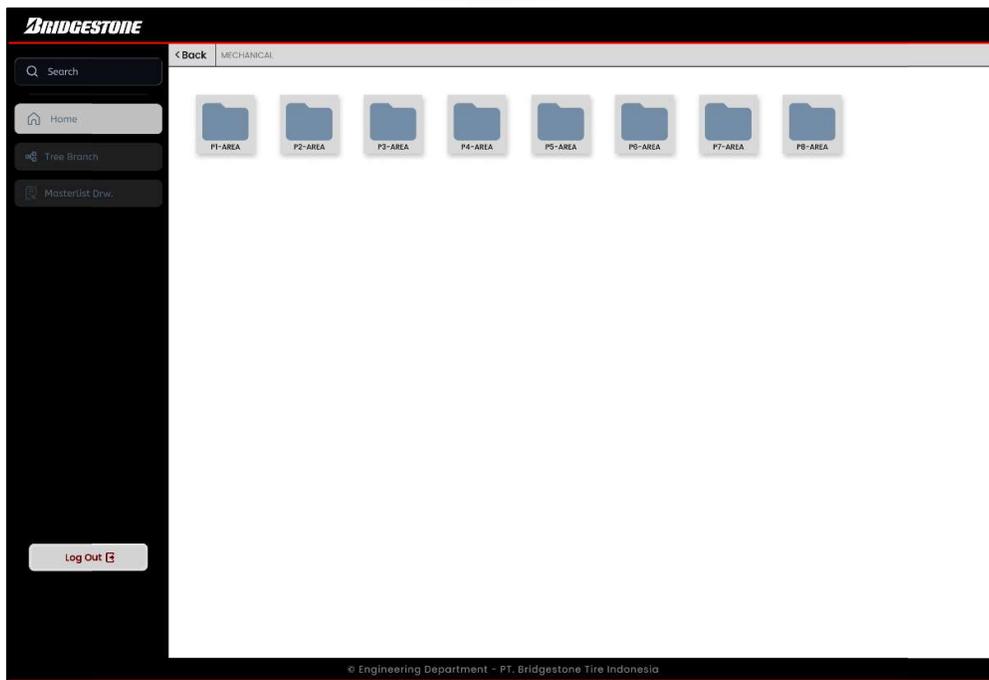
Gambar 3. 10 Alert Password Update



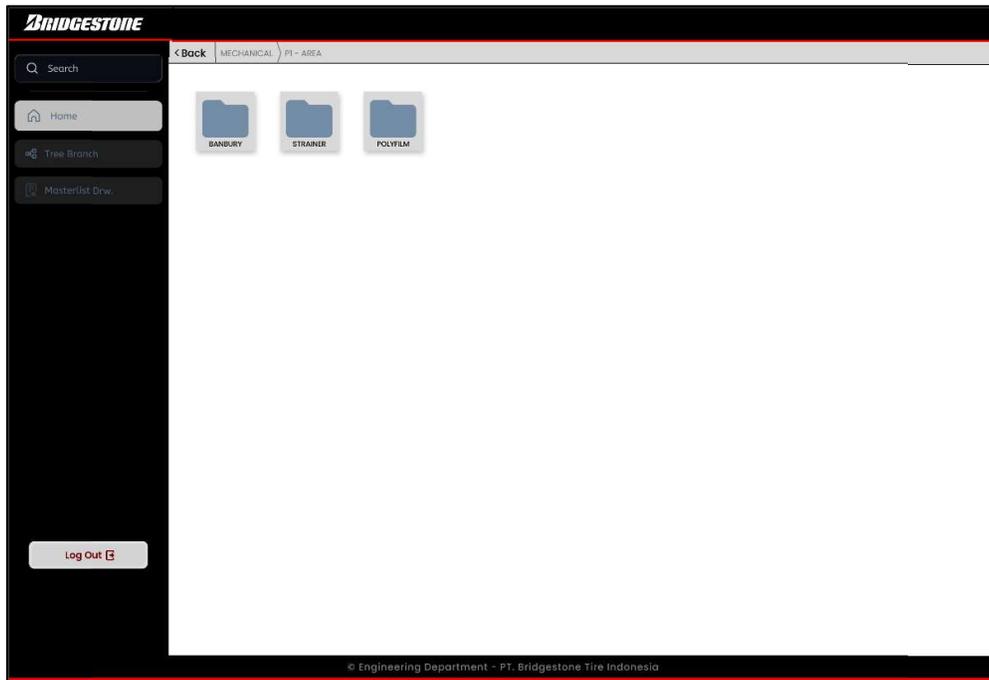
Gambar 3. 11 Re-Login Page



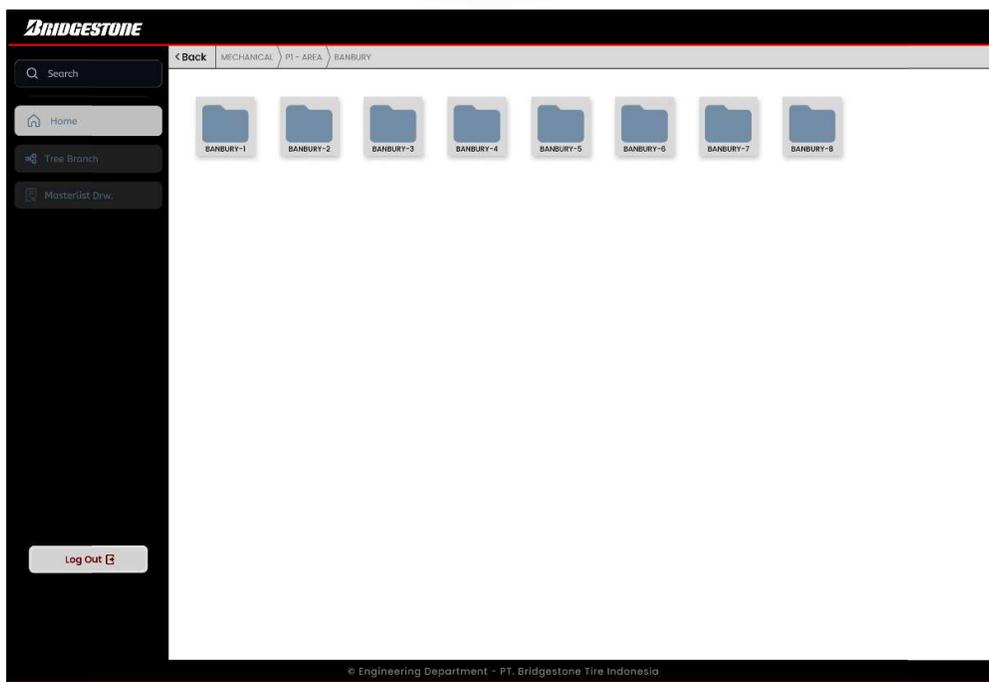
Gambar 3. 12 Home Page User



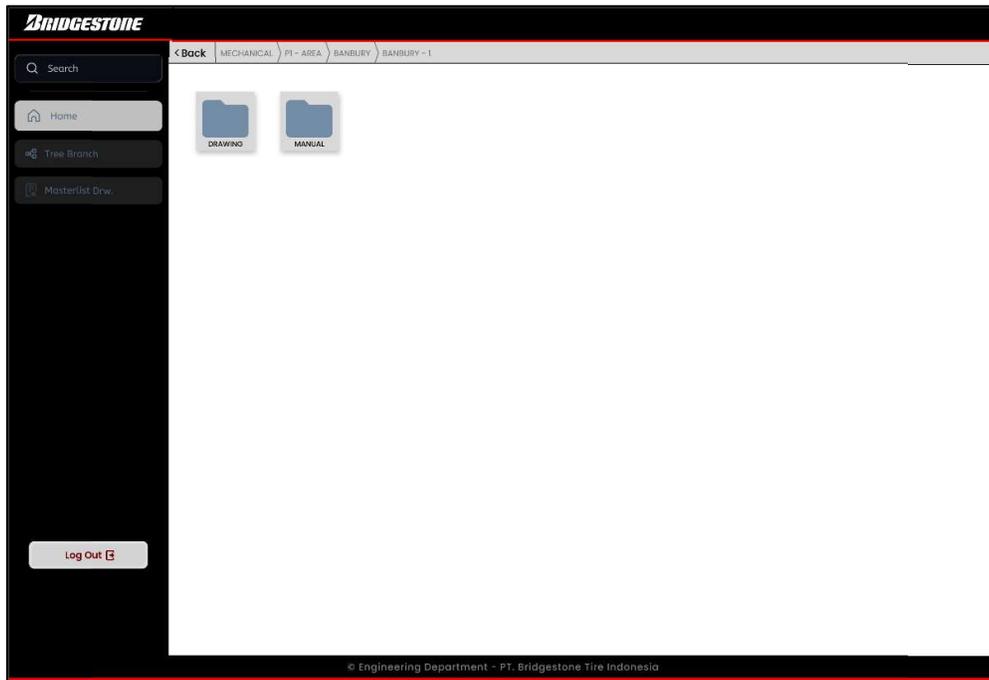
Gambar 3. 13 Folder Mechanical Page



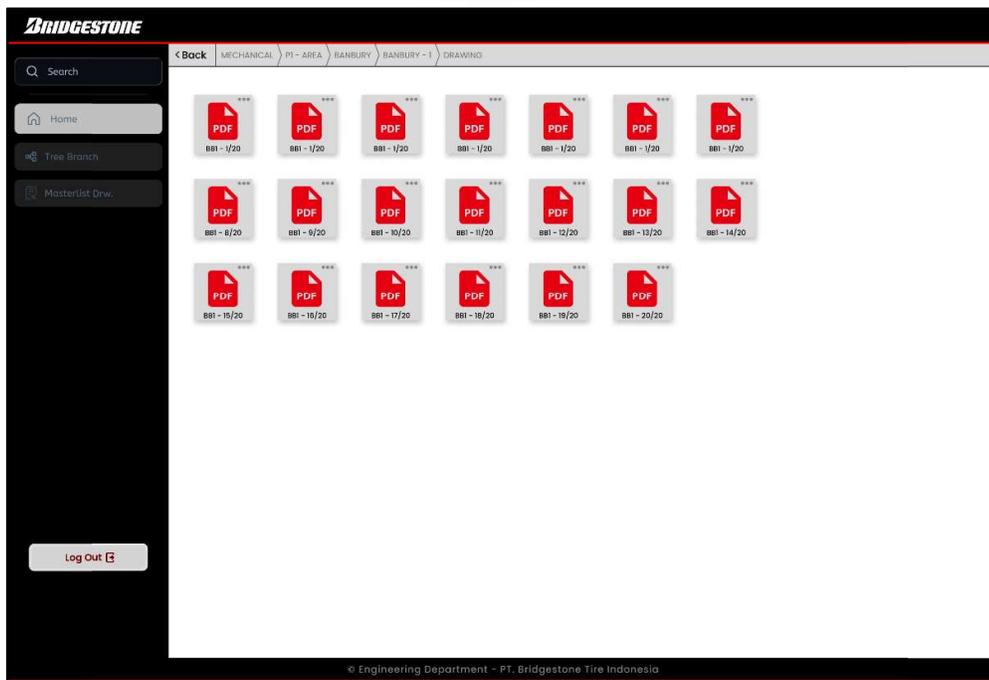
Gambar 3. 14 Folder P1-AREA Page



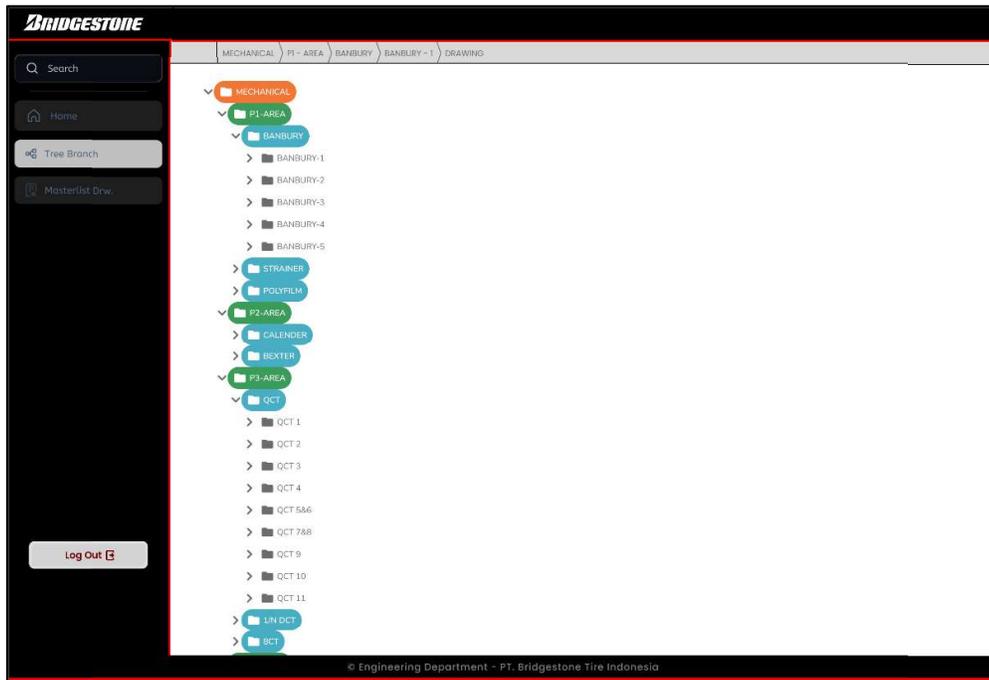
Gambar 3. 15 Folder Banbury Page



Gambar 3. 16 Folder Banbury-1 Page



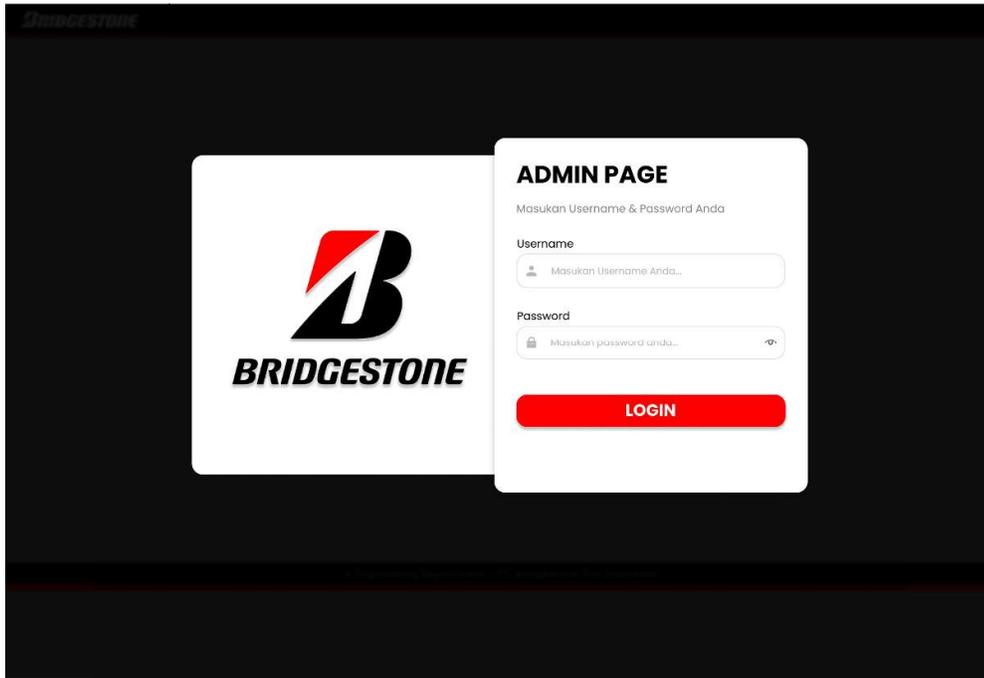
Gambar 3. 17 Folder Drawing Page



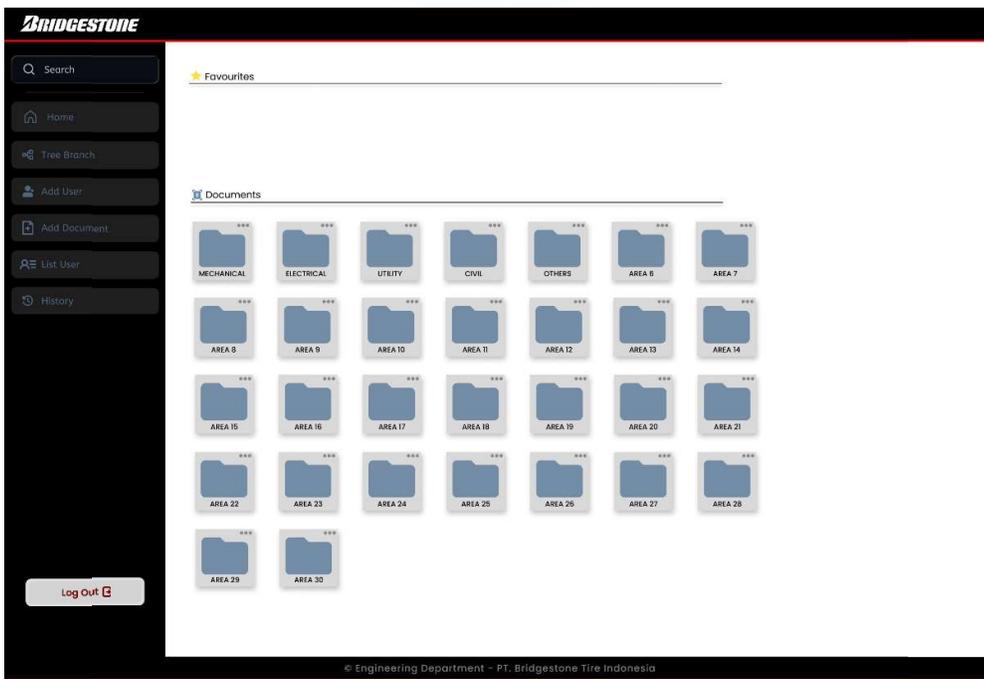
Gambar 3. 18 Tree View (Tree Branch) Page

Dilakukan juga perancangan desain antarmuka (UI) untuk *role admin*. Gambar 3.19 sampai dengan gambar 3.31 merupakan desain antarmuka (UI) awal dari *role admin*.

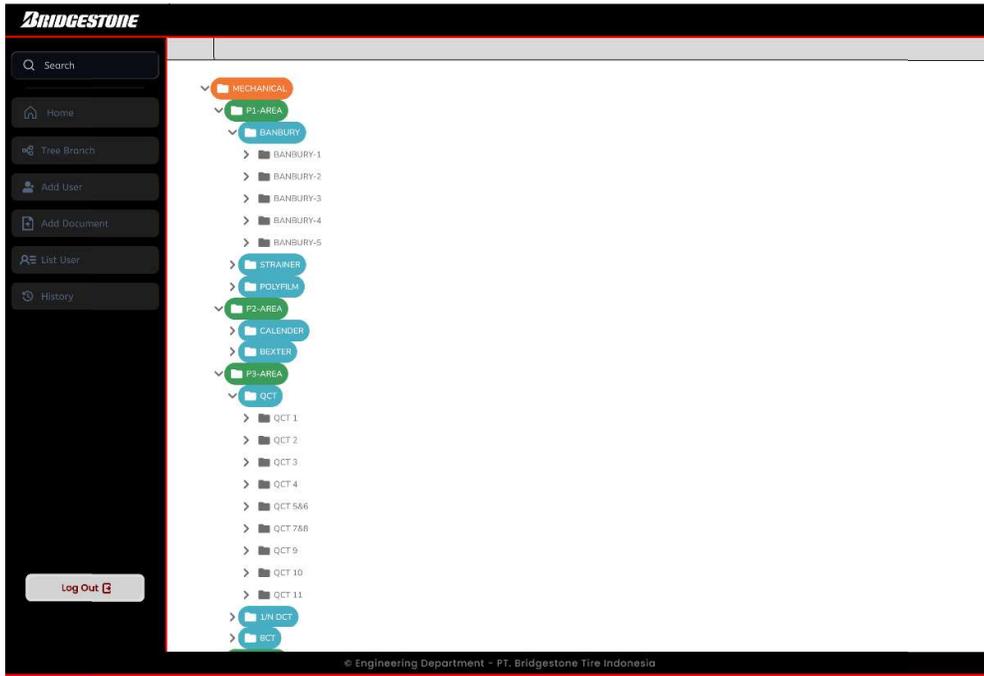




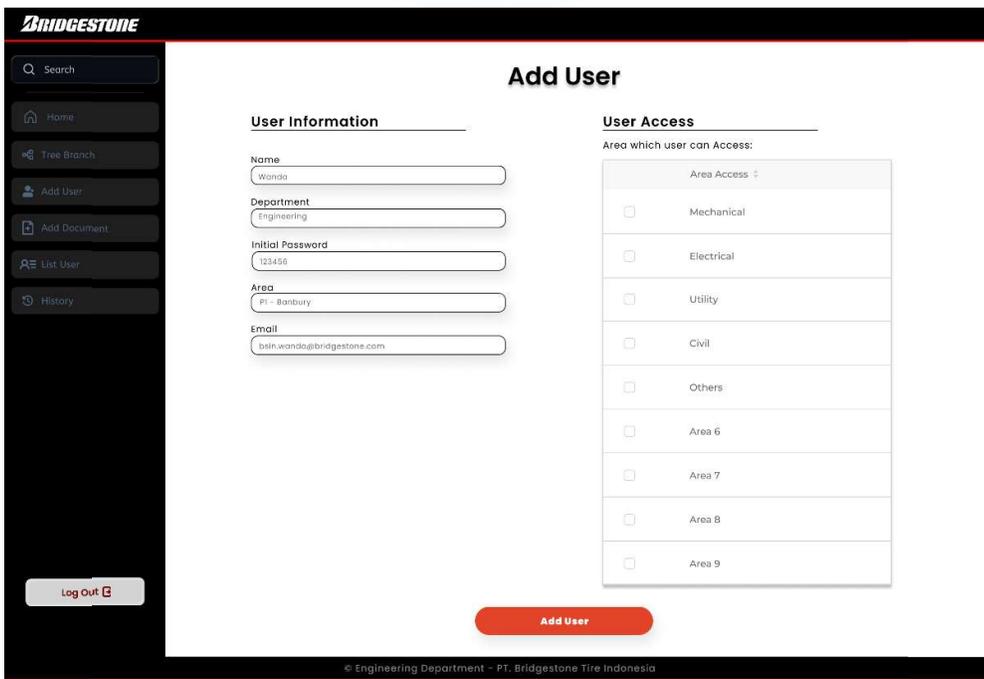
Gambar 3. 19 Login Page Admin



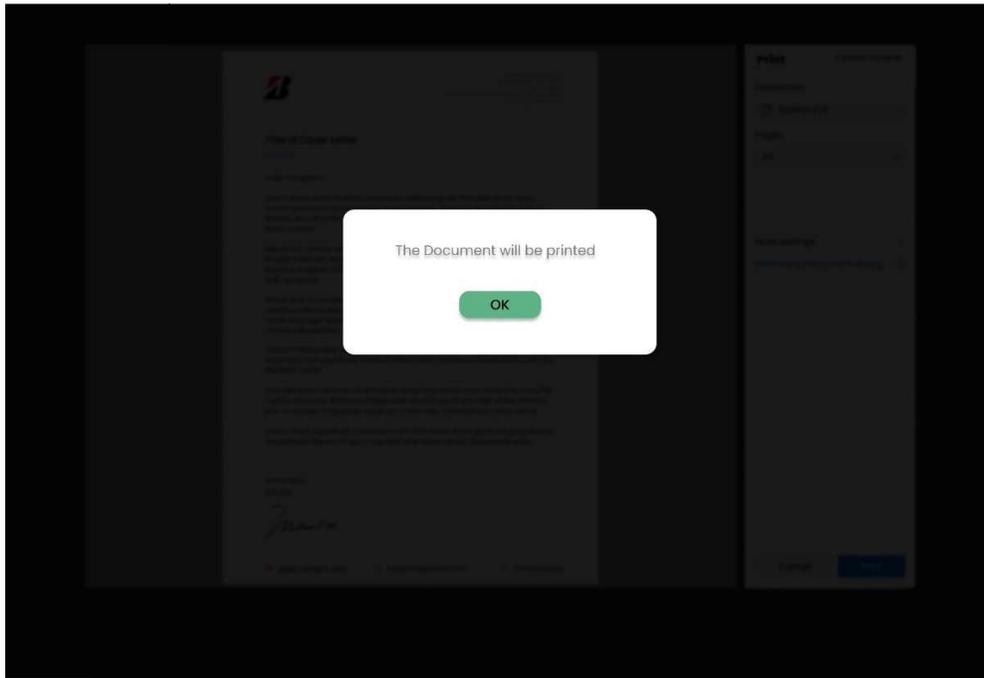
Gambar 3. 20 Home Page Admin



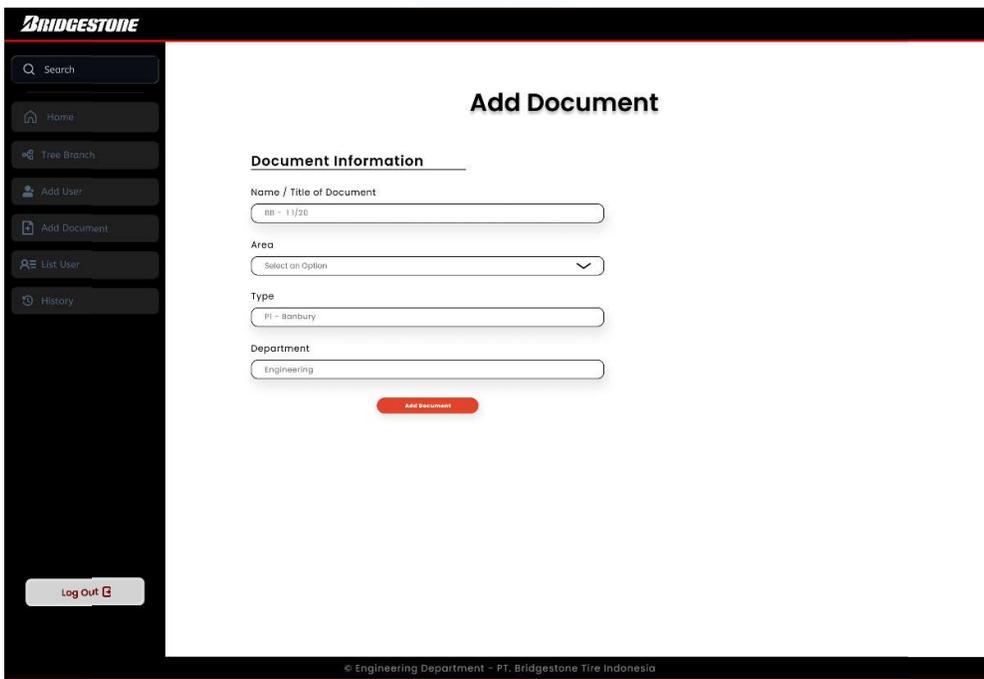
Gambar 3. 21 Tree View (Tree Branch) Admin Page



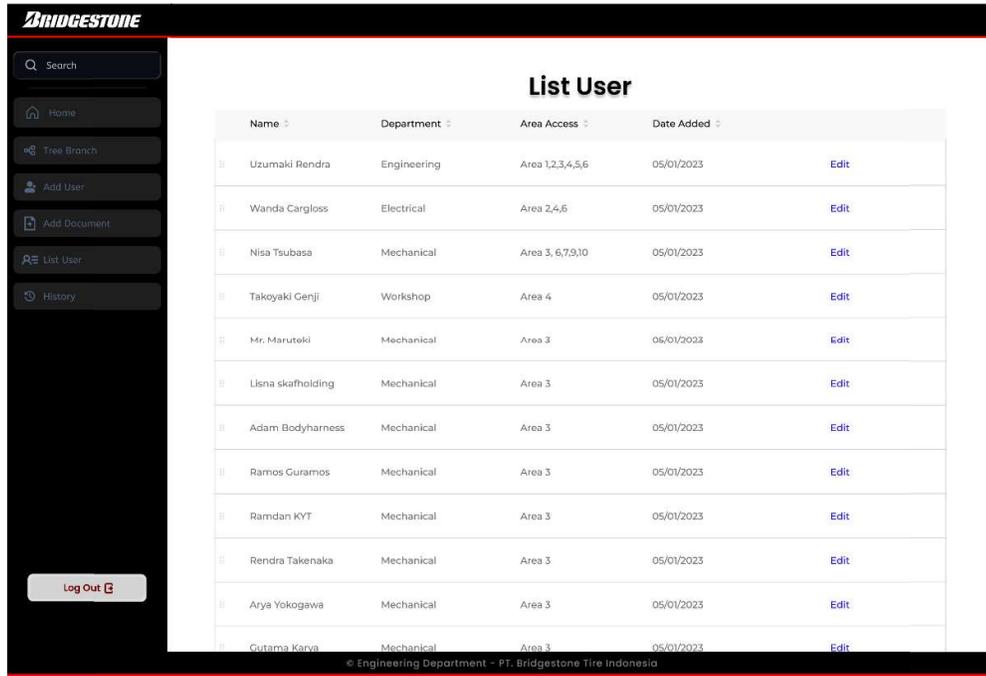
Gambar 3. 22 Add User Page



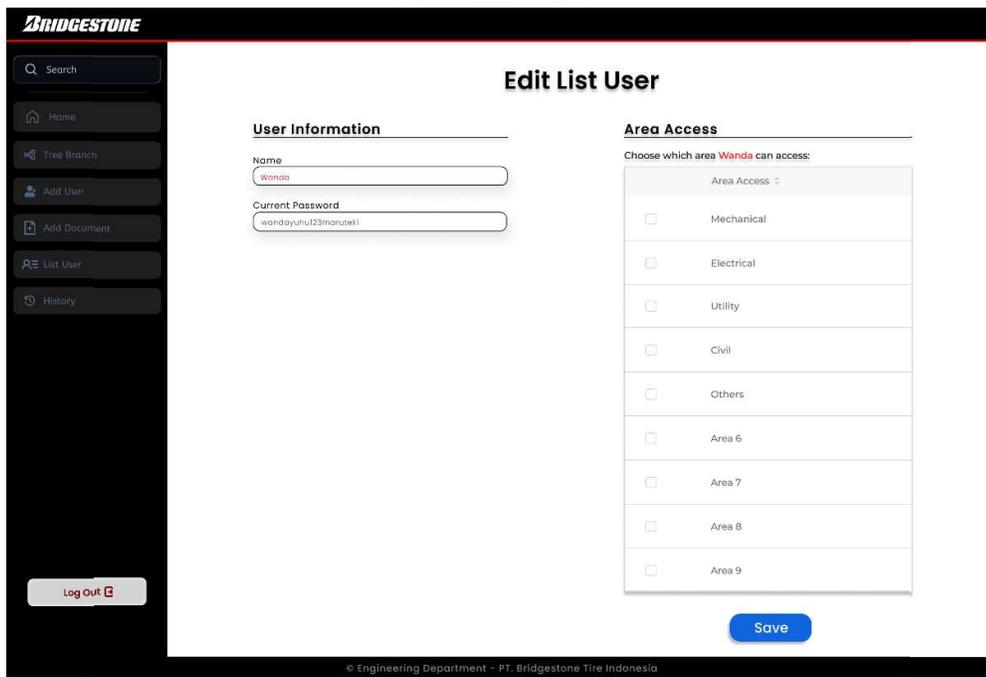
Gambar 3. 23 Document Printed Modal



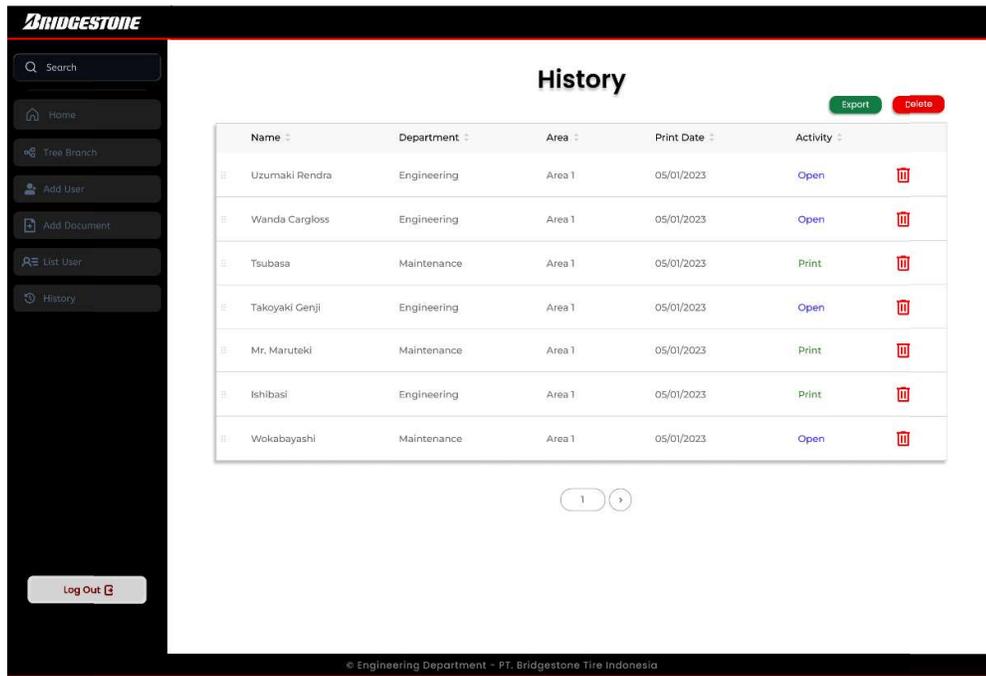
Gambar 3. 24 Add Document Page



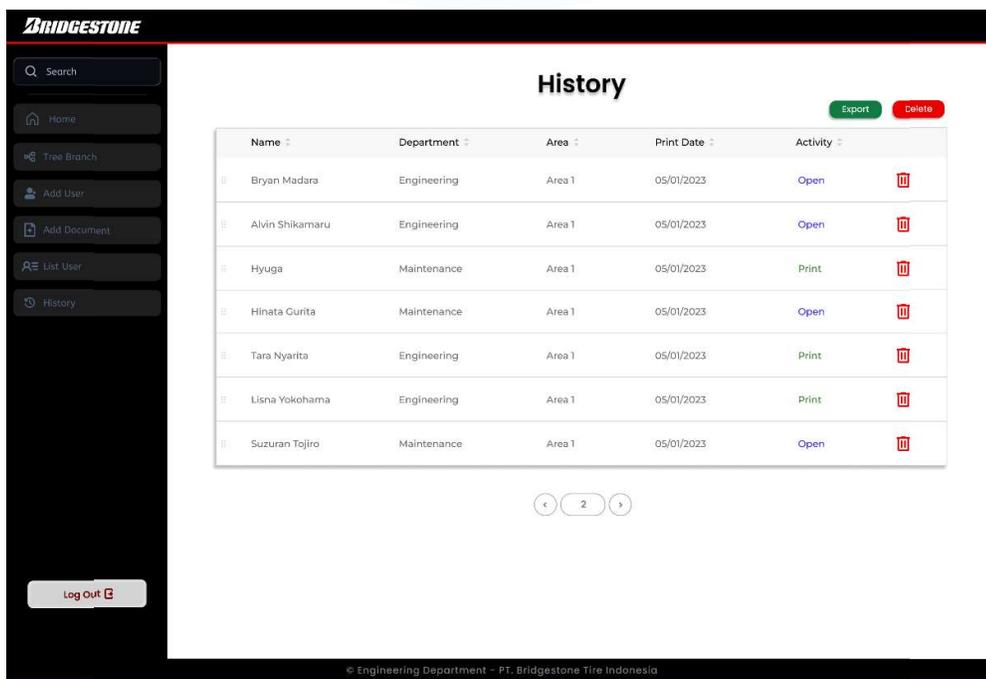
Gambar 3. 25 List User Page



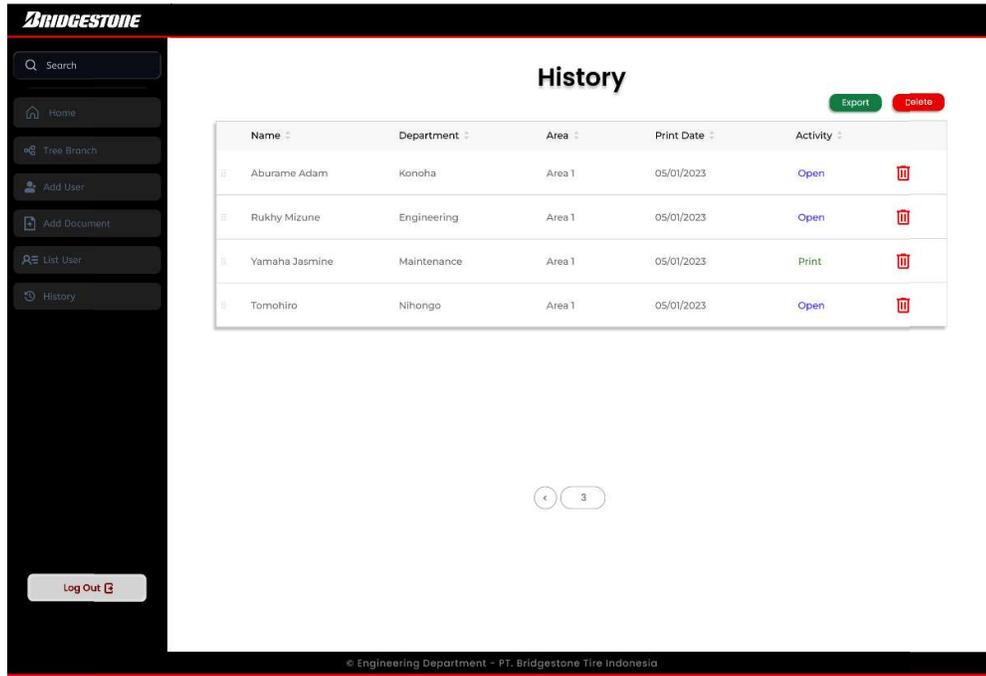
Gambar 3. 26 Edit List User Page



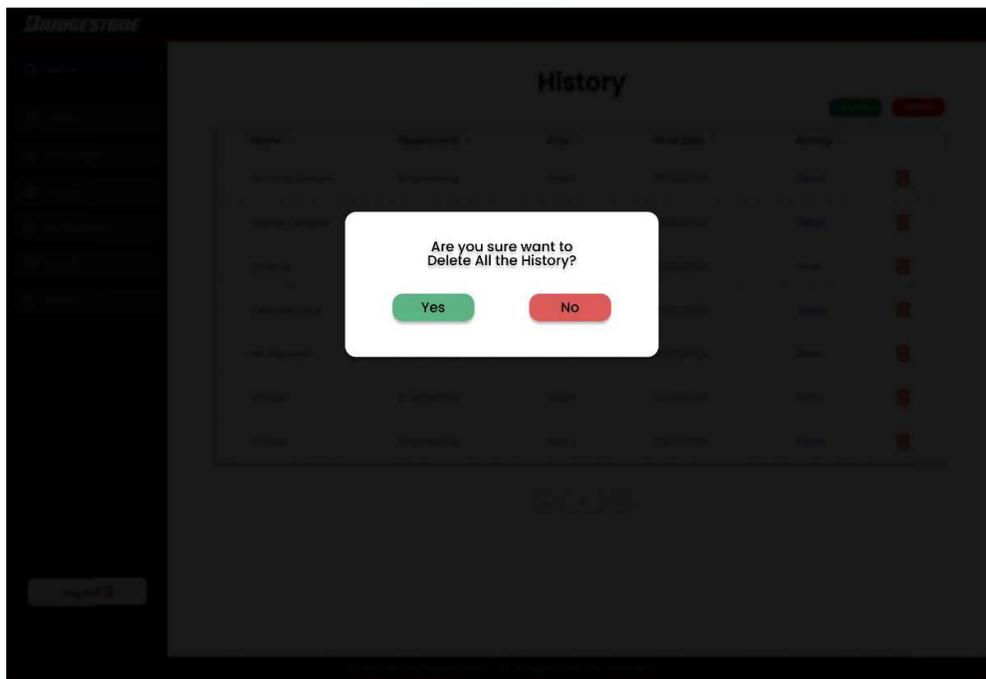
Gambar 3. 27 History Page | Page 1



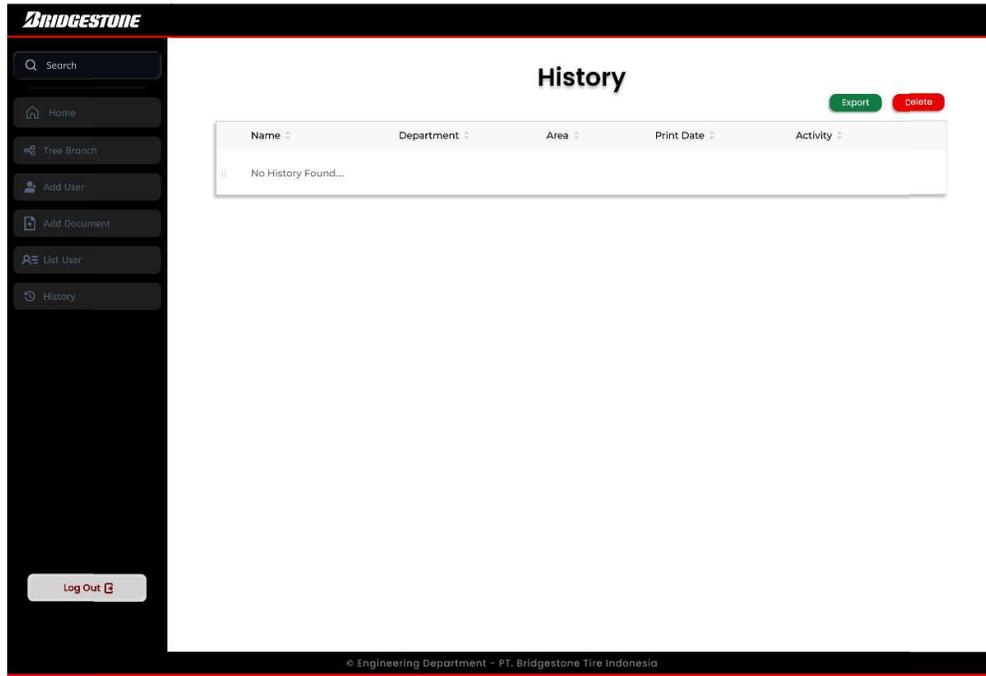
Gambar 3. 28 History Page | Page 2



Gambar 3. 29 History Page | Page 3



Gambar 3. 30 Delete All History Modal Confirmation



Gambar 3. 31 No History Page

3.2.17 Melakukan Perancangan website

Dalam pelaksanaan kerja magang, setelah melakukan perancangan pada antarmuka pengguna (UI), mahasiswa melakukan perancangan pada *website* menggunakan *framework* bootstrap dan *software* visual studio code.

```

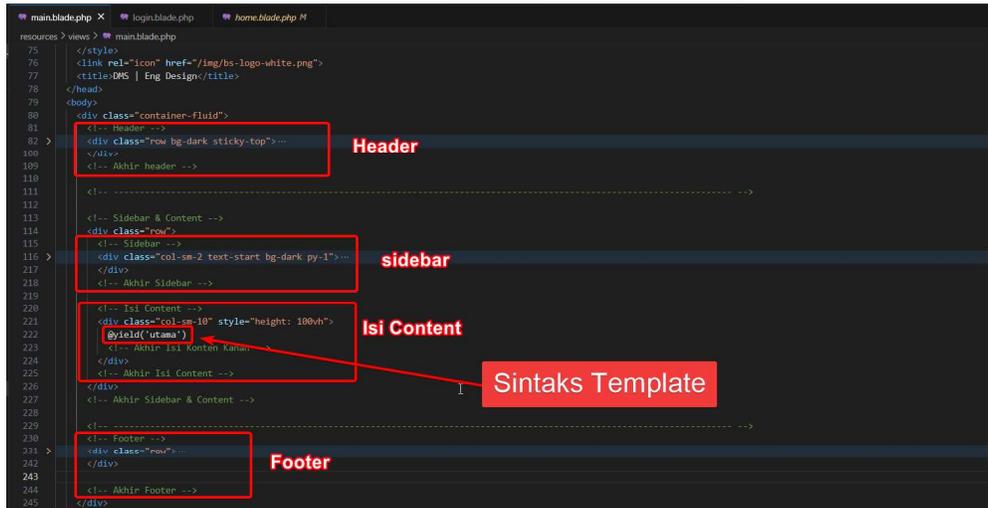
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <!-- Required meta tags -->
5 <meta charset="utf-8" />
6 <meta name="viewport" content="width=device-width, initial-scale=1" />
7 <meta name="csrf token" content="{{ csrf_token }}" />
8
9 {{-- DataTable --}}
10 <!-- <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"/>
11 <link href="https://cdn.datatables.net/1.10.21/css/jquery.dataTables.min.css" rel="stylesheet">
12 <link href="https://cdn.datatables.net/1.10.21/css/dataTables.bootstrap4.min.css" rel="stylesheet"> --}}
13 <link rel="stylesheet" href="{{ asset('vendor/DataTables/DataTables-1.13.4/css/jquery.dataTables.min.css') }}" />
14 <link rel="stylesheet" href="{{ asset('vendor/DataTables/DataTables-1.13.4/css/dataTables.bootstrap5.min.css') }}" />
15
16 <!-- Bootstrap CSS -->
17 <link rel="stylesheet" href="{{ asset('assets/bootstrap/css/bootstrap.min.css') }}" />
18
19 <!-- My CSS -->
20 <link rel="stylesheet" href="css/home.css" />
21
22
23 <!-- Link Bootstrap Icons -->
24 <script src="{{ asset('assets/bootstrap/js/bootstrap.min.js') }}"></script>
25 <link rel="stylesheet" href="{{ asset('assets/bootstrap/bootstrap-icons-1.10.5/font/bootstrap-icons.css') }}" />
26
27 <!-- Link CDN Buat Dropzone -->
28 <script src="{{ asset('vendor/dropzone/dist/dropzone-min.js') }}"></script>
29 <link href="{{ asset('vendor/dropzone/dist/dropzone.css') }}" rel="stylesheet" type="text/css" />
30

```

Gambar 3. 32 Penggunaan Framework Bootstrap

Link dan *script* diatas ditulis agar komponen yang disediakan *framework* bootstrap dapat termuat (ter-load) pada *website* yang dibuat. Framework bootstrap

diinstal di *local storage*. Hal ini dikarenakan *deployment website* dilakukan pada server lokal atau jaringan lokal.



```
75 </style>
76 <link rel="icon" href="/img/bs-logo-white.png">
77 <title>DMS | Eng Design</title>
78 </head>
79 <body>
80 <div class="container-fluid">
81 <!-- Header -->
82 <div class="row bg-dark sticky-top">...
100 </div>
109 <!-- Akhir header -->
110
111
112
113 <!-- Sidebar & Content -->
114 <div class="row">
115 <!-- Sidebar -->
116 <div class="col-sm-2 text-start bg-dark py-1">...
217 </div>
218 <!-- Akhir Sidebar -->
219
220 <!-- Isi Content -->
221 <div class="col-sm-10" style="height: 100vh">
222 @yield('utama')
223 <!-- Akhir Isi Konten Kanan -->
224 </div>
225 <!-- Akhir Isi Content -->
226 </div>
227 <!-- Akhir Sidebar & Content -->
228
229
230 <!-- Footer -->
231 <div class="row">...
242 </div>
243 <!-- Akhir Footer -->
244
245 </div>
```

Gambar 3. 33 Penggunaan Template Inheritance

Sintaks “@yield('utama')” digunakan dalam *framework* Laravel untuk menentukan suatu area di *template* yang akan diisi dengan konten dari bagian lain.

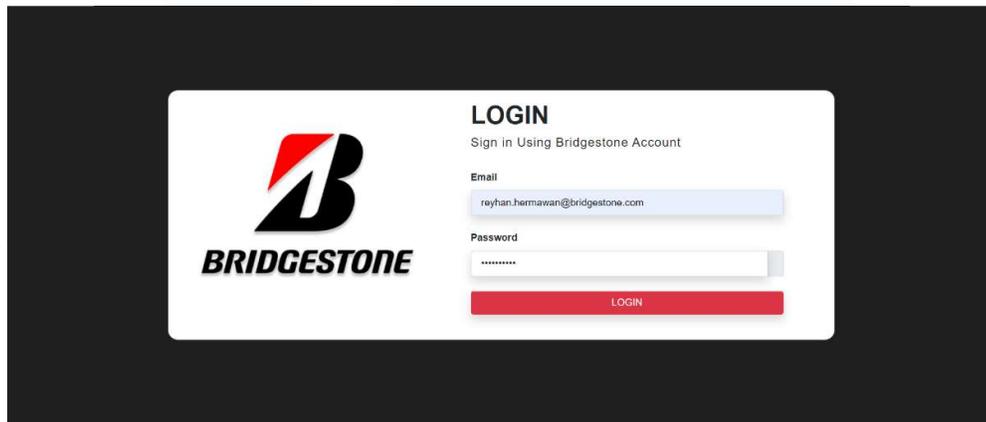
Dalam pelaksanaannya, mahasiswa membagi tampilan menjadi beberapa bagian yang terpisah seperti *header*, *sidebar*, *footer*, dan konten utama. “@yield('utama')” adalah contoh dari *directive Blade* yang digunakan untuk menunjukkan area di *template* yang akan diisi dengan konten dari halaman spesifik. Pada kasus ini, *template* utamanya adalah file “main.blade.php”.



```
resources > views > home.blade.php
1 @extends('main')
2
3 @section('utama')
4 @php
5 $userAccess = explode(',', auth()->user()->user_access);
6 $role = auth()->user()->role == "admin"
7 @endphp
8 <div class="overflow-auto" style="max-width: 100%; max-height: 100%">
9 <div class="container-fluid m-0 p-0">
10 <!-- Baris 3: Document Text -->
11 <div class="row my-3 ms-4">
12 <h4><i class="bi bi-folder-fill me-2 text-primary"></i>Documents</h4>
13 <div class="row">
14 | <hr class="ms-0" />
15 </div>
16 </div>
```

Gambar 3. 34 Halaman Spesifik (home.blade.php)

Pada halaman-halaman spesifik, mahasiswa menggunakan “@section” untuk menentukan konten yang akan diisi dalam “@yield('utama)”. Dalam hal ini, konten yang didefinisikan di dalam “@section('utama)” pada *file* "home.blade.php" akan diisi di dalam “@yield('utama)” pada *template* utama.

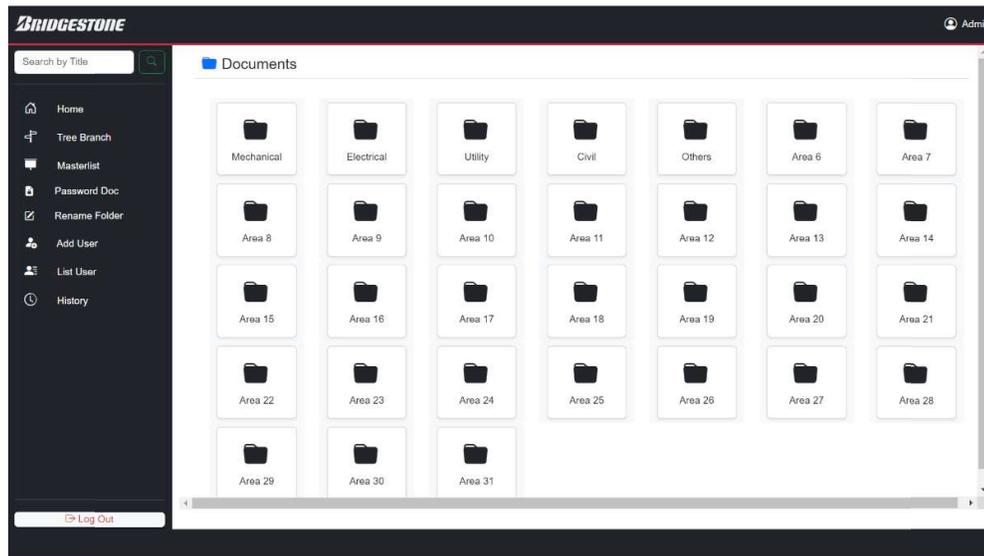


Gambar 3. 35 Login page

```
login.blade.php X
resources > views > auth > login.blade.php
34 <body>
35 <!-- Box Login-->
36
37 <section id="logincard">
38 <section class="vh-100" style="background-color: #000000e0">
39 <div class="container py-5 h-100">
40 <div class="row d flex justify-content center align-items center h 100">
41 <div class="col col-xl-10">
42 <div class="card" style="border-radius: 1rem">
43 <div class="row g-0">
44 <div class="col-md-6 col-lg-5 d-none d-md-block mt-4 pt-4">
45 
48 </div>
49 <div class="col-md-6 col-lg-6 d-flex align-items-center ms-4">
50 <div class="card-body text-black">
51 <form action="{{ 'postlogin' }}" method="post">...
52 </form>
53 </div>
54 </div>
55 </div>
56 </div>
57 </div>
58 </div>
59 </div>
60 </div>
61 </div>
62 </div>
63 </div>
64 </div>
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 </div>
82 </div>
83 </div>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 </div>
97 </div>
98 </div>
99 <!-- Akhir Box Login-->
100
```

Gambar 3. 36 Login Page (Code)

Pada code halaman login, dibuat *section* dengan *id* logincard sebagai penanda bahwa *section* tersebut merupakan card untuk melakukan login. Pada halaman login diberikan class *container* sebagai pembungkus card login (*Login Form* beserta gambar logo) dan diberikan baris (*row*) dan kolom (*column*) didalamnya. Hal ini dilakukan agar tampilan *website* dapat lebih responsif. Pada tampilan diatas, card dibagi menjadi 2 kolom, kolom pertama didalam class card tepatnya pada line 44, berisi gambar logo dari perusahaan pemegang. Sedangkan kolom kedua berisi *form* seperti *email*, *password* dan tombol *login*.



Gambar 3. 37 Home (Role Admin)

Pada tampilan *Home* Admin di gambar 3.37, terdapat beberapa menu tambahan seperti *Password document*, *rename folder*, *add user*, *list user* dan *history* pada bagian *sidebar*. Isi konten dari tampilan *home* admin terdapat beberapa folder yang dibuat dari *card* pada *code*-nya, tampilan *card* tersebut berfungsi sebagai folder yang telah disesuaikan dengan folder asli dari departemen *engineering design*.

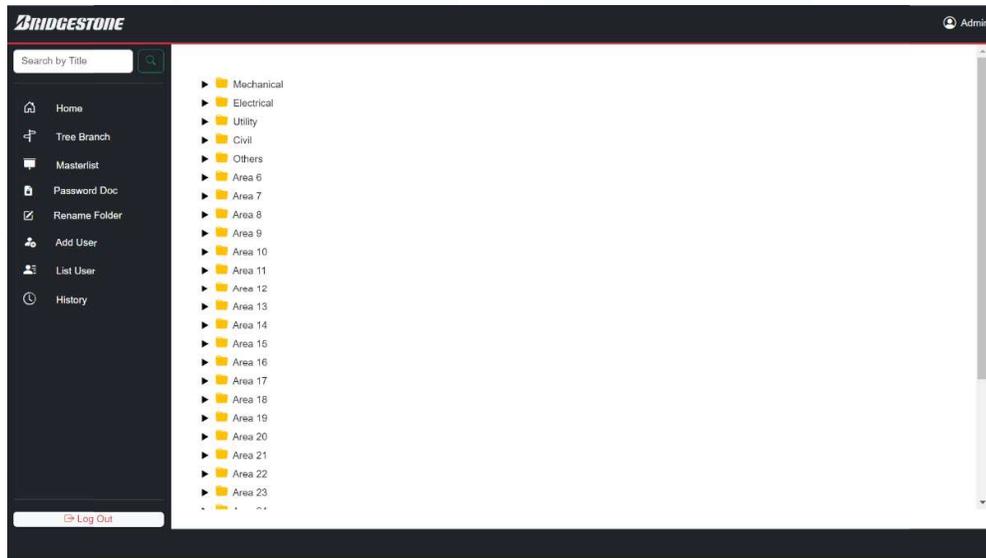
```

18 <!-- Baris 4 : Card Folder -->
19 {{!-- <div class="overflow-auto" style="max-width: 100%; max-height: 700px" --}}
20 <div class="container-fluid">
21 <div class="row my-3 ms-4 fs-5 row-cols-auto">
22 <!-- Card MECHANICAL -->
23 @if (in_array('mech', $userAccess) || $role)
24 <div class="col">
25 <a href="{{route('mech')}}">
26 <button type="button" class="btn btn-primary btn-light">
27 <div class="card shadow-sm" style="width: 8rem">
28 <div class="card-body text-center">
29 <i class="bi bi-folder-fill me-1 fs-1"></i>
30 @foreach ($folders as $folder)
31 @if ($folder->id === 3)
32 <span class="tt" data-bs-placement="top" title="{{ $folder->name }}">
33 <p class="card-text word-wrap text-truncate">{{ $folder->name }}</p>
34 </span>
35 @endif
36 @endforeach
37 </div>
38 </div>
39 </button>
40 </a>
41 </div>
42 @endif
43 <!-- Akhir Card MECHANICAL -->
44 <!-- Card ELECTRICAL -->

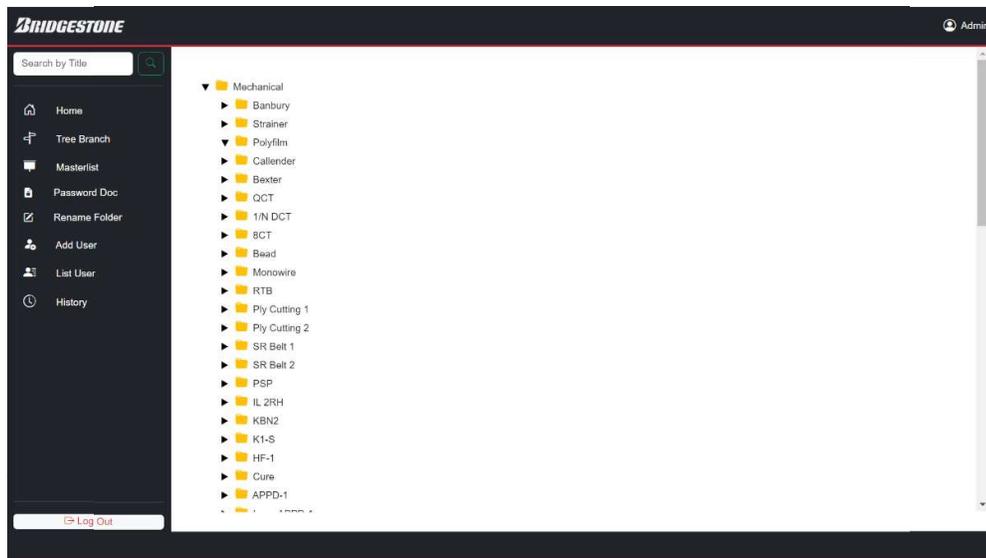
```

Gambar 3. 38 Contoh Card / Folder (Code)

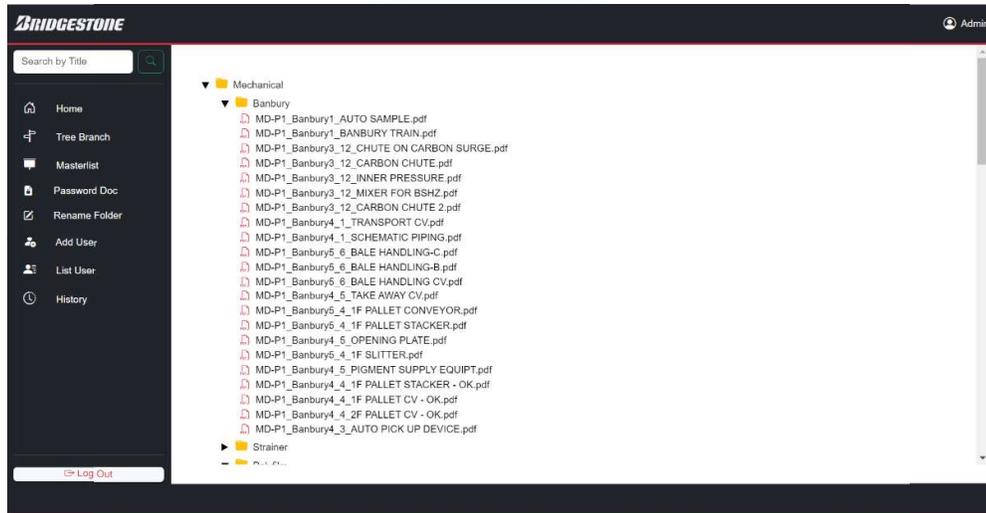
Seperti pada gambar 3.38, folder yang terbuat dari *card* berisi *icon* dan nama folder. *Card* dibungkus oleh *tag button* agar bisa dilakukan *action click* dan dilakukan *styling* sesuai rencana, dan *tag button* juga dibungkus oleh *tag <a>* untuk mengatur atau menentukan URL atau alamat setelah dilakukan klik pada *card*.



Gambar 3. 39 Tree view / Tree Branch (Role Admin)



Gambar 3. 40 Tree View / Tree Branch - Mechanical Folder (Role Admin)



Gambar 3. 41 Tree View / Tree Branch - Mechanical - Banbury Folder (Role Admin)

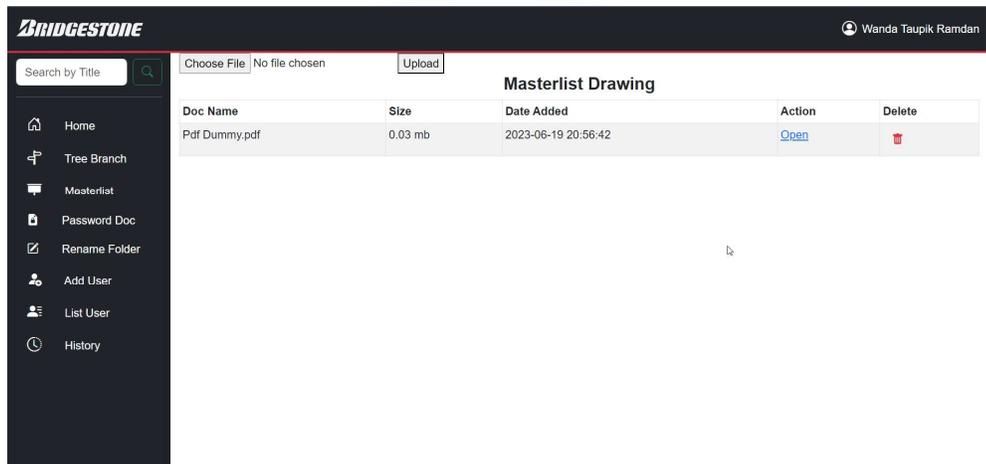
```

<!-- scroll -->
<div class="overflow-auto" style="max-width: 100%; max-height: 100%;>
<section class="container-fluid">
<div class="row content d-flex">
<div class="col-lg-10 py-4">
<div class="box p-4 rounded-3">
<ul id="myUL">
<!-- MECHANICAL -->
@if (in_array('mech', $userAccess) || $role)
<li>
@foreach ($folders as $folder) ...
@endforeach
<ul class="nested">
<!-- BANBURY -->
<li>
@foreach ($folders as $folder) ...
@endforeach
<ul class="nested">
@foreach ($documents as $document)
@if ($document->category == "banbury")
<li>
<span>
<a href="{route('view-tr', $document->id)}" target="_blank" style="text-decoration: none; color:black">
<i class="bi bi-filetype-pdf text-danger fs-6 me-1"></i>
{{ $document->doc_name }}</a>
</span>
</li>
@endif
@endforeach
@endforeach
<!-- AKHIR BANBURY -->
<li>
<!-- STRAINER -->
<li>
@foreach ($folders as $folder)
@if ($folder->id == 39)

```

Gambar 3. 42 Tree View / Tree Branch - Mechanical - Banbury Folder - Document (Code)

Dalam pelaksanaannya, perancangan halaman *tree view* menggunakan *unordered list* (``) dan *list* (``) untuk memperlihatkan visual pohon. Folder terluar merupakan folder list “Mechanical” dimana *list* ini berisi *list* lagi didalamnya yang berisi folder turunannya dan *file* dokumennya seperti yang dapat dilihat pada gambar 3.42.



Gambar 3. 43 Masterlist Drawing (Role Admin)

Tampilan *Masterlist drawing* merupakan tampilan tabel yang berisi dokumen panduan untuk menemukan daftar folder dan dokumen tertentu. Dokumen dapat ditambahkan dan dihapus oleh *role* admin.

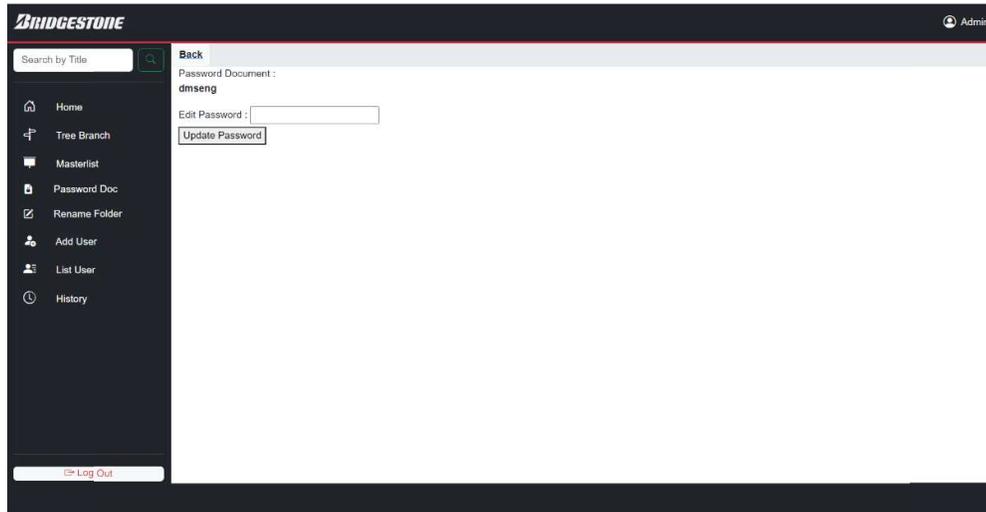
```

14 <h4 style="text-align: center;"><b>Masterlist Drawing</b></h4>
15 <table class="table table-sm table-bordered table-striped table-hover">
16 <thead>
17 <tr>
18 <th scope="col">Doc Name</th>
19 <th scope="col">Size</th>
20 <th scope="col">Date Added</th>
21 <th scope="col">Action</th>
22 @if (auth()->user()->role == "admin")
23 <th scope="col">Delete</th>
24 @endif
25 </tr>
26 </thead>
27 <tbody>
28 @foreach ($documents as $document)
29 @if ($document->category == "masterlist")
30 <tr>
31 <td>{{ $document->doc_name }}</td>
32 <td>{{ $document->size }} mb</td>
33 <td>{{ $document->created_at }}</td>
34 <td><a href="{{ route('masterlist.show', $document->id) }}" target="_blank">Open</a></td>
35 @if (auth()->user()->role == "admin")
36 <td><a href="{{ route('deleteML', ['id' => $document->id]) }}" type="button"
37 class="btn btn-link"><i class="bi bi-trash-fill text-danger"></a></td>
38 @endif
39 </tr>
40 @endif
41 @endforeach
42 </tbody>
43 </table>
44 @endaction

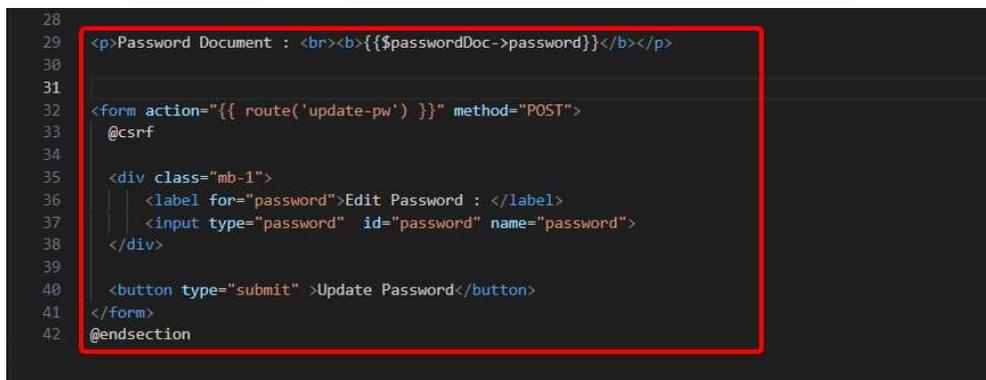
```

Gambar 3. 44 Tampilan Tabel Masterlist Drawing (Code)

Tampilan Tabel menggunakan *tag table* (<table>) untuk perancangannya. Tabel *Masterling drawing* terdiri dari 5 kolom utama seperti nama dokumen, ukuran dokumen (dalam Mb), *Action* (untuk membuka dan melihat dokumen) dan *Delete* (untuk menghapus dokumen masterlist).

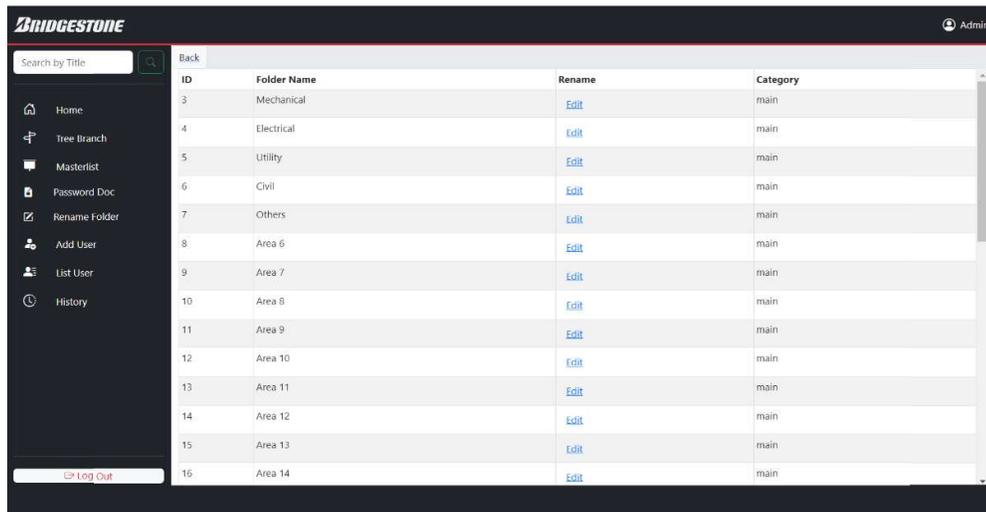


Gambar 3. 45 Password Document (Role Admin)



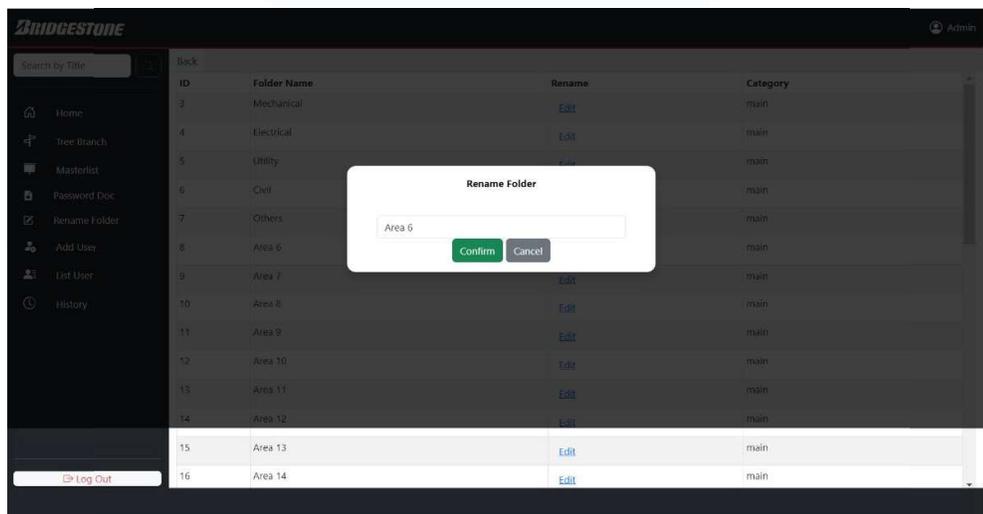
Gambar 3. 46 Password Document (Code)

Pada tampilan *password document*, hanya dibutuhkan keterangan *password* dan *form* yang dapat menginput *password* baru untuk dilakukan *update*. Terdapat tulisan menggunakan *tag p* (<p>) dan *tag label* (<label>), *form* untuk melakukan *input password* menggunakan *tag form* (<form>), dan tombol “Update Password” menggunakan *tag button* (<button>).



Gambar 3. 47 Rename Folder (Role Admin)

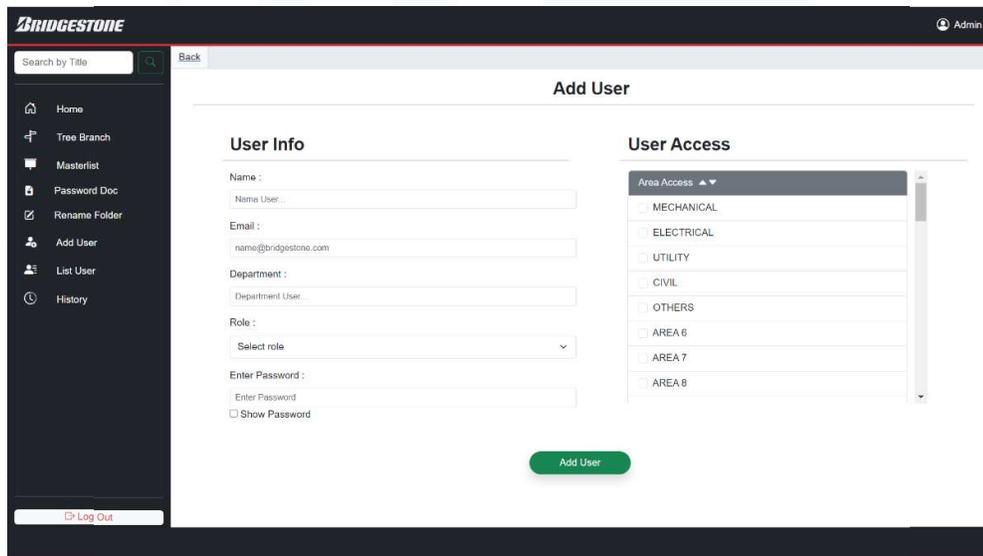
Diberlakukan tampilan tabel pada halaman *rename* folder. Tabel untuk melakukan perubahan nama folder ini dibuat menjadi 4 Kolom yakni ID, Folder name, *Rename* (untuk melakukan *edit* nama) dan *Category*.



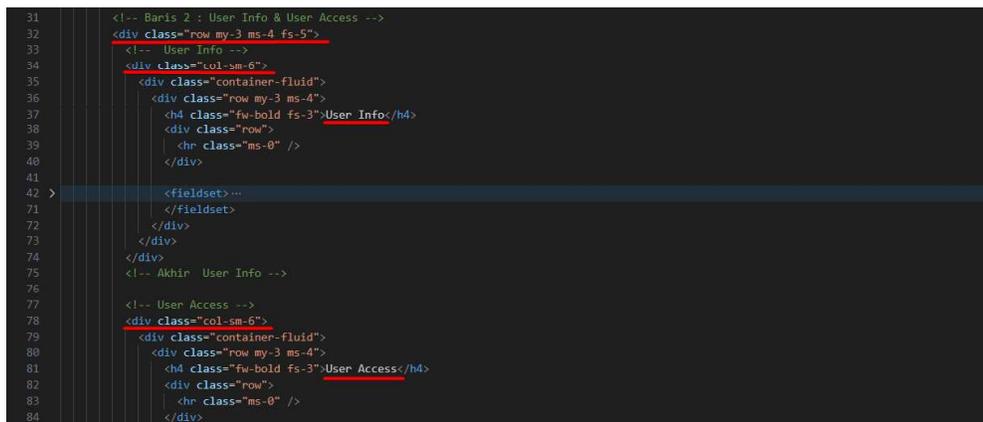
Gambar 3. 48 Contoh Card Rename Folder (Role Admin)

Saat dilakukan aksi klik pada tulisan *edit* pada gambar 3.47, akan termuat tampilan seperti pada gambar 3.48. Tampilan ini berupa *modal* yang menampilkan *card*. *Modal* adalah komponen yang disediakan oleh *framework* bootstrap untuk menampilkan konten tambahan, *pop-up*, atau jendela dialog yang tumpang tindih

dengan konten utama pada *website*. Terdapat *form input label* pada *modal* untuk mengganti nama folder, terdapat juga tombol “Cancel” dan tombol “Confirm” untuk melakukan konfirmasi mengenai perubahan nama folder.



Gambar 3. 49 Add User (Role Admin)



Gambar 3. 50 Add User (Code)

Tampilan *add user* dibagi menjadi 2 kolom sebagai pemisah antara “*User Info*” dengan “*User Access*” sepertinya yang dapat dilihat pada gambar 3.50.

```

31 <!-- Baris 2 : User Info & User Access -->
32 <div class="row my-3 ms-4 fs-5">
33 <div class="col-sm-6">
34 <div class="container-fluid">
35 <div class="row my-3 ms-4">
36 <div class="row my-3 ms-4">
37 <div class="fw-bold fs-3">User Info/</div>
38 <div class="row">
39 <hr class="ms-0" />
40 </div>
41
42
43 <fieldset>
44 <div class="mb-3 fs-6">
45 <label for="name" class="form-label">Name </label>
46 <input type="text" name="name" id="name" class="form-control form-control-sm" placeholder="Nama User..." />
47 </div>
48 <div class="mb-3 fs-6">
49 <label for="email" class="form-label">Email </label>
50 <input type="email" class="form-control form-control-sm" name="email" id="email" placeholder="name@bridgestone.com" />
51 </div>
52 <div class="mb-3 fs-6">
53 <label for="department" class="form-label">Department </label>
54 <input type="text" id="department" name="department" class="form-control form-control-sm" placeholder="Department User..." />
55 </div>
56 <div class="mb-3 fs-6">
57 <label for="role" class="form-label">Role </label>
58 <select class="form-select" name="role" id="role" aria-label="Default select example">
59 <option selected>Select role</option>
60 <option value="user">user</option>
61 <option value="admin">admin</option>
62 </select>
63 </div>
64 <div class="mb-3 fs-6">
65 <label for="password" class="form-label">Enter Password </label>
66 <input type="password" id="password" name="password" class="form-control form-control-sm" placeholder="Enter Password" />
67 <input type="text" name="password_visible" id="password_visible" class="form-control form-control-sm" style="display: none;" />
68 <label for="show_password">Show Password
69 <input type="checkbox" id="show_password"> Show Password
70 </label>
71 </div>
72 </fieldset>
73 </div>

```

Gambar 3. 51 Add User - User Info (Code)

Pada kolom pertama di tampilan *Add User*, yaitu “*User Info*”, digunakan *tag fieldset* untuk mengelompokkan beberapa *form* menjadi satu. Pada gambar 3.51, terdapat beberapa *form* seperti nama, *email*, departemen, *role* dan *password*. Digunakan *dropdown* dengan opsi *user* dan *admin* pada *form select role*.

The screenshot shows a web form with the following elements:

- An email input field containing "name@bridgestone.com".
- A "Department :" label followed by an input field containing "Department User...".
- A "Role :" label followed by a dropdown menu. The dropdown is open, showing "Select role" (highlighted in blue), "user", and "admin".
- An "Enter Password" label followed by a password input field and a "Show Password" checkbox.
- On the right side, there is a vertical list of radio buttons with labels "U", "C", "C", "A", "A", "A".

Gambar 3. 52 dropdown dengan opsi user dan admin pada form select role

```

85
86
87 <!-- scroll -->
88 <div class="overflow-auto" style="max-width: 500px; max-height: 380px">
89 <ul class="list-group fs-6">
90 <li class="list-group-item bg-secondary text-light sticky-top">Area Access<input class="bi bi-caret-up-fill ms-2"/><input class="bi bi-caret-down-fill"/></li></ul>
91 <li class="list-group-item">
92 <input class="form-check-input me-1" name="user_access[]" type="checkbox" value="mech"/>
93 MECHANICAL
94 </li>
95 <li class="list-group-item">
96 <input class="form-check-input me-1" name="user_access[]" type="checkbox" value="elect"/>
97 ELECTRICAL
98 </li>
99 <li class="list-group-item">
100 <input class="form-check-input me-1" name="user_access[]" type="checkbox" value="util"/>
101 UTILITY
102 </li>
103 </ul>

```

Gambar 3. 53 Add User - User Access (Code)

Pada kolom kedua dari tampilan *Add User*, yaitu “*User Access*”, ditampilkan *list* dari nama-nama dokumen yang dapat diakses menggunakan *tag unordered list* () dan *tag list* () dengan tipe *checkbox*. Kumpulan *list* ini dibungkus oleh *class* “*overflow-auto*” agar pengguna dapat melakukan *scrolling* pada kumpulan *list* yang menumpuk.

Name	Department	Role	Email	User Access	Delete	Edit
Wanda Taupik Ramdan	MIT Design	admin	wandataufik@gmail.com	mech		
Deni Kusmana	Adm Eng Design	admin	deni.kusmana@bridgestone.com	mech,elect,util,civil,oth		
Husna Mubarak	MIT Design	admin	husna.mubarak@bridgestone.com	mech,elect,util,civil,oth		
Khoirunisa	Adm Eng Design	user	khoirunisa@gmail.com	mech,elect		
Zaenal Arifin	Engineering	admin	zaenal.arifin@bridgestone.com			
Junardi	Engineering	admin	bsin.junardi@bridgestone.com	mech,elect,util,civil		
Asep Suhyana	Engineering	admin	asep.suhyana@bridgestone.com	mech		
User	Engineering	user	user@gmail.com	mech,elect,util,civil,oth		
Admin	Engineering	admin	administrator@gmail.com			

Gambar 3. 54 List User (Role Admin)

```

resources > views > admin > listuser.blade.php
24      {{ session('success1') }}
25    </div>
26  @endif
27
28  <!-- Baris 2: List User Text -->
29  <div class="row my-3 ms-4">
30    <!-- scroll -->
31    @if (session('success'))
32      <div class="alert alert-danger">
33        {{ session('success') }}
34      </div>
35    @endif
36    <div class="overflow-auto" style="max-width: 100%; max-height: 100%;>
37      <div class="table-responsive">
38        <table class="table table-sm table-bordered table-striped table-hover">
39          <caption>
40            List of users
41          </caption>
42          <thead>
43            <tr>
44              <th scope="col">Name</th>
45              <th scope="col">Department</th>
46              <th scope="col">Role</th>
47              <th scope="col">Email</th>
48              <!-- Created at -->
49              <th scope="col">Created at</th>
50              <th scope="col">User Access</th>
51              <th scope="col" style="text-align: center">Delete</th>
52              <th scope="col" style="text-align: center">Edit</th>
53            </tr>
54          </thead>
55          <tbody>

```

Gambar 3. 55 List User (Code)

Tampilan *List User* menggunakan tampilan tabel seperti yang dapat dilihat pada gambar 3.55. Digunakan juga class “*overflow-auto*” yang disediakan *framework* bootstrap untuk dapat melakukan *scrolling* jika kumpulan *list user* bertumpuk sampai kebawah.

Description	Dokumen	Waktu	Action
Admin (Opened)	["MD-P1_Banbury1_AUTO SAMPLE.pdf"]	2023-06-14T00:36:21.000000Z	Hapus
Admin (Opened)	["Drawing Cure 3-26.pdf"]	2023-06-14T17:15:45.000000Z	Hapus
Admin (Opened)	["MD-P1_Banbury1_BANBURY TRAIN.pdf"]	2023-06-14T17:17:26.000000Z	Hapus
Admin (Opened)	["Curing Line 5 and renewal PLC AB.pdf"]	2023-06-15T17:38:06.000000Z	Hapus
Admin (View)	["MD-P1_Banbury1_AUTO SAMPLE.pdf"]	2023-06-14T00:47:29.000000Z	Hapus
Admin (View)	["Drawing Cur LT-XI LTXL OK.PDF"]	2023-06-14T17:14:29.000000Z	Hapus
Admin (View)	["Drawing Cur 4-17.pdf"]	2023-06-14T17:14:47.000000Z	Hapus
Admin (View)	["DRAWING CURING NRM YOKOGAWA.pdf"]	2023-06-14T17:15:10.000000Z	Hapus
Admin (View)	["Drawing Cure 3-26.pdf"]	2023-06-14T17:15:31.000000Z	Hapus
Admin (View)	["MD-P1_Banbury4_5_PIGMENT SUPPLY EQUIPT.pdf"]	2023-06-14T17:16:21.000000Z	Hapus

Gambar 3. 56 History (Role Admin)

```

88 <!-- Baris 3: Table History -->
89 <div class="row my-3 ms-4">
90 <!-- scroll -->
91 <div class="overflow-auto" style="max-width: 1200px; max-height: 500px">
92 <div class="table-responsive">
93 <table class="table table-sm table-bordered table-striped table-hover">
94 <caption>
95 List of History
96 </caption>
97 <thead>
98 <tr>
99 <th scope="col">#</th>
100 <th scope="col">File Name</th>
101 <th scope="col">Name</th>
102 <th scope="col">Department</th>
103 <th scope="col">Print Date</th>
104 <th scope="col">Activity</th>
105 <th scope="col">...</th>
106 </tr>
107 </thead>
108 <tbody>

```

Gambar 3. 57 Tampilan Tabel History (Code)

Tampilan pada *page history* menggunakan tampilan tabel untuk mengetahui aktivitas dari pengguna. Admin dapat melihat apakah pengguna melakukan “view” atau *download / print* (“opened”).

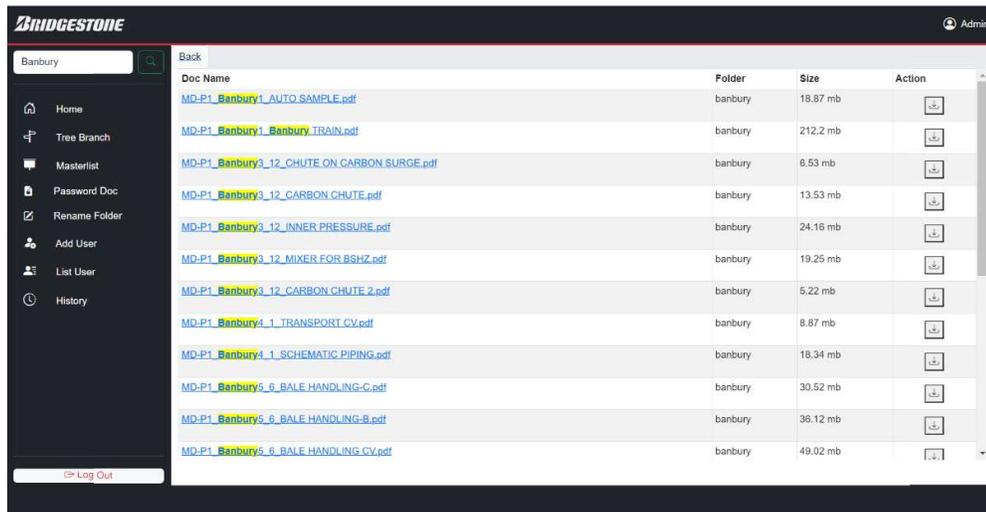
```

142
143 <!-- Baris 4: Pagination -->
144 <div class="row">
145 <div class="col">
146 <nav aria-label="Page navigation example">
147 <ul class="pagination justify-content-center">
148 <li class="page-item">
149 <a class="page-link" href="#" aria-label="Previous">
150 <span aria-hidden="true">&laquo;</span>
151 </a>
152 </li>
153 <li class="page-item"><a class="page-link" href="#">1</a></li>
154 <li class="page-item"><a class="page-link" href="#">2</a></li>
155 <li class="page-item"><a class="page-link" href="#">3</a></li>
156 <li class="page-item">
157 <a class="page-link" href="#" aria-label="Next">
158 <span aria-hidden="true">&raquo;</span>
159 </a>
160 </li>
161 </ul>
162 </nav>
163 </div>
164 </div>
165 <!-- Akhir Baris 4 -->
166 </div>
167 <!-- Akhir Isi Konten Kanan -->
168 </div>

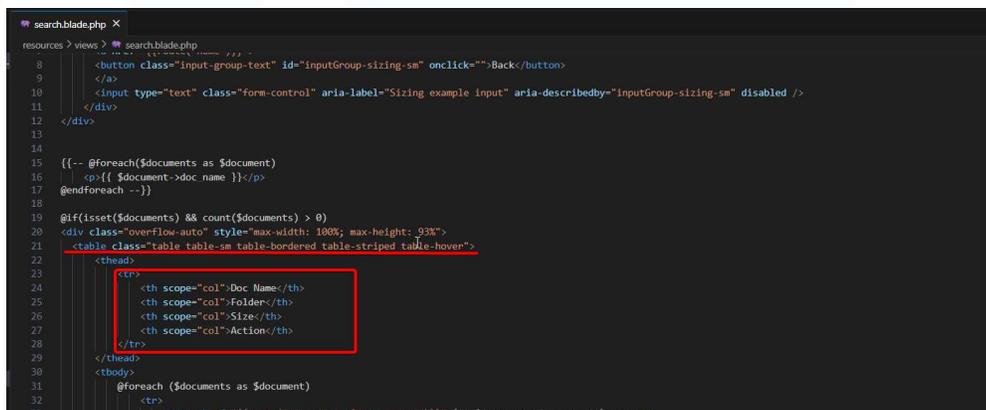
```

Gambar 3. 58 Tampilan Pagination pada page History (Code)

Pada bagian bawah *page history*, terdapat *pagination* yang dapat melakukan pindah halaman. Tampilan *pagination* ini dibuat menggunakan komponen yang telah disediakan oleh *framework bootstrap* yaitu dengan menambahkan “pagination” pada *class* dengan *tag ul* ().

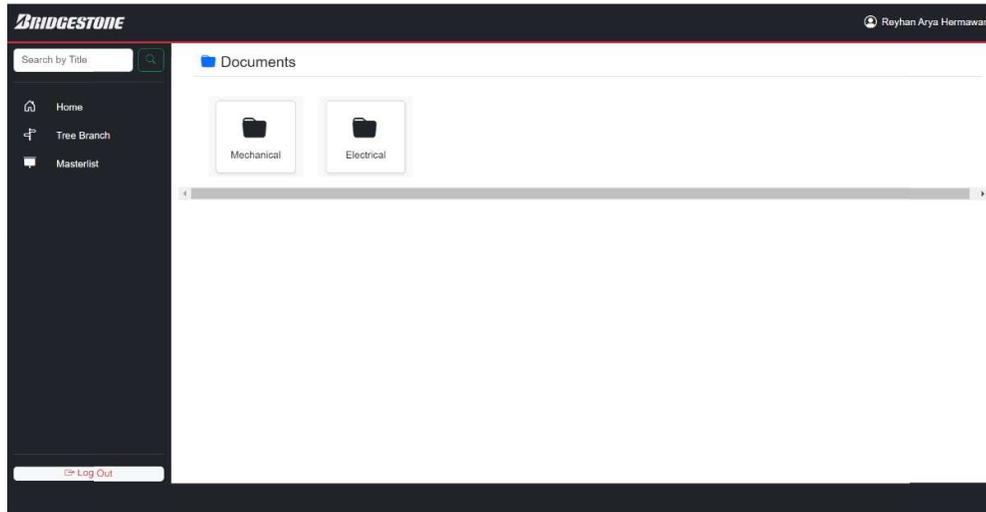


Gambar 3. 59 Fitur Search by Title

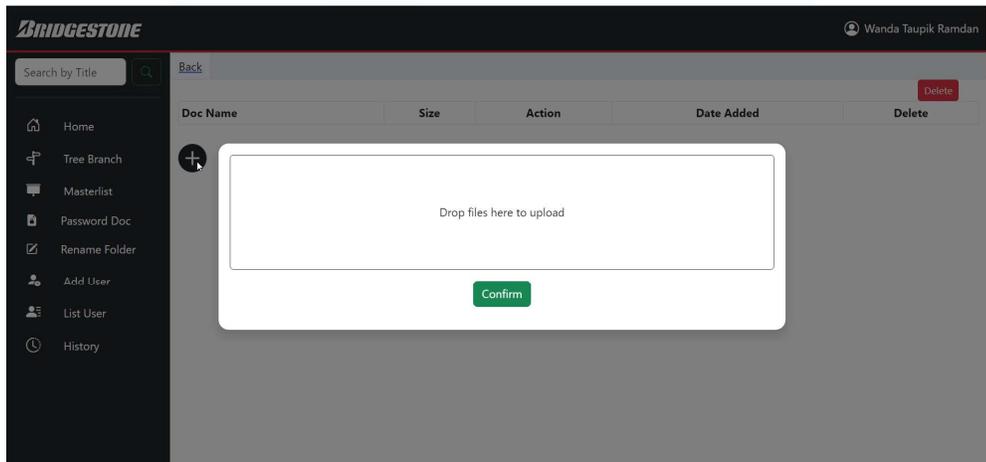


Gambar 3. 60 Fitur Search by Title (Code)

Tampilan tabel diimplementasikan kedalam fitur “*Search by Title*”. Saat melakukan pencarian dokumen, ditampilkan tabel yang memuat 4 kolom yakni, nama dokumen, folder alamat dokumen yang dicari, ukuran dokumen, dan *action* dimana pengguna dapat langsung membuka dokumen setelah melakukan pencarian.



Gambar 3. 61 Home (Role User)



Gambar 3. 62 Dropzone

```

126 <!-- Button Plus -->
127 @if (auth()->user()->role == "admin")
128 <div class="position-relative">
129 <!-- Button trigger modal -->
130
131 <a href="#" class="rounded" aria-current="page" data-bs-toggle="modal" data-bs-target="#exampleModalbtnplus">
132 <span class="text" data-bs-placement="top" title="Add Document">
133 <i class="bi bi-plus-circle-fill fs-1 text-dark"></i>
134 </span>
135 </a>
136
137 <!-- Button -->
138 <button class="btn btn-dark btn-lg btn-block bg-danger" type="button">Login/Buttons -->
139
140 <!-- Modal -->
141 <div class="modal fade" id="exampleModalbtnplus" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
142 <div class="modal-dialog modal-dialog-centered modal-lg">
143 <div class="modal-content shadow p-3" style="border-radius: 15px">
144
145 <div class="text-center">
146 <!-- Dropzone Baru -->
147 <form action="{{route('upload')}}" method="POST" enctype="multipart/form-data" id="pdf upload" class="dropzone">
148 @csrf
149 </form>
150 <!-- Akhir Dropzone baru -->
151
152 <a href="#">
153 <button type="button" class="btn btn-secondary btn-success mt-3" data-bs-dismiss="modal">Confirm/Buttons
154 </a>
155 </div>
156 </div>
157 </div>
158 </div>
159 </div>
160 @endif
161 <!-- Akhir Button Plus -->

```

Gambar 3. 63 Dropzone (Code)

Code diatas menjelaskan bahwa *dropzone* akan tertampil jika tombol “+” diklik, dimana aksi klik pada tombol “+” akan melakukan *trigger* pada *modal* yang berisi tampilan *dropzone*.

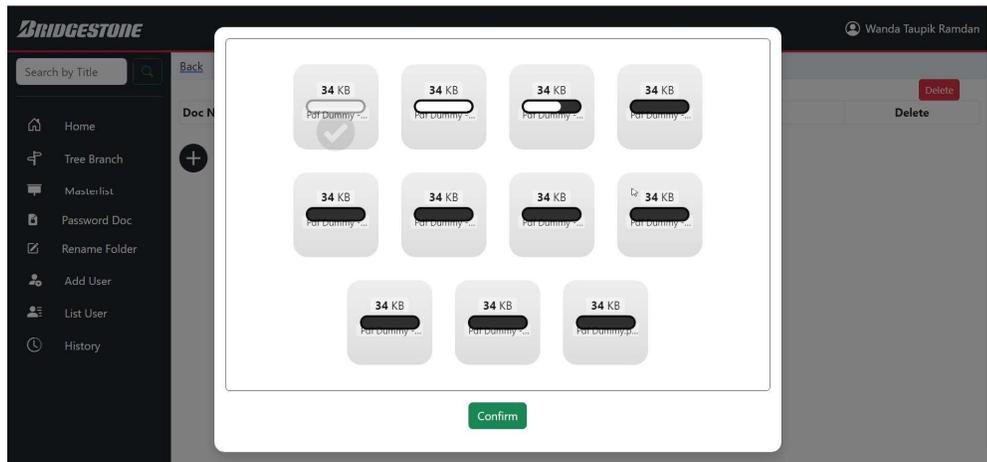
```

258 <script type="text/javascript">
259 new Dropzone('#pdf-upload', {
260   thumbnailWidth: 200,
261   maxFileSize: 1000,
262   acceptedFiles: '.jpeg,.jpg,.png,.pdf',
263 });
264 </script>
265
266 {{-- Dropzone --}}
267 <script src="../../node_modules/dropzone/dist/min/dropzone.min.js"></script>
268 <script src="../../assets/js/dropzone.js"></script>
269

```

Gambar 3. 64 Javascript Dropzone

Gambar 3.64 merupakan javascript untuk menampilkan dan mengatur *dropzone*. Javascript ini terdapat pada *file* utama yakni *file* “main.blade.php”.



Gambar 3. 65 Upload File pada Dropzone

File yang dapat diunggah pada *dropzone* dapat diatur pada javascript yang berada pada *file* “main.blade.php” seperti yang dapat dilihat pada gambar 3.64 tepatnya pada line 262 di bagian “*acceptedFiles*”.

3.2.18 Melakukan Website Testing

Sebelum dilanjutkan ke tahap *deployment* atau perilsan, dilakukan proses pengujian fungsionalitas, kinerja dan kualitas keseluruhan dari *website* DMS. Terdapat beberapa jenis pengujian yang dilakukan saat melakukan *website testing*:

1. *Functional Testing*

Tim DMS memastikan bahwa semua fungsi dan fitur yang ada pada *website* dapat berjalan sesuai yang diharapkan.

2. *Usability Testing*

Dilakukan juga pengujian ini untuk mengevaluasi pengalaman pengguna mengenai tata letak desain visual.



Gambar 3. 66 Button add ("+") dengan posisi dibawah

Tombol berbentuk bulat dengan tanda “+” yang ditandai kotak merah pada gambar 3.66 berada dibawah tabel. Hal tersebut dapat membuat keefektivitasan admin terhambat dikarenakan jika dokumennya sudah terlalu banyak menumpuk kebawah, maka tombol “+” akan tenggelam dan sulit untuk dijangkau



Gambar 3. 67 Button add ("+") dengan posisi diatas

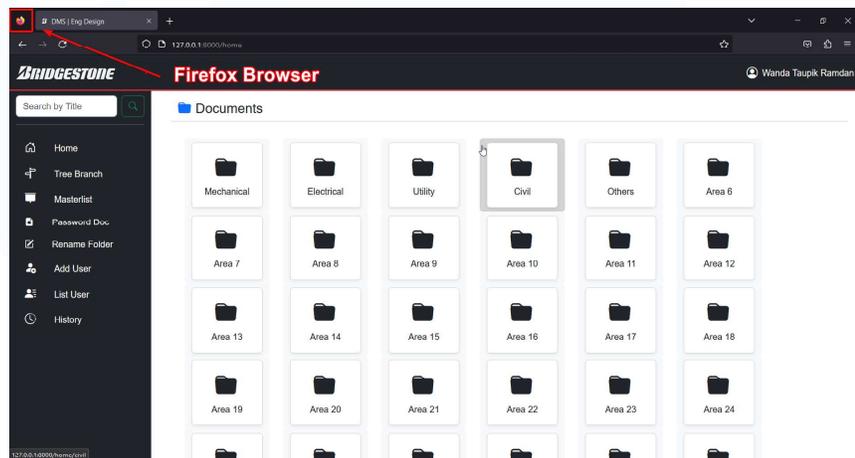
Dilakukan perbaikan mengenai tata letak visual dari tombol “+” yang berada pada posisi bawah (pada gambar 3.66) menjadi tombol yang berada pada posisi atas seperti pada gambar 3.67. Hal ini dilakukan untuk meningkatkan efektivitas admin saat ingin menambah dokumen. Admin tidak perlu melakukan scrolling kebawah saat ingin menambahkan dokumen seperti pada saat perbaikan belum dilakukan.

3. Performance Testing

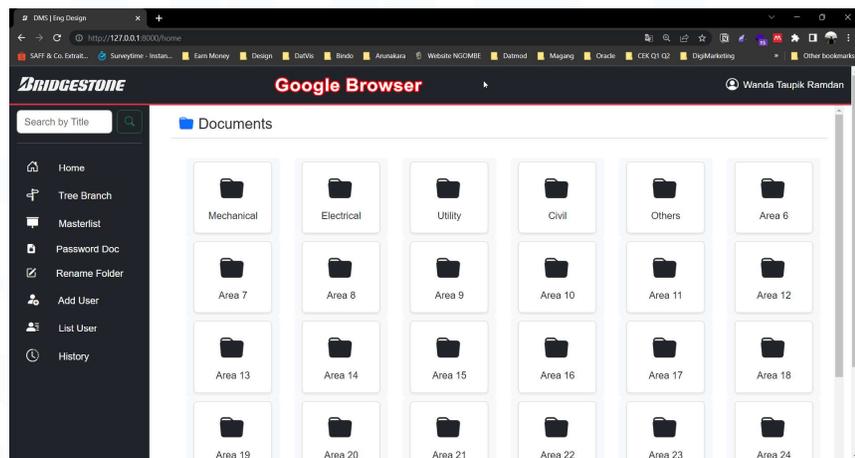
Dilakukan juga pengujian performa seperti waktu respons *website* dan kecepatan loading halaman

4. Compatibility Testing

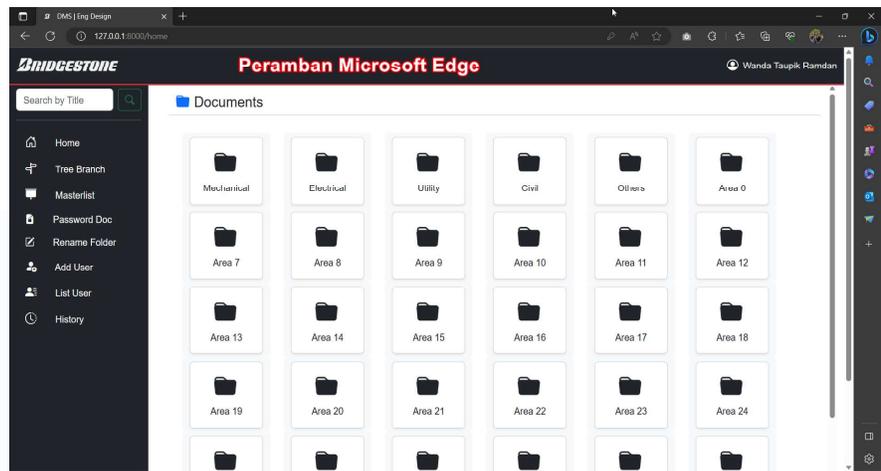
Pengujian ini dilakukan dengan cara memastikan *website* dapat berfungsi dengan baik di berbagai peramban atau *browser* serta perangkat yang berbeda.



Gambar 3. 68 Web DMS dijalankan pada peramban Firefox



Gambar 3. 69 Web DMS dijalankan pada peramban Google



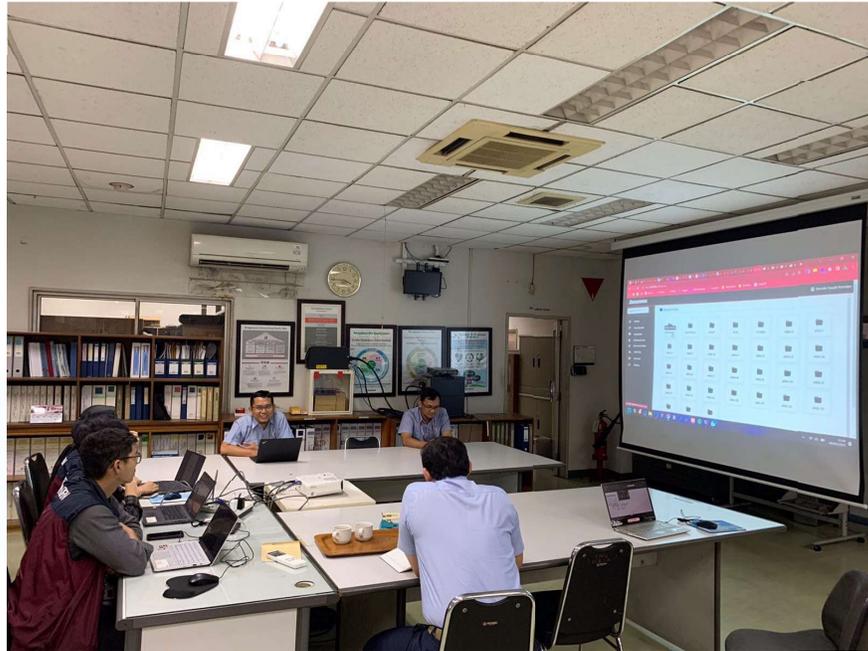
Gambar 3. 70 Web DMS dijalankan pada peramban Microsoft Edge

3.2.19 Melakukan Deployment Website Ke Server

Deployment website ke server dilakukan setelah proses perancangan selesai. Saat melakukan *deployment*, website yang sebelumnya berjalan di lingkungan pengembangan lokal (seperti komputer pribadi) akan dipindahkan dan diaktifkan di server yang dapat diakses oleh pengguna secara *online*. Singkatnya, *deployment website* ini adalah proses mengunggah *website* yang telah selesai dikembangkan. Dalam proses pelaksanaannya, dilakukan *deployment website* ke server lokal (jaringan lokal atau jaringan internal departemen *engineering*).

3.2.20 Melakukan Pelatihan Terhadap Calon User dan Website

Dilakukan pelatihan terhadap calon user mengenai penggunaan *website* DMS yang telah dibuat. Dalam pelaksanaannya, pelatihan biasanya dilakukan saat meeting “*update progress*”, kegiatan pelatihan diantaranya mencakup demonstrasi mengenai tata cara penggunaan *website* DMS kepada calon pengguna. Selain itu, dilakukan juga pembagian *manual book* mengenai *website* DMS yang telah dibuat oleh tim DMS.



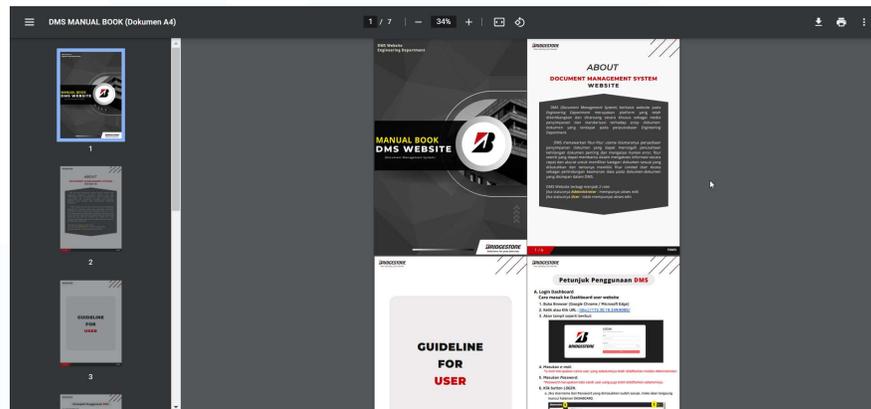
Gambar 3. 71 Demonstrasi web DMS



Gambar 3. 72 Demonstrasi web DMS



Gambar 3. 73 Demonstrasi web DMS



Gambar 3. 74 Manual Book DMS

Manual book dibuat menggunakan *tools* atau aplikasi canva dan figma. *Manual book* ini berfungsi sebagai panduan pengguna dan admin dalam menggunakan *website* DMS. Untuk menyesuaikan calon pengguna yang ada pada *departemen engineering*, tim DMS diminta untuk membagikan *manual book* dalam bentuk atau format PDF. Hal ini dilakukan untuk menghemat biaya cetak pada kertas dan lebih efisien.

3.2.21 Melakukan Penyelesaian dan Serah Terima Project

Setelah selesai melakukan tahap *testing* dan *deployment*, dilakukan *handover* atau serah terima proyek kepada departemen *engineering*. Diberikan *file source code* dan dokumen SRS dari tim proyek DMS kepada perwakilan departemen *engineering*.



Gambar 3. 75 Serah terima Proyek DMS

3.3 Kendala yang Ditemukan

Dalam pelaksanaan kerja magang, ditemukan beberapa kendala dan kesulitan yang ditemukan selama proses kerja magang:

1. Keterlambatan fasilitas laptop dari kantor karena stok yang tersedia terbatas sehingga harus menunggu sampai laptop yang baru dibeli datang.
2. Laptop dan *desktop* yang dipinjamkan memiliki *firewall* sehingga aksesnya terbatas. Sebagai contoh; situs seperti youtube yang digunakan untuk menonton *tutorial* tidak dapat diakses; Butuh akses yang sulit dan waktu yang panjang untuk

melakukan instalasi berbagai *software* yang dibutuhkan untuk keperluan *coding* karena harus melewati beberapa SOP perusahaan yang berhubungan dengan perizinan; Sulitnya melakukan pengiriman data atau dokumen terkait kebutuhan proyek karena banyak *software* pengiriman data yang diblokir. Hal ini membuat pengerjaan proyek menjadi sangat terbatas dan tidak efisien.

3. Tidak terkoneksi *wifi* karena perusahaan tidak memperbolehkan akses *wifi* kepada *device* pribadi.
4. Seringkali kehabisan kuota atau paket data seluler karena penggunaan hotspot pribadi yang terus-menerus.
5. Terjadi kesalahan pada saat menjalankan framework yang digunakan sehingga menyebabkan *style* Bootstrap tidak ter-load pada *framework* Laravel karena kurangnya kedalaman ilmu mengenai *framework* Laravel dan Bootstrap.

3.4 Solusi dari Kendala yang Ditemukan

Adapun solusi yang dilakukan untuk mengatasi kendala yang ditemukan selama proses kerja magang antara lain:

1. Melakukan *sharing* mengenai kendala yang dialami kepada mentor sehingga peserta magang diberikan *desktop* sementara untuk menggantikan laptop yang belum tersedia.
2. Menggunakan laptop pribadi, sehingga akses terhadap semua sumber yang dibutuhkan dapat dengan mudah didapatkan, meskipun kinerja lambat, karena kapasitas RAM yang belum memadai.
3. Menggunakan *Hotspot* pribadi dari perangkat seluler ke laptop pribadi.
4. Meminta *Work From Home* agar dapat menggunakan *wifi* karena tempat peserta magang tinggal tersedia koneksi *wifi*.

5. Melakukan pendalaman ilmu dan materi terkait proyek yang dikerjakan melalui *tutorial* youtube dan kursus *online* berbayar yang disediakan perusahaan. Jadi, dilakukan penyisipan *link* CDN untuk memanggil *framework* Bootstrapnya



UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA