

BAB III PELAKSANAAN KERJA PROFESI

3.1 Bidang Kerja

Kegiatan Kerja Profesi dilaksanakan selama tiga bulan di PT Senayan Trikarya Sempana, saya ditempatkan di divisi elektrik yang mencakup maintenance panel elektrik dan perbaikan pada jaringan listrik, dan juga sebagai programmer untuk *project* aplikasi data stok barang. Dalam laporan ini saya berfokus pada *project* pembuatan aplikasi yang dimana saya sebagai sistem analis dan juga *programmer*. Dan dalam proyek ini saya mengerjakan sendiri diawasi langsung oleh Assistant Manager.

Aplikasi berbasis web yang akan dibuat ini akan digunakan oleh *Supervisor* dan juga *Assistant Manager* untuk memantau stok barang yang tersedia melalui aplikasi web. Dan juga menghasilkan print out laporan stok barang.

3.2 Pelaksanaan Kerja

Kerja Profesi dimulai sejak tanggal 17 Juni 2023 – 15 September 2023 di PT Senayan Trikarya Sempana, keahlian yang harus dimiliki disini adalah kemampuan merancang web aplikasi secara jelas dan rinci kebutuhan dan *goals* aplikasi tersebut.

Untuk mencapai target, mahasiswa harus aktif dalam *update progress* setiap menyelesaikan sebuah *progress* dalam jangka waktu tertentu dan aktif dalam berkomunikasi dengan *Supervisor* atau *Assistant Manager*.

Berikut merupakan tabel rincian kegiatan yang dilakukan selama kegiatan Kerja Profesi di PT Senayan Trikarya Sempana:

Tabel 3. 1 Tabel Pelaksanaan Kerja Profesi

NO	Deskripsi Kegiatan	Juni				Juli				Agustus				September			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Analisis Kebutuhan Website			✓	✓												
2	Proses Perancangan UI					✓	✓	✓	✓	✓							
3	Proses Perancangan Website										✓	✓	✓	✓	✓		
4	Menyusun Laporan Kerja Profesi													✓	✓	✓	✓

3.2.1 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem adalah tahap awal dan krusial dalam pembuatan dasar untuk pengembangan sistem yang akan datang. Pada dasarnya, analisis kebutuhan sistem melibatkan analisis kebutuhan fungsional dan analisis kebutuhan non-fungsional. Analisis kebutuhan fungsional berfokus pada fungsi-fungsi atau fitur-fitur yang harus ada dalam sistem. Analisa kebutuhan non-fungsional berfokus pada aspek-aspek yang tidak berkaitan dengan fungsi langsung sistem, tetapi mempengaruhi kinerja, keamanan, dan pengalaman pengguna.

Berikut merupakan tabel rincian analisis kebutuhan fungsional sistem aplikasi Inventory secara keseluruhan:

Tabel 3. 2 Analisis Kebutuhan Fungsional

No	Kebutuhan Fungsional	Keterangan
1	Inventory	Pengguna dapat melihat stok <i>terupdate</i> saat ini.
2	Barang Masuk	Pengguna dapat menambahkan stok barang kedalam aplikasi sehingga stok dapat <i>terupdate</i>

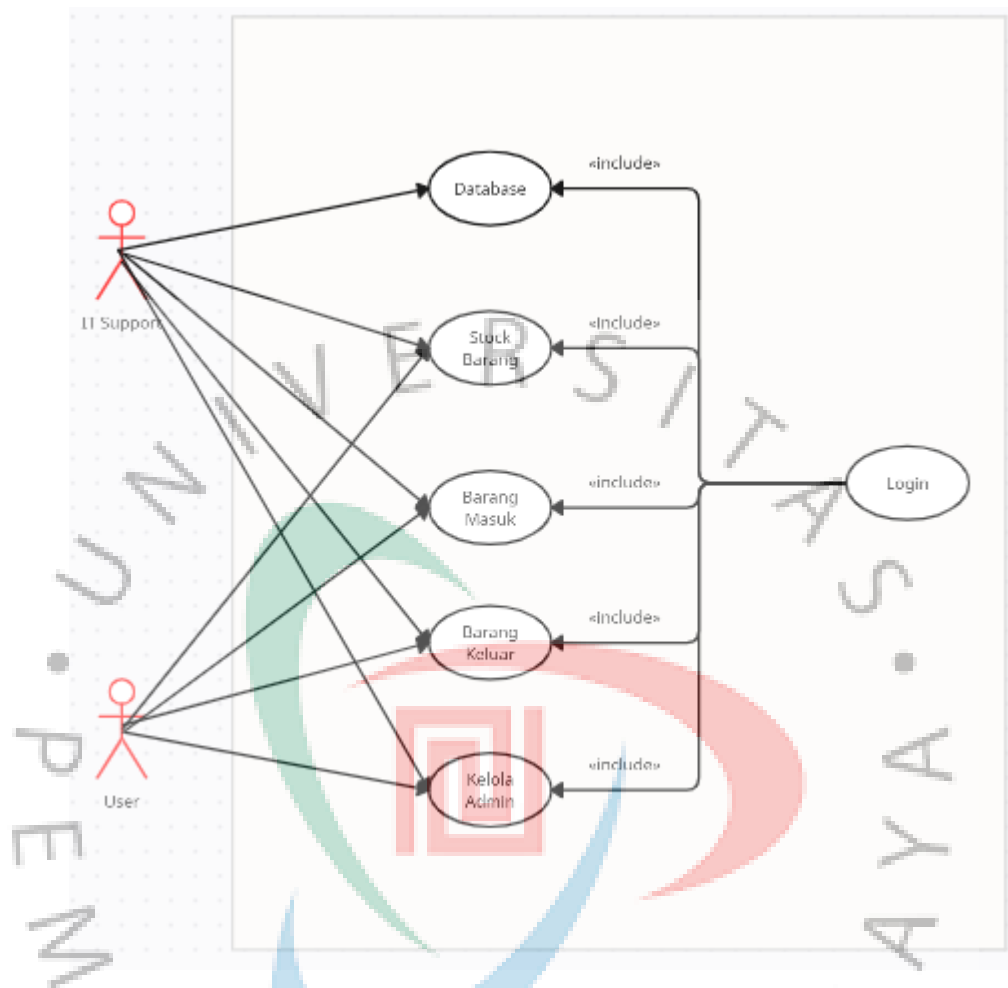
3	Barang Keluar	Pengguna dapat mengurangi stok barang sesuai dengan request pengiriman
4	Kelola User	Pengguna dapat menambahkan atau edit email dan password pengguna.
5	Logout	Pengguna dapat <i>logout</i> dari aplikasi web

Selain itu, terdapat kebutuhan non-fungsional pada sistem aplikasi Inventory:

Tabel 3. 3 Analisis Kebutuhan Non-Fungsional

No.	Kebutuhan Non-Fungsional	Keterangan
1	Responsif	Sistem diharapkan memberikan pengalaman responsif dengan waktu respons yang cepat ketika pengguna berinteraksi.
2	Keamanan	Sistem perlu memberikan mekanisme keamanan yang andal, mencakup enkripsi data pengguna dan autentikasi yang kuat.
3	Aksesibilitas Tinggi	Sistem dirancang untuk memberikan akses dan penggunaan yang mudah bagi pengguna.

3.2.2 Use Case Diagram



Gambar 3. 1 Gambar Use Case Diagram

Pada Gambar 3.1, tergambar interaksi antara dua aktor, yaitu IT Support dan *User*, dengan sistem aplikasi *inventory*. Sebagai *IT Support*, *User* memiliki kewenangan untuk melakukan login, mengelola database, mengurus stok barang, mengelola barang masuk, mengelola barang keluar, dan mengurus *User*.

Sementara itu, dalam kapasitas sebagai *User*, *user* dapat melakukan login dan memiliki kemampuan untuk mengelola stok barang, barang masuk, barang keluar, serta mengelola *User*.

3.2.3 Use Case Scenario

Use Case Scenario adalah deskripsi suatu interaksi antara seorang pengguna (aktor) dan sistem yang sedang dikembangkan. Use case scenario merinci langkah-langkah khusus yang diambil oleh pengguna dan sistem untuk mencapai tujuan tertentu. Berikut adalah detail dari tabel use case scenario dalam konteks pengembangan aplikasi inventory.

Tabel 3. 4 Use Case Scenario Login

<i>Use Case</i>	<i>Halaman Home</i>	
Deskripsi	User/Staff ingin Login	
Aktor Utama	User, Staff	
Aktor Pendukung	-	
Pra-Syarat	- User / Staff telah memiliki akun terdaftar di sistem.	
Alur Utama	<i>Step</i>	<i>Action</i>
	1	User atau Staff membuka laman login
	2	Sistem menampilkan formulir login yang mengharapkan pengguna untuk memasukkan username dan password
	3	User atau Staff memasukkan informasi yang benar pada kolom <i>username</i> dan <i>password</i>
	4	User atau Staff tekan tombol "Login"
	5	Sistem melakukan validasi terhadap data login
	6	Apabila data login terverifikasi, sistem mengalihkan pengguna ke halaman dashboard. Sebaliknya, sistem menampilkan pesan kesalahan jika data login tidak valid.
Alur Alternatif	-	
Alur Pengecualian	<i>Step</i>	<i>Action</i>

	1	Apabila <i>email</i> dan <i>password</i> yang dimasukkan tidak sesuai, sistem akan menunjukkan pesan kesalahan.
--	---	---

Tabel 3. 5 Use Case Scenario Halaman Stok Barang

<i>Use Case</i>	<i>Halaman Stok Barang</i>	
Deskripsi	User/Staff ingin mengakses halaman Stok Barang	
Aktor Utama	User, Staff	
Aktor Pendukung	-	
Pra-Syarat	- User / Staff telah memiliki akun terdaftar di sistem.	
Alur Utama	<i>Step</i>	<i>Action</i>
	1	User / Staff mengakses Stok Barang
	2	Sistem menampilkan halaman Stok Barang
Alur Alternatif	-	
Alur Pengecualian	<i>Step</i>	<i>Action</i>

Tabel 3. 6 Use Case Scenario Halaman Barang Masuk

<i>Use Case</i>	<i>Halaman Barang Masuk</i>	
Deskripsi	User/Staff ingin mengakses halaman Barang Masuk	
Aktor Utama	User, Staff	
Aktor Pendukung	-	
Pra-Syarat	- User / Staff telah memiliki akun terdaftar di sistem.	
Alur Utama	<i>Step</i>	<i>Action</i>

	1	User / Staff mengakses Barang Masuk
	2	Sistem menampilkan halaman Barang Masuk
Alur Alternatif	-	
Alur Pengecualian	Step	Action

Tabel 3. 7 Use Case Scenario Halaman Barang Keluar

Use Case	<i>Halaman Barang Keluar</i>	
Deskripsi	User/Staff ingin mengakses halaman barang keluar	
Aktor Utama	User, Staff	
Aktor Pendukung	-	
Pra-Syarat	- User / Staff telah memiliki akun terdaftar di sistem.	
Alur Utama	<i>Step</i>	<i>Action</i>
	1	User / Staff mengakses Barang Keluar
	2	Sistem menampilkan halaman Barang Keluar
Alur Alternatif	-	
Alur Pengecualian	Step	Action

Tabel 3. 8 Use Case Scenario Halaman Kelola User

Use Case	<i>Halaman Kelola User</i>	
Deskripsi	User/Staff ingin mengakses halaman kelola User	
Aktor Utama	User, Staff	
Aktor Pendukung	-	

Pra-Syarat	- User / Staff telah memiliki akun terdaftar di sistem.	
Alur Utama	<i>Step</i>	<i>Action</i>
	1	User / Staff mengakses Kelola user
	2	Sistem menampilkan halaman Kelola User
Alur Alternatif	-	
Alur Pengecualian	<i>Step</i>	<i>Action</i>

Tabel 3. 9 Use Case Scenario Tambah Stok Barang

<i>Use Case</i>	<i>Tambah Stok Barang</i>	
Deskripsi	User/Staff ingin menambahkan stok barang	
Aktor Utama	User, Staff	
Aktor Pendukung	-	
Pra-Syarat	- User / Staff telah memiliki akun terdaftar di sistem.	
Alur Utama	<i>Step</i>	<i>Action</i>
	1	User / Staff mengarahkan ke halaman Stok Barang
	2	User / Staff tekan tombol / ikon "Tambah Barang"
	3	Sistem menampilkan halaman pop up Tambah Barang
	4	User / staff mengisi form Nama Barang, Deskripsi Barang, Jumlah Stok Barang
	5	User / staff tekan tombol "Submit"
	6	Jika data barang valid, sistem menyimpan data barang ke dalam database.
	7	Sistem mengarahkan ke halaman Stok Barang dan menampilkan pesan

		informasi bahwa data barang berhasil ditambahkan
Alur Alternatif	-	
Alur Pengecualian	Step	Action
	1	Jika data barang tidak valid, sistem akan memberikan informasi kesalahan.

Tabel 3. 10 Use Case Scenario Ubah Data Stok Barang

Use Case	Ubah Data Stok Barang	
Deskripsi	User/Staff ingin mengubah stok barang	
Aktor Utama	User, Staff	
Aktor Pendukung	-	
Pra-Syarat	- User / Staff telah memiliki akun terdaftar di sistem.	
Alur Utama	Step	Action
	1	User / Staff mengarahkan ke halaman Stok Barang
	2	User / staff tekan tombol / ikon "Edit"
	3	Sistem menampilkan pop up Edit Barang
	4	User / staff mengubah data Nama Barang, Deskripsi Barang.
	5	User / staff tekan tombol "Submit"
	6	Sistem memvalidasi data barang
	7	Jika data barang valid, sistem menyimpan data barang ke dalam database.
8	Sistem mengarahkan ke halaman Stok Barang dan menampilkan pesan informasi bahwa data barang berhasil ditambahkan	

Alur Alternatif	-	
Alur Pengecualian	Step	Action
	1	Jika data barang tidak valid, sistem akan memberikan informasi kesalahan.

Tabel 3. 11 Use Case Scenario Hapus Data Stok Barang

Use Case	Hapus Data Stok Barang	
Deskripsi	User/Staff ingin menghapus stok barang	
Aktor Utama	User, Staff	
Aktor Pendukung	-	
Pra-Syarat	- User / Staff telah memiliki akun terdaftar di sistem.	
Alur Utama	Step	Action
	1	User / Staff mengarahkan ke halaman Stok Barang
	2	User / Staff tekan tombol / ikon "Delete" pada halaman Stok Barang
	3	Sistem menampilkan pesan konfirmasi "Apakah data akan dihapus? Iya / Tidak"
	4	Jika User / Staff memilih "Iya" data barang akan dihapus
	5	Sistem menghapus data barang masuk di database
Alur Alternatif	-	
Alur Pengecualian	Step	Action
	1	Jika User / Staff memilih "Tidak" data barang tidak akan dihapus

Tabel 3. 12 Use Case Scenario Tambah Data Barang Masuk

<i>Use Case</i>	<i>Tambah Data Barang Masuk</i>	
Deskripsi	User/Staff ingin menambah barang masuk	
Aktor Utama	User, Staff	
Aktor Pendukung	-	
Pra-Syarat	- User / Staff telah memiliki akun terdaftar di sistem.	
Alur Utama	<i>Step</i>	<i>Action</i>
	1	User / Staff mengarahkan ke halaman Tambah Barang
	2	User / Staff tekan tombol / ikon "Barang Masuk" pada halaman Barang Masuk
	3	Sistem menampilkan pop up Tambah Barang Masuk
	4	User / Staff mengisi formulir Nama Barang, Jumlah Barang Masuk, Nama Pengirim.
	5	User / Staff tekan tombol "Submit"
	6	Sistem memvalidasi data Barang Masuk
	7	Jika data barang valid, sistem menambah jumlah data di Barang Masuk beserta tanggal dan jam masuk
	8	Jumlah di Stok Barang dan jumlah di database akan otomatis bertambah
Alur Alternatif	-	
Alur Pengecualian	<i>Step</i>	<i>Action</i>
	1	Jika data barang tidak valid, sistem akan memberikan informasi kesalahan.

Tabel 3. 13 Use Case Scenario Ubah Data Barang Masuk

<i>Use Case</i>	<i>Ubah Data Barang Masuk</i>
-----------------	-------------------------------

Deskripsi	User/Staff ingin mengubah barang masuk	
Aktor Utama	User, Staff	
Aktor Pendukung	-	
Pra-Syarat	- User / Staff telah memiliki akun terdaftar di sistem.	
Alur Utama	<i>Step</i>	<i>Action</i>
	1	User / Staff mengarahkan ke halaman Barang Masuk
	2	User / staff tekan tombol / ikon "Edit"
	3	Sistem menampilkan pop up Edit Barang
	4	User / staff mengubah data formulir Nama Pengirim, Jumlah Barang.
	5	User / staff tekan tombol "Submit"
	6	Sistem memvalidasi data Barang Masuk
	7	Jika data barang valid, sistem menyimpan data Barang Masuk beserta ke dalam database
	8	Sistem mengarahkan ke halaman Barang Masuk dan menampilkan pesan konfirmasi bahwa data barang masuk berhasil diubah
Alur Alternatif	-	
Alur Pengecualian	<i>Step</i>	<i>Action</i>
	1	Jika data barang tidak valid, sistem akan memberikan informasi kesalahan.

Tabel 3. 14 Use Case Scenario Hapus Data Barang Masuk

<i>Use Case</i>	<i>Hapus Data Barang Masuk</i>
Deskripsi	User/Staff ingin menghapus data barang masuk

Aktor Utama	User, Staff	
Aktor Pendukung	-	
Pra-Syarat	- User / Staff telah memiliki akun terdaftar di sistem.	
Alur Utama	<i>Step</i>	<i>Action</i>
	1	User / Staff mengarahkan ke halaman Barang Masuk
	2	User / staff tekan tombol / ikon "Delete" pada halaman barang masuk
	3	Sistem menampilkan pesan konfirmasi "Apakah data akan dihapus? Iya / Tidak"
	4	Jika User / Staff memilih "Iya" data barang akan dihapus
	5	Sistem menghapus data barang masuk di database
Alur Alternatif	-	
Alur Pengecualian	<i>Step</i>	<i>Action</i>
	1	Jika User / Staff memilih "Tidak" data barang tidak akan dihapus

Tabel 3. 15 Use Case Scenario Tambah Barang Keluar

<i>Use Case</i>	<i>Tambah Barang Keluar</i>	
Deskripsi	User/Staff ingin menambah barang keluar	
Aktor Utama	User, Staff	
Aktor Pendukung	-	
Pra-Syarat	- User / Staff telah memiliki akun terdaftar di sistem.	
Alur Utama	<i>Step</i>	<i>Action</i>
	1	User / Staff mengarahkan ke halaman Barang Keluar

	2	User / staff tekan tombol / ikon "Barang Keluar" pada halaman barang keluar
	3	Sistem menampilkan pop up Tambah Barang Keluar
	4	User / staff mengisi formulir Nama Barang, Jumlah Barang, Nama Penerima
	5	User / staff tekan tombol "Submit"
	6	Sistem memvalidasi data Barang Keluar
	7	Jika data barang valid, sistem mengurangi data barang di dalam database.
	8	Sistem mengarahkan ke halaman barang keluar dan menampilkan pesan informasi bahwa data barang berhasil dikurangi
Alur Alternatif	-	
Alur Pengecualian	Step	Action
	1	Jika data barang tidak valid, sistem akan memberikan informasi kesalahan.

Tabel 3. 16 Use Case Scenario Ubah Data Barang Keluar

Use Case	Ubah Data Barang Keluar	
Deskripsi	User/Staff ingin mengubah data barang keluar	
Aktor Utama	User, Staff	
Aktor Pendukung	-	
Pra-Syarat	- User / Staff telah memiliki akun terdaftar di sistem.	
Alur Utama	Step	Action
	1	User / Staff mengarahkan ke halaman Barang Keluar
	2	User / Staff tekan tombol / ikon "Edit" pada halaman Barang Keluar

	3	Sistem menampilkan pop up Edit Barang
	4	User / staff mengubah data formulir Nama Penerima, Jumlah Barang.
	5	Sistem memvalidasi data Barang Keluar
	6	Jika data barang valid, sistem menyimpan data Barang Keluar beserta ke dalam database
	7	Sistem mengarahkan ke halaman Barang Keluar dan menampilkan pesan informasi bahwa data barang keluar berhasil diubah
Alur Alternatif	-	
Alur Pengecualian	Step	Action
	1	Jika data barang tidak valid, sistem akan memberikan informasi kesalahan.

Tabel 3. 17 Use Case Scenario Hapus Data Barang Keluar

Use Case	Hapus Data Barang Keluar	
Deskripsi	User/Staff ingin menghapus data barang keluar	
Aktor Utama	User, Staff	
Aktor Pendukung	-	
Pra-Syarat	- User / Staff telah memiliki akun terdaftar di sistem.	
Alur Utama	Step	Action
	1	User / Staff mengarahkan ke halaman Barang Keluar
	2	User / Staff tekan tombol / ikon "Delete" pada halaman Barang Keluar
	3	Sistem menampilkan pesan konfirmasi "Apakah data akan dihapus? Iya / Tidak"

Alur Alternatif	4	Jika User / Staff memilih “Iya” data barang akan dihapus
	5	Sistem menghapus data barang keluar di database
	-	
Alur Pengecualian	Step	Action
	1	Jika User / Staff memilih “Tidak” data barang tidak akan dihapus

Tabel 3. 18 Use Case Scenario Tambah Kelola User

Use Case	Tambah Kelola User	
Deskripsi	User/Staff ingin menambah User	
Aktor Utama	User, Staff	
Aktor Pendukung	-	
Pra-Syarat	- User / Staff telah memiliki akun terdaftar di sistem.	
Alur Utama	Step	Action
	1	User / Staff mengarahkan ke halaman Kelola User
	2	User / Staff tekan tombol / ikon “Tambah User” pada halaman Kelola User
	3	Sistem menampilkan pop up Tambah User
	4	User / staff mengisi formulir Email, dan Password
	5	User / staff tekan tombol “Submit”
	7	Sistem memvalidasi data Kelola User
	8	Jika data user valid, sistem menyimpan data Kelola User beserta ke dalam database

	9	Sistem mengarahkan ke halaman Kelola User dan menampilkan pesan informasi bahwa data User berhasil ditambah
Alur Alternatif	-	
Alur Pengecualian	Step	Action
	1	Jika data user tidak valid, sistem akan memberikan informasi kesalahan.

Tabel 3. 19 Use Case Scenario Ubah Data Kelola User

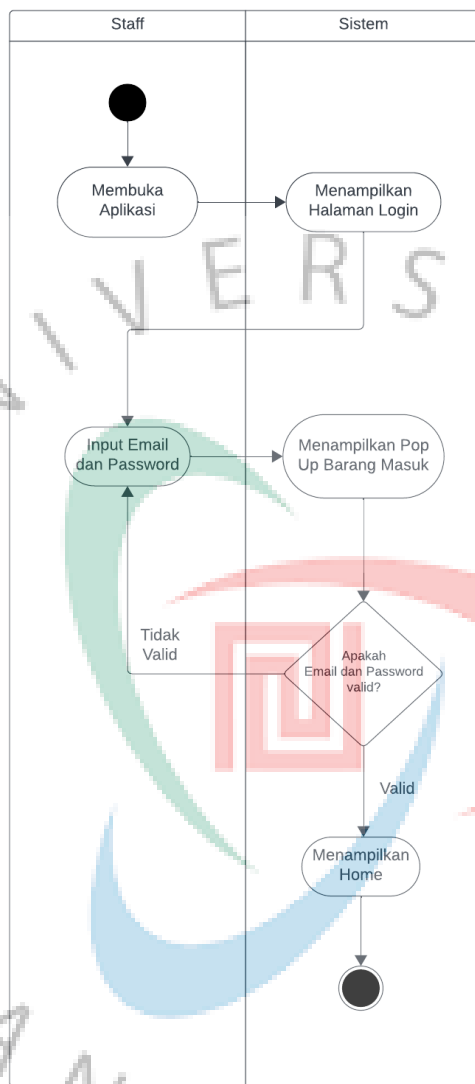
Use Case	Ubah Data Kelola User	
Deskripsi	User/Staff ingin mengubah data kelola User	
Aktor Utama	User, Staff	
Aktor Pendukung	-	
Pra-Syarat	- User / Staff telah memiliki akun terdaftar di sistem.	
Alur Utama	Step	Action
	1	User / Staff mengarahkan ke halaman Kelola User
	2	User tekan ikon "Edit" pada halaman Kelola User
	3	Sistem menampilkan pop up Edit
	4	User / staff mengubah formulir Email, dan Password
	5	User / staff tekan tombol "Submit"
	7	Sistem memvalidasi data Kelola User
	8	Jika data user valid, sistem menyimpan data Kelola User beserta ke dalam database

	9	Sistem mengarahkan ke halaman Kelola User dan menampilkan pesan informasi bahwa data User berhasil diubah
Alur Alternatif	-	
Alur Pengecualian	Step	Action
	1	Jika data user tidak valid, sistem akan memberikan informasi kesalahan.

Tabel 3. 20 Use Case Scenario Hapus Data Kelola User

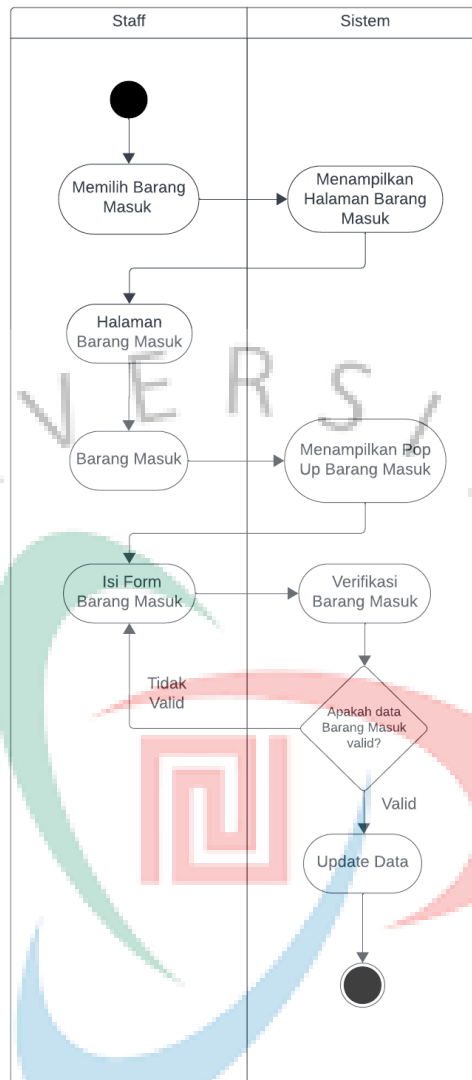
Use Case	<i>Hapus Data Kelola User</i>	
Deskripsi	User/Staff ingin menghapus data kelola User	
Aktor Utama	User, Staff	
Aktor Pendukung	-	
Pra-Syarat	- User / Staff telah memiliki akun terdaftar di sistem.	
Alur Utama	<i>Step</i>	<i>Action</i>
	1	User / Staff mengarahkan ke halaman Kelola User
	2	User / Staff tekan tombol / ikon "Delete" pada halaman Kelola User
	3	Sistem menampilkan pesan konfirmasi "Apakah data akan dihapus? Iya / Tidak"
	4	Jika User / Staff memilih "Iya" data user akan dihapus
	5	Sistem menghapus data User di database
Alur Alternatif	-	
Alur Pengecualian	Step	Action
	1	Jika User / Staff memilih "Tidak" data User tidak akan dihapus

3.2.4 Activity Diagram



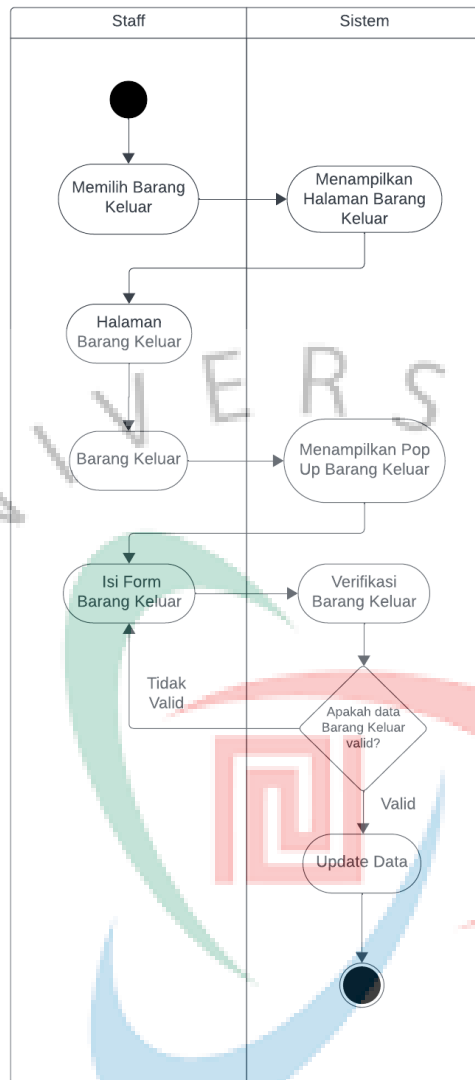
Gambar 3. 2 Login Activity Diagram

Gambar 3.2 adalah menggambarkan tahapan proses autentikasi pada platform ini, di mana pengguna terlibat dalam menginputkan informasi kredensial berupa alamat email dan kata sandi. Langkah ini diperlukan untuk memperoleh akses ke dalam aplikasi.



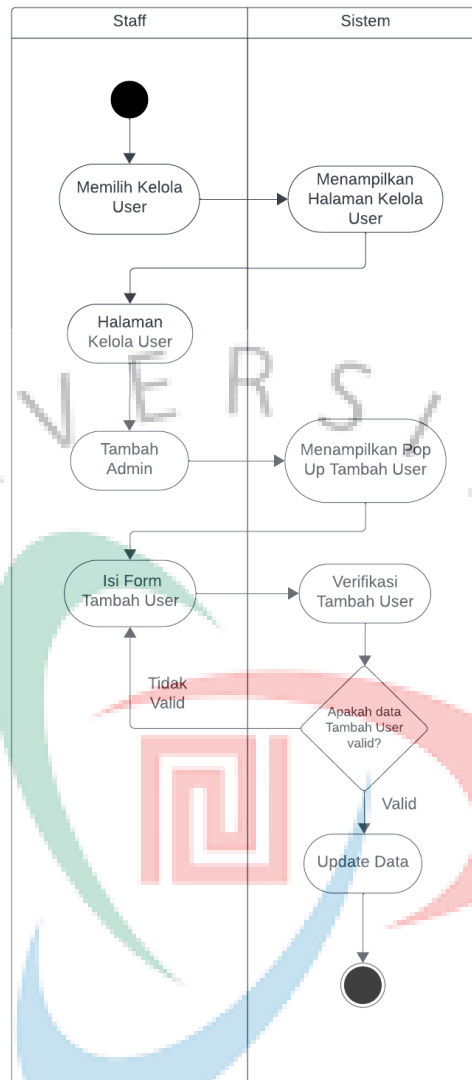
Gambar 3. 3 Barang Masuk Activity Diagram

Gambar 3.3 adalah menggambarkan tahapan proses penerimaan barang pada halaman ini. Pengguna diberikan kemampuan untuk menambahkan item sesuai dengan pengiriman yang diterima. Fasilitas ini memungkinkan pengguna untuk mengidentifikasi jumlah barang yang diterima, memberikan nama pada setiap item, serta mencatat tanggal kedatangan barang tersebut.



Gambar 3. 4 Barang Keluar Activity Diagram

Gambar 3.4 menggambarkan tahapan proses pengeluaran barang pada halaman ini. Pengguna diberikan kemampuan untuk mengurangi jumlah barang yang akan dikeluarkan sejalan dengan jumlah yang diambil oleh staf.



Gambar 3. 5 Kelola User Activity Diagram



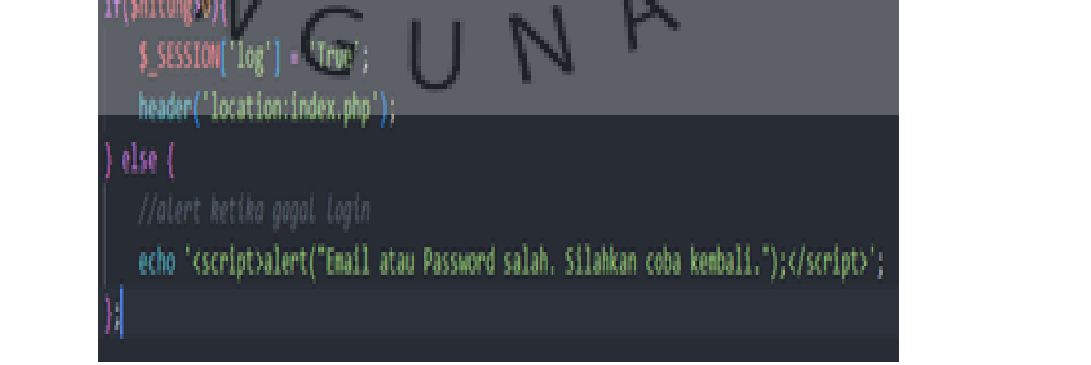
Gambar 3.5 memvisualisasikan iterasi proses Kelola User pada antarmuka pengguna ini. Pengguna disediakan dengan kemampuan untuk melakukan modifikasi terhadap informasi identifikasi akun, yakni melalui fungsi penggantian alamat surel (email) atau penyesuaian kata sandi. Selain itu, sistem mengizinkan pengguna untuk mengeksekusi penambahan administrator baru ke dalam entitas sistem ini. Fitur-fitur ini terintegrasi harmonis dalam kerangka kerja tatakelola yang disajikan, memungkinkan pengguna untuk melakukan manajemen administratif dengan efisiensi dan keandalan yang optimal.


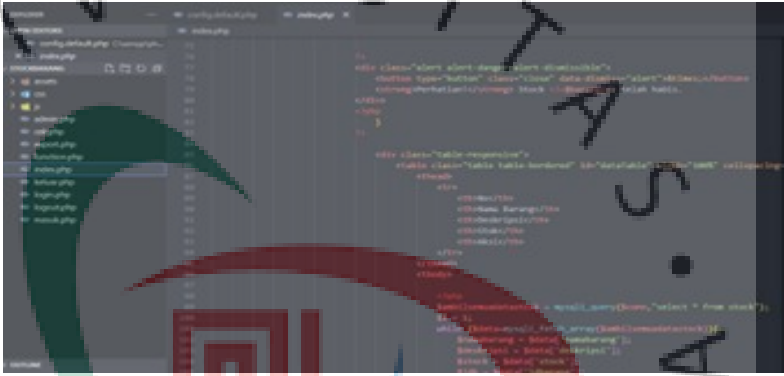

3.2.5 White Box


No.	Script	Hasil yang ditampilkan
1	Login <pre> require "function.php"; if(isset(\$_POST['login'])){ \$email = \$_POST['email']; \$password = \$_POST['password']; //koneksi ke database \$koneksi = mysqli_connect(\$host, \$username, \$password, \$database); \$sql = "SELECT * FROM login where email='\$email' and password='\$password'"; \$result = mysqli_query(\$koneksi, \$sql); \$row = mysqli_fetch_row(\$result); </pre>	Dapat masuk ke Web Stok Barang 
2	login Failed <pre> if(\$hitung>0){ \$_SESSION['login'] = true; header("location:index.php"); } else { alert("Keriting gagal login"); echo "<script>alert('Email atau Password salah. Silahkan coba kembali.'); </pre>	Pesan informasi email dan password salah <p>localhost says Email atau Password salah. Silahkan coba kembali.</p> 
3	index.php 	Tampilan laman Stok Barang 
4	Tambah stok barang <pre> <div class="modal" id="myModal"> <div class="modal-dialog"> <div class="modal-content"> <div class="modal-header"> <h4>Tambah Barang </div> <div class="modal-body"> <input type="text" name="nama_barang" placeholder="Nama Barang" class="form-control"> <input type="text" name="deskripsi" placeholder="Deskripsi" class="form-control"> <input type="number" name="stock" placeholder="Stock" class="form-control"> </div> <div class="modal-footer"> <input type="button" value="Submit" class="btn btn-primary"> </div> </div> </div> </div> </pre>	Form Tambah Barang 
5	Edit stok barang	Form edit Stok Barang


	<pre> <div class="modal fade" id="delete-<?= \$id; ?>"> <div class="modal-dialog"> <div class="modal-content"> <div class="modal-header"> <div class="modal-title">Hapus Barang?</div> <button type="button" class="close" data-dismiss="modal"></button> </div> <div class="modal-body"> <form method="post"> Apakah anda yakin ingin menghapus Broco 3 Lubang? <input type="hidden" name="id" value="<?= \$id; ?>"> <input type="hidden" name="id" value="<?= \$id; ?>"> </form> <button type="submit" class="btn btn-danger" name="hapusbarang">Hapus</button> </div> </div> </div> </div> </pre>	
15	<p>user.php</p> <pre> <div class="modal fade" id="delete-<?= \$id; ?>"> <div class="modal-dialog"> <div class="modal-content"> <div class="modal-header"> <div class="modal-title">Hapus User?</div> <button type="button" class="close" data-dismiss="modal"></button> </div> <div class="modal-body"> <form method="post"> Apakah anda yakin ingin menghapus Broco 3 Lubang? <input type="hidden" name="id" value="<?= \$id; ?>"> <input type="hidden" name="id" value="<?= \$id; ?>"> </form> <button type="submit" class="btn btn-danger" name="hapusbarang">Hapus</button> </div> </div> </div> </div> </pre>	<p>Laman kelola user</p>
16	<p>Tambah user</p> <pre> <div class="modal fade" id="formuser"> <div class="modal-dialog"> <div class="modal-content"> <div class="modal-header"> <div class="modal-title">Tambah user?</div> <button type="button" class="close" data-dismiss="modal"></button> </div> <div class="modal-body"> <form method="post"> <input type="text" name="email" class="form-control" required=""> <input type="password" name="password" class="form-control" required=""> <input type="password" name="password" class="form-control" value=""> <input type="submit" class="btn btn-primary" name="adduser">Submit</button> </form> </div> </div> </div> </div> </pre>	<p>Form tambah user</p>
17	<p>Edit user</p> <pre> <div class="modal fade" id="edit-<?= \$id; ?>"> <div class="modal-dialog"> <div class="modal-content"> <div class="modal-header"> <div class="modal-title">Edit user?</div> <button type="button" class="close" data-dismiss="modal"></button> </div> <div class="modal-body"> <form method="post"> <input type="text" name="email" value=""> <input type="password" name="password" class="form-control" value=""> <input type="password" name="password" class="form-control" value=""> <input type="submit" class="btn btn-primary" name="updateuser">Submit</button> </form> </div> </div> </div> </div> </pre>	<p>Form edit user</p>
18	<p>Hapus user</p> <pre> <div class="modal fade" id="delete-<?= \$id; ?>"> <div class="modal-dialog"> <div class="modal-content"> <div class="modal-header"> <div class="modal-title">Hapus User?</div> <button type="button" class="close" data-dismiss="modal"></button> </div> <div class="modal-body"> <form method="post"> Apakah anda yakin ingin menghapus elektriks@gmail.com? <input type="hidden" name="id" value="<?= \$id; ?>"> </form> <button type="submit" class="btn btn-danger" name="hapususer">Hapus</button> </div> </div> </div> </div> </pre>	<p>Tampilan konfirmasi hapus user</p>

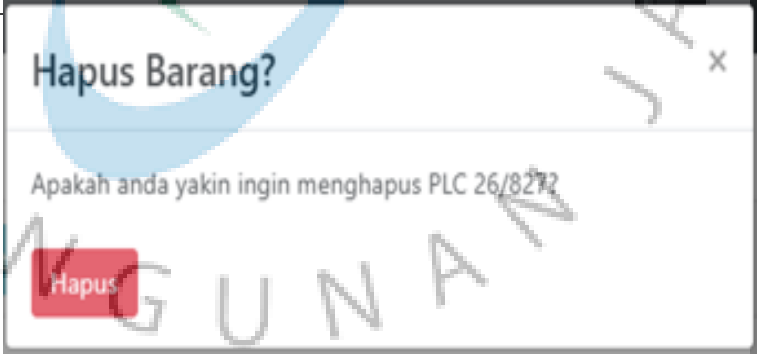
3.2.6 Black Box

1	Skenario Pengujian User melakukan login	Hasil Jika data valid, akan masuk ke aplikasi berbasis web
 <pre> require 'function.php'; //cek login, register atau tidak if(isset(\$_POST['login'])){ \$email = \$_POST['email']; \$password = \$_POST['password']; //cek user dan password database \$koneksi = mysql_query(\$con, "SELECT * FROM login where email = '\$email' and password = '\$password'"); \$hitung = mysql_num_rows(\$koneksi); </pre>		
2	Login Gagal	Jika data tidak valid, akan menampilkan pesan informasi "Email atau Password salah. Silahkan coba kembali."
		
3	User mengakses barang masuk	Menampilkan laman barang masuk
 <pre> if(\$hitung>0){ \$_SESSION['log'] = 'True'; header("location:index.php"); } else { //alert ketika gagal login echo '<script>alert("Email atau Password salah. Silahkan coba kembali.");</script>'; }; </pre>		
4	User tekan tombol barang masuk	Menampilkan form barang masuk

	<p>localhost says</p> <p>Email atau Password salah. Silahkan coba kembali.</p> <p style="text-align: right;"></p>	
5	User tekan tombol edit	Menampilkan form edit barang masuk
		
6	User tekan tombol hapus	Menampilkan konfirmasi hapus data
		
7	User tekan tab barang keluar	Menampilkan laman barang keluar

	<pre> <!-- Modal --> <div class="modal fade" id="myModal"> <div class="modal-dialog"> <div class="modal-content"> <!-- Modal header --> <div class="modal-header"> <h4 class="modal-title">Tambah Barang</h4> <button type="button" class="close" data-dismiss="modal">&times;</button> </div> <!-- Modal body --> <form method="post"> <div class="modal-body"> <input type="text" name="namabarang" placeholder="Nama Barang" class="form-control" required>
 <input type="text" name="deskripsi" placeholder="Deskripsi" class="form-control" required>
 <input type="number" name="stock" placeholder="Stock" class="form-control" required>
 <button type="submit" class="btn btn-primary" name="addnbarang">Submit</button> </div> </form> </div> </div> </div> </pre>	
8	User tekan tombol barang keluar	Menampilkan form barang keluar
		
9	User tekan tombol edit	Menampilkan form edit barang keluar
	<pre> <!-- Modal header --> <div class="modal-header"> <h4 class="modal-title">Edit Barang</h4> <button type="button" class="close" data-dismiss="modal">&times;</button> </div> <!-- Modal body --> <form method="post"> <div class="modal-body"> <input type="text" name="namabarang" value="{<- \$namabarang;}" class="form-control" required>
 <input type="text" name="deskripsi" value="{<- \$deskripsi;}" class="form-control" required>
 <input type="hidden" name="id" value="{<- \$id;}"> <button type="submit" class="btn btn-primary" name="updatebarang">Submit</button> </div> </form> </pre>	
10	User tekan tombol hapus	Menampilkan konfirmasi hapus data

		
11	User mengakses tab kelola user	Menampilkan laman kelola user
		<pre> <div class="modal-dialog"> <div class="modal-content"> <!-- Modal header --> <div class="modal-header"> <h4 class="modal-title">Hapus Barang?</h4> <button type="button" class="close" data-dismiss="modal">&times;</button> </div> <!-- Modal body --> <form method="post"> <div class="modal-body"> Apakah anda yakin ingin menghapus <?-\$namabarang;?> <input type="hidden" name="ldb" value="<?-\$ldb;?>" />

 <button type="submit" class="btn btn-danger" name="hapusbarang">Hapus</button> </div> </form> </pre>
12	User tekan tombol tambah user	Menampilkan form tambah user
		
13	User tekan tombol edit	Menampilkan form edit user

```

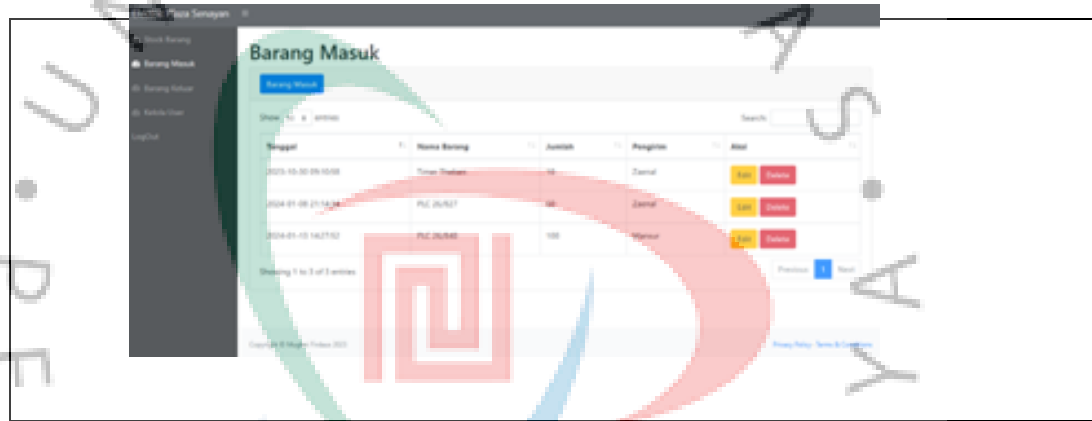
<div class="card mb-4">
  <div class="card-header">
    <!-- Button to open the Modal -->
    <button type="button" class="btn btn-primary" data-toggle="modal" data-target="#myModal">
      Barang Masuk
    </button>
  </div>
  <div class="card-body">
    <div class="table-responsive">
      <table class="table table-bordered" id="datatable" width="100%" cellpadding="0">
        <thead>
          <tr>
            <th></th>
            <th>Tanggal</th>
            <th>Nama Barang</th>
            <th>Jumlah</th>
            <th>Kategori</th>
            <th>Aksi</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td></td>
            <td></td>
            <td></td>
            <td></td>
            <td></td>
            <td></td>
          </tr>
        </tbody>
      </table>
    </div>
  </div>
  <script>
    $exampledatatock = mysql_query($conn,"select * from masuk m, stock s where s.idbarang
    while ($data=mysql_fetch_array($exampledatatock)){
      $idm = $data['idbarang'];
      $idms = $data['idmasuk'];
      $tanggal = $data['tanggal'];
      $jumlahbarang = $data['jumlahbarang'];
      $qty = $data['qty'];
    }
  </script>

```

14

User tekan tombol hapus

Menampilkan konfirmasi hapus data user



3.2.7 Hasil

The image shows a login interface with a light gray header containing the word 'Login'. Below the header, there are two input fields: 'Email' with the text 'elektrikps@gmail.com' and 'Password' with masked characters. A blue 'Login' button is located at the bottom left of the form area.

Gambar 3. 6 halaman login

Pada Gambar 3.6, terlihat sistem yang menampilkan tampilan masuk untuk aplikasi Manajemen Stok Barang. Bagian masuk ini memegang peranan vital dalam struktur sistem karena memungkinkan pengguna untuk membuka dan menggunakan berbagai fitur aplikasi dengan keamanan yang terjamin. Pada layar masuk, pengguna diminta untuk memasukkan informasi login yang akurat, seperti alamat email dan kata sandi, guna mengidentifikasi dan memverifikasi identitas mereka. Setelah informasi login diverifikasi, pengguna diberikan hak akses ke berbagai fitur aplikasi yang tersedia, termasuk Stok Barang, Barang Masuk, Barang Keluar, dan Pengelolaan User.

No	Nama Barang	Deskripsi	Stok	Aksi
1	MCB 10A 1 Phase	Breaker	50	Edit Delete
2	PLC 26/840	Lampu	20	Edit Delete
3	PLC 26/827	Lampu	50	Edit Delete
4	PLC 18/840	Lampu	50	Edit Delete
5	PLC 18/827	Lampu	100	Edit Delete
6	Broco 3 Lubang	Stop Kontak	85	Edit Delete

Gambar 3. 7 Halaman Home

Pada Gambar 3.7, terdapat sistem yang memperlihatkan laman Beranda pada aplikasi Inventory. Laman Beranda ini berfungsi sebagai antarmuka awal dalam sistem tersebut sekaligus memperlihatkan laman Stok Barang, memberikan kesempatan bagi pengguna untuk membuka dan memanfaatkan berbagai fitur aplikasi seperti Stok Barang, Barang Masuk, Barang Keluar, dan Pengelolaan User.

3.2.8 Tools dan Teknologi

Dalam konteks pekerjaan di bidang Teknologi Informasi (TI) pada proyek pengembangan aplikasi inventory, terdapat sejumlah perangkat yang dapat digunakan untuk mendukung pelaksanaan tugas dan tanggung jawab TI. Beberapa peralatan yang bermanfaat bagi TI dalam melakukan pengujian perangkat lunak meliputi:

- 1) Visual Studio Code



Gambar 3. 8 Visual Studio Code

Visual Studio Code, sebuah pengedit kode sumber lintas platform, mendapat popularitas yang besar sebagai produk yang dikembangkan oleh Microsoft. Dikenal di kalangan pengembang perangkat lunak karena keunggulan, kemampuan fleksibel, dan ketersediaannya di berbagai platform, Visual Studio Code menjadi pilihan utama. Memberikan lingkungan pengembangan yang efisien dan bertenaga, alat ini memberikan kemudahan dalam pembuatan dan penyuntingan kode sumber.

2) Bootstrap



Gambar 3. 9 Bootstrap

Bootstrap adalah sebuah kerangka kerja (framework) front-end yang mendapat popularitas luas di seluruh dunia, baik digunakan dalam pengembangan aplikasi berukuran kecil maupun besar. Keandalan, performa yang optimal, dan fleksibilitasnya dalam membangun antarmuka pengguna membuatnya menjadi pilihan utama bagi pengembang dan organisasi. Bootstrap menyediakan beragam komponen dan alat yang mempermudah desain responsif, memastikan tampilan yang konsisten di berbagai perangkat. Dengan kemampuannya yang canggih, Bootstrap menjadi opsi yang diminati dalam membangun dan mengelola tata letak serta gaya visual, memungkinkan akses yang efisien terhadap pengembangan dan tampilan antarmuka aplikasi.

3) XAMPP



Gambar 3. 10 Xampp

XAMPP adalah salah satu paket pengembangan web yang sangat populer dan sering digunakan di berbagai belahan dunia, baik untuk proyek kecil maupun proyek skala besar. Kelebihanannya terletak pada ketersediaan lengkapnya yang melibatkan Apache sebagai server web, MySQL sebagai Sistem Manajemen Basis Data Relasional (RDBMS), PHP sebagai bahasa pemrograman, dan Perl sebagai bahasa skrip. Kombinasi ini memberikan keandalan yang tinggi, kinerja optimal, serta fleksibilitas dalam pengelolaan dan pengembangan aplikasi web. XAMPP menjadi pilihan utama bagi pengembang dan organisasi yang ingin menyimpan, mengelola, dan mengakses data mereka dengan mudah dan efisien dalam lingkungan pengembangan web lokal.

4) MySQL



Gambar 3. 11 MySQL

MySQL merupakan salah satu Sistem Manajemen Basis Data Relasional (RDBMS) yang sangat populer di seluruh dunia, baik digunakan dalam aplikasi berukuran kecil maupun besar. Keunggulannya dalam keandalannya, kinerja yang optimal, serta fleksibilitasnya dalam pengelolaan data membuatnya menjadi opsi yang diminati oleh pengembang dan organisasi untuk menyimpan, mengelola, dan mengakses data mereka.

3.3 Kendala Yang Dihadapi

Dalam proyek pengembangan aplikasi stok barang elektrik berbasis web PT. Senayan Trikarya Sempana, Praktikan menghadapi beberapa kendala. Beberapa kendala dihadapi adalah sebagai berikut:

1. Praktikan menghadapi kendala dalam pemahaman mendalam terkait penggunaan Bootstrap 4 dan database MySQL dalam konteks pengembangan aplikasi berbasis web. Kendala ini muncul karena Praktikan memiliki keterbatasan pemahaman terhadap konsep-konsep kritis yang terkandung dalam Bootstrap 4 dan MySQL.
2. Menu data barang menimbulkan kendala karena adanya kebutuhan untuk mengelola stok barang secara otomatis agar selalu terupdate.
3. Kendala lainnya muncul dari kesibukan pembimbing, yang dapat menghambat praktikan dalam mengkoordinasikan kemajuan proyek yang sedang dikerjakan.

3.4 Cara Mengatasi Kendala

Praktikan memiliki beberapa solusi untuk mengatasi kendala yang dihadapi Praktikan selama melaksanakan kegiatan Kerja Profesi. Berikut adalah cara mengatasi kendala yang dialami oleh Praktikan selama pembuatan aplikasi *inventory* berbasis web :

1. Praktikan harus menggali pengetahuan melalui sumber-sumber seperti *Google* atau membaca jurnal tentang aplikasi berbasis web untuk dapat mengadaptasi konsep-konsep tersebut dalam pengembangan aplikasi inventaris.
2. Menanggulangi hambatan otomatisasi pembaruan stok barang dengan melakukan pencarian informasi secara *online*.
3. Atur pertemuan rutin dengan pembimbing kerja untuk membahas kemajuan proyek dan mengatasi hambatan.

3.5 Pembelajaran Yang Diperoleh dari Kerja Profesi

Praktikan memperoleh beberapa pembelajaran yang bermanfaat untuk dijadikan bekal di dunia kerja nantinya. Berikut beberapa pembelajaran yang diperoleh selama melaksanakan Kerja Profesi :

1. Praktikan menunjukkan keterampilan adaptasi melalui usaha memahami cara berkomunikasi yang efektif dengan mentor dan rekan kerja.
2. Tanggung jawab terhadap tugas yang diberikan oleh perusahaan menjadi landasan prinsip bagi Praktikan.
3. Memperoleh pengetahuan dalam pengembangan aplikasi berbasis web dan implementasi sistem CRUD (*Creat Read Update and Delete*).

