

LAPORAN AKHIR
PEMROGRAMAN BERBASIS WEBSITE



D-PANAN :
SISTEM PENYEWAAN LAPANGAN OLAHRAGA BERBASIS WEBSITE

Farda Rezkiya Zulfa (222410103028)

Moh Amirul Hamzah (222410103044)

M. Ainol Yakin (222410103090)

PRODI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS JEMBER

2024

BAB I

PENDAHULUAN

1.1 Latar Belakang

Olahraga pada dasarnya merupakan salah satu aspek kehidupan yang penting untuk setiap manusia untuk menjaga kondisi fisik dan kesehatan dengan baik. Masa sekarang, kesadaran masyarakat terhadap kesehatan semakin tinggi sehingga masyarakat mulai tertarik untuk mendatangi tempat-tempat olahraga baik seorang diri atau bersama dengan teman. Hal ini disadari oleh para pengelola lapangan olahraga sehingga pengelola lapangan perlu meningkatkan fasilitas, pelayanan dan menyediakan informasi yang dibutuhkan oleh para penyewa.

Proses sewa juga menjadi salah satu fasilitas pelayanan yang perlu diperhatikan dan perlu mempertimbangkan efisiensi dari prosedur sewa. Tidak jarang proses reservasi dilakukan melalui telepon maupun kunjungan langsung. Sistem manual ini bisa dibilang menjadi sebuah permasalahan tatkala penyewa terkendala waktu, jarak dan tempat.

Kami menyadari pentingnya memfasilitasi proses penyewaan lebih mudah dan efisien. Oleh karena itu, kami memutuskan untuk merancang sistem penyewaan lapangan olahraga bagi pengelola dan penyewa lapangan.

Dengan menggunakan sistem ini, pengelola dapat memfasilitasi penyediaan informasi mengenai fasilitas lapangan, jadwal ketersediaan lapangan terupdate, pengelolaan penyewa serta analisis dan laporan kinerja. Sistem ini juga memfasilitasi penyewa untuk mencari lokasi dan jenis olahraga yang diinginkan, melihat informasi detail lapangan, dan kemudahan melakukan reservasi.

Sistem ini memiliki fitur untuk melakukan reservasi dengan mudah untuk penyewa sehingga penyewa tidak perlu kunjungan langsung atau melalui telepon karena penyediaan informasi jadwal lapangan terupdate secara *real-time* dan dapat diakses dimanapun. Begitu juga untuk pengelola, fitur tersebut dapat memudahkan pencatatan dan pengelolaan penyewa secara efisien. Dengan sistem ini, kami bermaksud memberikan dampak positif dalam peningkatan efisiensi, kenyamanan, dan kemudahan baik dari pengelola lapangan maupun penyewa.

1.2 Rumusan Masalah

- 1.) Bagaimana merancang sistem penyewaan lapangan olahraga yang dapat menyediakan informasi terupdate mengenai ketersediaan lapangan dan fasilitas yang ada, serta memudahkan proses reservasi bagi penyewa secara online?
- 2.) Bagaimana meningkatkan efisiensi pengelolaan penyewa, pencatatan, dan analisis kinerja lapangan olahraga bagi pengelola, sehingga dapat memberikan pelayanan yang lebih baik dan responsif?

1.3 Tujuan

- 1.) Merancang sistem penyewaan lapangan olahraga yang dapat menyediakan informasi terupdate mengenai ketersediaan lapangan dan fasilitas yang ada, serta memudahkan proses reservasi bagi penyewa secara online.
- 2.) Meningkatkan efisiensi pengelolaan penyewa, pencatatan, dan analisis kinerja lapangan olahraga bagi pengelola, sehingga dapat memberikan pelayanan yang lebih baik dan responsif.

1.4 Manfaat

Manfaat secara spesifik dari penelitian ini adalah menghasilkan alat bantu dalam proses reservasi atau penyewaan lapangan olahraga bagi pengelola dan penyewa dengan sistem informasi yang terupdate dan *real-time*.

BAB II

RANCANGAN APLIKASI

2.1 Kebutuhan Sistem

- **Fitur Registrasi**

Fitur ini memungkinkan user untuk membuat akun baru pada sistem sebelum bisa mengakses fitur dalam aplikasi.

- **Fitur Login**

Fitur ini memungkinkan untuk user yang sudah terdaftar sistem sehingga user dapat mengakses fitur - fitur yang ada dalam sistem.

- **Fitur Logout**

Fitur ini memungkinkan user untuk mengeluarkan akun nya dari sistem.

- **Role Admin**

Role ini memiliki akses penuh dalam fitur manajemen dan administrasi. **Admin** bertanggung jawab untuk mengelola dan memelihara data, pengguna, dan operasional sistem.

Berikut beberapa fitur dari **Admin** :

- **Fitur Melihat Data Detail Gedung**

Fitur ini memungkinkan **Admin** untuk mengakses dan meninjau informasi lengkap mengenai setiap gedung yang terdaftar.

- **Fitur Mengatur Status Gedung**

Fitur ini memungkinkan **Admin** untuk mengubah dan mengelola status operasional gedung yang terdaftar. Status ini mencerminkan kondisi dan ketersediaan gedung untuk disewa atau digunakan.

- **Role Renter**

Role ini ditujukan untuk user pengguna yang ingin melakukan reservasi penyewaan lapangan olahraga. **Renter** memiliki akses ke fitur-fitur yang berkaitan dengan penyewaan, transaksi, dan pengelolaan jadwal.

Berikut beberapa fitur dari **Renter** :

- **Fitur Melihat Jadwal Lapangan**

Fitur ini memungkinkan **Renter** untuk melihat ketersediaan dan jadwal penggunaan lapangan atau fasilitas yang bisa disewa.

- **Fitur Menambah Data Transaksi**

Fitur ini memungkinkan **Renter** untuk mencatat dan menyimpan detail transaksi penyewaan yang telah dilakukan.

- **Fitur Melihat Data Transaksi**

Fitur ini memungkinkan **Renter** untuk mengakses dan meninjau riwayat transaksi penyewaan mereka.

- **Fitur Mencari Data Transaksi**

Fitur ini memungkinkan **Renter** untuk menemukan transaksi tertentu berdasarkan kriteria pencarian yang spesifik.

- **Role Owner**

Role ini ditujukan untuk pemilik gedung atau fasilitas yang disewakan. Pemilik (owner) memiliki akses ke fitur-fitur yang berkaitan dengan pengelolaan gedung, transaksi, penyewa, dan laporan keuangan.

Berikut beberapa fitur dari **Owner** :

- **Menambah Data Transaksi**

Fitur ini memungkinkan **Owner** untuk mencatat dan menyimpan detail transaksi penyewaan yang telah dilakukan oleh **Renter** yang melakukan kunjungan langsung/*offline*.

- **Mengedit Data Transaksi**

Fitur ini memungkinkan **Owner** untuk memodifikasi atau memperbarui informasi yang terkait dengan seluruh transaksi penyewaan lapangan.

- **Menghapus Data Transaksi**

Fitur ini memungkinkan **Owner** untuk menghapus data transaksi penyewaan lapangan yang salah dicatat, tidak relevan, atau tidak sah.

- **Melihat Data Transaksi**

Fitur ini memungkinkan **Owner** untuk mengakses dan meninjau seluruh riwayat transaksi penyewaan lapangan.

- **Mencari Data Transaksi**

Fitur ini memungkinkan **Owner** untuk menemukan transaksi penyewaan lapangan tertentu berdasarkan kriteria pencarian yang spesifik.

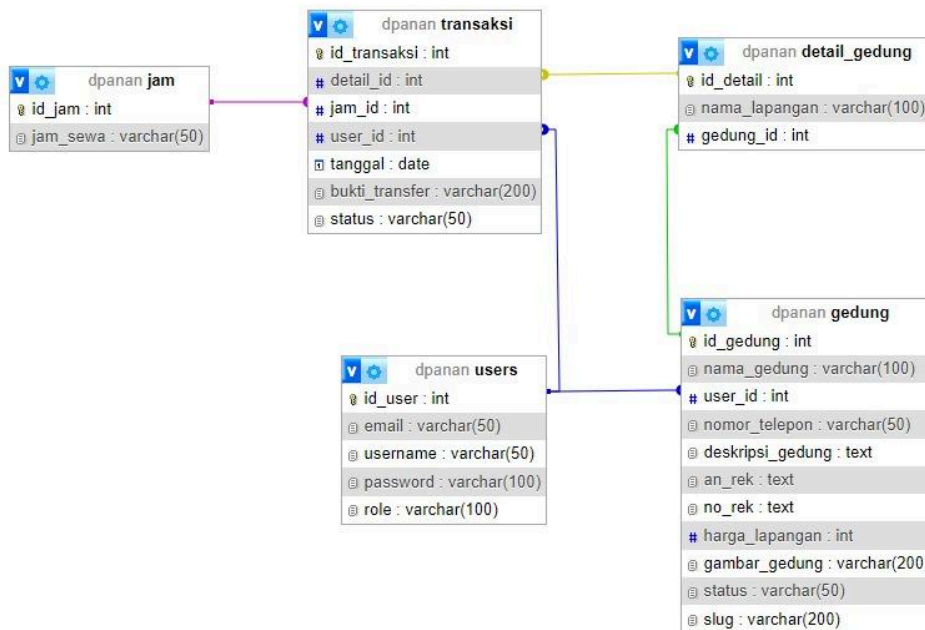
- **Menerima pesanan**

Fitur ini memungkinkan **Owner** untuk mengkonfirmasi transaksi yang telah dilakukan oleh **Renter** apabila pembayaran telah sesuai.

- **Menolak pesanan**

Fitur ini memungkinkan **Owner** untuk mengkonfirmasi transaksi yang telah dilakukan oleh **Renter** apabila pembayaran tidak sesuai.

2.2 ERD



Data Transaksi		
NO	ATTRIBUTE	TIPE DATA
1.	id_transaksi (pk)	int
2.	detail_id (fk)	int
3.	jam_id (fk)	int
4.	user_id (fk)	int
5.	tanggal	date
6.	bukti_transfer	varchar(200)
7.	status	varchar(50)

Data User		
NO	ATTRIBUTE	TIPE DATA
1.	id_user (pk)	int
2.	email	varchar(50)
3.	username	varchar(50)
4.	password	varchar(100)
5.	role	varchar(100)

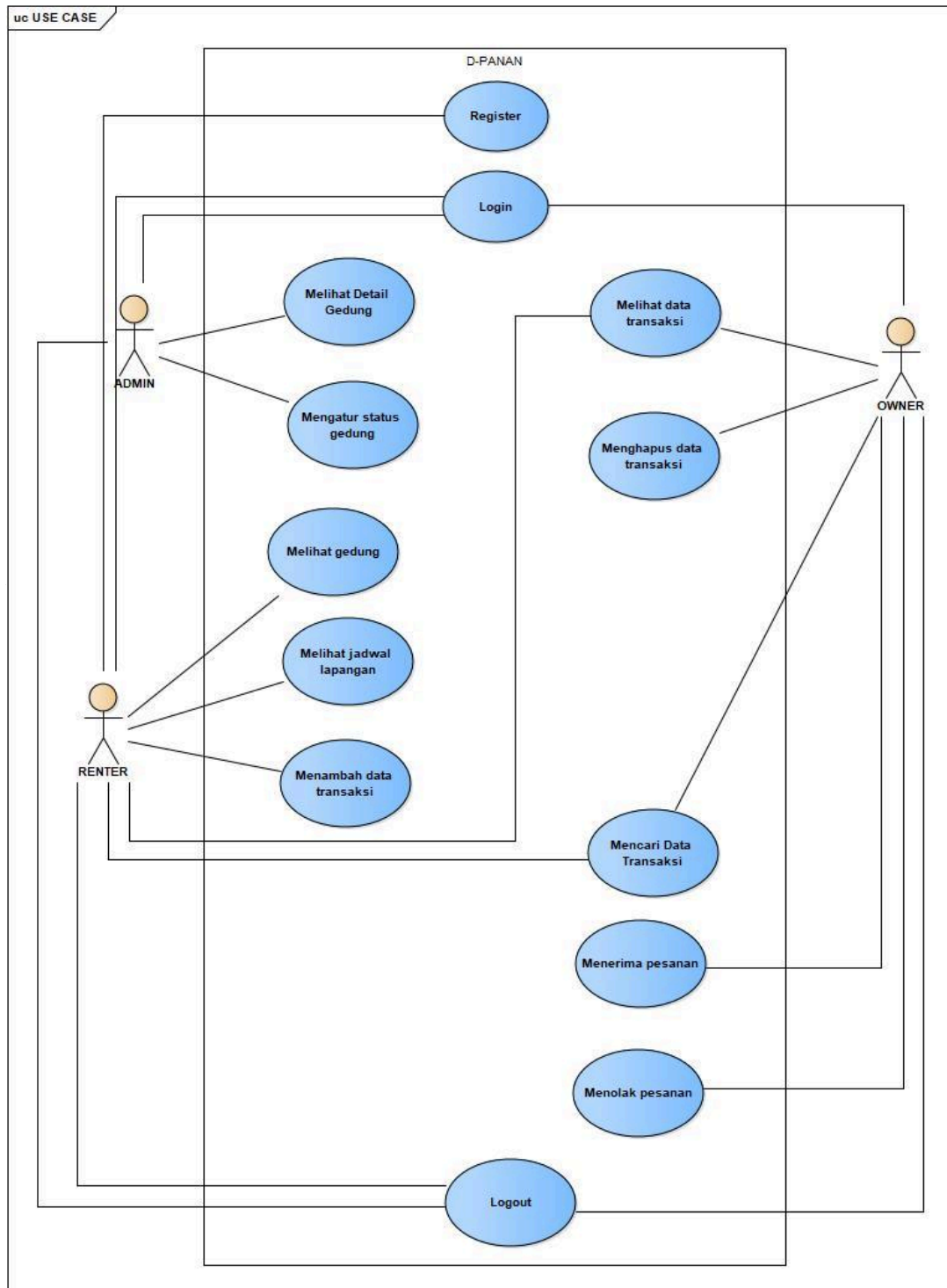
Data Gedung		
NO	ATTRIBUTE	TYPE DATA
1.	id_gedung (pk)	int
2.	nama_gedung	varchar(100)
3.	user_id (fk)	int
4.	nomor_telepon	varchar(50)

5.	deskripsi_gedung	text
6.	an_rek	text
7.	no_rek	text
8.	harga_lapangan	int
9.	gambar_gedung	varchar(200)
10.	status	varchar(50)
11.	slug	varchar(200)

Data Detail Gedung		
NO	ATTRIBUTE	TIPE DATA
1.	id_detail (pk)	int
2.	nama_lapangan	varchar(100)
3.	gedung_id (fk)	int

Data Jam		
NO	ATTRIBUTE	TIPE DATA
1.	id_jam (pk)	int
2.	jam_sewa	varchar(50)

2.3 USE CASE DIAGRAM



BAB III IMPLEMENTASI

3.1 Function/Object

Untuk Function dan Object, disini kami menerapkan OOP (Object Oriented Programming) yang diimplementasikan pada Controller dan Model. Contohnya yakni misal pada transaksi_model.php berikut.

```
11 references | 0 implementations | You, 3 hours ago | 1 author (You)
class Transaksi
{
    1 reference | 0 overrides
    static function updateTerima($id_transaksi)
    {
        global $conn;
        $status = 'diterima';
        $stmt = $conn->prepare("UPDATE transaksi set status=? WHERE id_transaksi=" . $id_transaksi);
        $stmt->bind_param("s", $status);
        $stmt->execute();
        $result = $stmt->affected_rows > 0 ? true : false;
        $stmt->close();
        return $result;
    }
    You, 3 hours ago • feat: add delete transaksi for owner

    1 reference | 0 overrides
    static function updateTolak($id_transaksi)
    {
        global $conn;
        $status = 'ditolak';
        $stmt = $conn->prepare("UPDATE transaksi set status=? WHERE id_transaksi=" . $id_transaksi);
        $stmt->bind_param("s", $status);
        $stmt->execute();
        $result = $stmt->affected_rows > 0 ? true : false;
        $stmt->close();
        return $result;
    }

    1 reference | 0 overrides
    static function selectTransaksiUser($username)
    {
        global $conn;
        $sql = "SELECT id_transaksi, username, tanggal, nama_gedung, jam_sewa, nama_lapangan, bukti_transfer, transaksi.status FROM `transaksi`
                INNER JOIN users ON user_id = id_user
                INNER JOIN detail_gedung ON detail_id = id_detail
                INNER JOIN gedung ON gedung_id = id_gedung
                INNER JOIN jam ON jam_id = id_jam
                WHERE username = '$username' ORDER BY id_transaksi DESC";
        $result = mysqli_query($conn, $sql);
        $data = array();
        if ($result->num_rows > 0) {
            while ($a = $result->fetch_array()) {
                $data[] = $a;
            }
        }
        return $data;
    }
}
```

Kami membuat class Transaksi pada model yang kemudian dipanggil sebagai object di transaksi_controller.php seperti berikut.

```

0 references | 0 implementations | You, 3 hours ago | 1 author (You)
class TransaksiController
{
    1 reference | 0 overrides
    static function detailTransaksi() {
        if (!isset($_SESSION['user'])) {
            header('Location: '.BASEURL.'auth?auth=false');
            exit;
        } else {
            if ($_SESSION['user']['role'] === 'renter') {
                view('renter/layout', [
                    'url' => 'detail_pesanan',
                    'transaksi' => Transaksi::selectTransaksiUser($_SESSION['user']['username'], )
                ]);
            } else {
                header('Location: '.BASEURL.'auth?auth=false');
                exit;
            }
        }
    }
}

```

Pada controller tersebut, pada function detailTransaksi, terdapat inisiasi model Transaksi, yakni pemanggilan Transaksi::selectTransaksiUser(). Contoh diatas merupakan salah satu contoh penerapan Function and Object yang kami terapkan.

3.2 Array

Untuk Array, banyak implementasi dari array yang kami terapkan di program D-PANAN ini. Salah satunya yakni ketika Model membaca data dari database, kemudian mereturn data berupa array, seperti berikut.

```

2 references | 0 overrides
static function selectTransaksiGedung($gedung)
{
    global $conn;
    $sql = "SELECT id_transaksi, username, tanggal, nama_gedung, jam_sewa, nama_lapangan, bukti_transfer, transaksi.status FROM `transaksi`
        INNER JOIN users ON user_id = id_user
        INNER JOIN detail_gedung ON detail_id = id_detail
        INNER JOIN gedung ON gedung_id = id_gedung
        INNER JOIN jam ON jam_id = id_jam
        WHERE slug = '$gedung' AND transaksi.status = 'diterima' ORDER BY id_transaksi DESC";
    $result = mysqli_query($conn, $sql);
    $data = array();
    if ($result->num_rows > 0) {
        while ($a = $result->fetch_array()) {
            $data[] = $a;
        }
    }
    return $data;
}

```

Function selectTransaksiGedung() tersebut menyimpan hasil querynya pada array dengan nama \$data dan kemudian mereturnnya. Return dari function ini kemudian ditangkap oleh controller seperti berikut.

```

1 reference | 0 overrides
static function detailGedung() {
    if (!isset($_SESSION['user'])) {
        header('Location: '.BASEURL.'auth?auth=false');
        exit;
    } else {
        if ($_SESSION['user']['role'] === 'renter') {
            view('renter/layout', [
                'url' => 'detail_gedung',
                'detailGedung' => Gedung::selectSatuGedung($_GET['gedung']),
                'lapangan' => Gedung::selectLapangan($_GET['gedung']),
                'jam' => Gedung::selectJam(),
                'transaksi' => Transaksi::selectTransaksiGedung($_GET['gedung']),
            ]);
        } else {
            header('Location: '.BASEURL.'auth?auth=false');
            exit;
        }
    }
}

```

You, yesterday • feat: add read feature for renter

Setelah ditangkap oleh controller, kemudian function ini dikirim ke View dan diinisiasi sebagai variabel 'transaksi' untuk menjadi sebuah variabel(array) yang akan diolah sebagai data pada View role Renter dengan file detail_gedung.php seperti berikut.

```

You, yesterday • feat: add read feature for renter
<?php foreach ($jam as $j) : ?>
    <tr>
        <td class="border-2 border-[#ffffff]"><?=$j['jam_sewa'] ?></td>
        <?php foreach ($lapangan as $lp) : ?>
            <td class="w-[250px] border-2 border-[#ffffff]">
                <?php
                foreach ($transaksi as $tr) {
                    if (isset($_GET['tanggal'])) {
                        if ($tr['tanggal'] == $_GET['tanggal']) {
                            if ($tr['jam_sewa'] == $j['jam_sewa']) {
                                if ($tr['nama_lapangan'] == $lp['nama_lapangan']) {
                                    if ($tr['username'] == $_SESSION['user']['username']) {
                                        echo ($tr['username']);
                                    } else {
                                        echo ('<p class="text-red-600 font-bold">Sudah Terisi!<p>');
                                    }
                                }
                            }
                        }
                    }
                }
            </td>
        <?php endforeach ?>
    </tr>
<?php endforeach ?>

```

Variabel \$transaksi ini diolah sebagai table data (td) pada View.

3.3 Date Time and File Handler

- Date Time

Untuk Date Time, kami banyak menerapkan implementasi ketika berurusan dengan transaksi, karena saat melakukan transaksi, perlu memasukkan data tanggal. Seperti berikut misalnya.

```
<form action="" method="get">
  <input id="gedung" type="hidden" name="gedung" value="<?= $_GET
    ['gedung'] ?>">
  <input id="tanggal" class="cari-data" type="text" value="<?= isset
    ($_GET['tanggal']) ? $_GET['tanggal'] : '' ?>" name="tanggal"
    size="10" placeholder="Masukkan Tanggal" onfocus="(this.
    type='date')" autocomplete="off" id="tanggal" />
  <button id="buttonCari" type="submit" class="btnn-cari">Cari Data</
    button>
</form>
```

Pada form input tersebut, terdapat input dengan type text, namun ketika onfocus berubah menjadi type date. Hal ini dilakukan agar dapat memberikan placeholder pada input, namun ketika diklik dan terfokus, maka akan berubah menjadi type date. Input ini akan memberikan value type date yang kemudian dikirim menggunakan form method GET sebagai keyword untuk melakukan pencarian transaksi berdasarkan tanggal.

- File Handler

Untuk File Handling, kami terapkan saat User Renter (Penyewa) melakukan add atau tambah data transaksi. Untuk bisa menambahkan data maka Renter perlu menambahkan foto sebagai bukti transaksi. Pada View renter submit form dengan input type file.

```
<label for="bukti_transfer">Bukti Transfer : </label>
<input type="file" name="bukti_transfer" id="bukti_transfer" required>
```

Kemudian file akan dikirimkan ke controller.

```
1 reference | 0 overrides
static function saveAdd()
{
    if (!isset($_SESSION['user'])) {
        header('Location: ' . BASEURL . 'auth?auth=false');
        exit;
    } else {
        $bukti_transfer = Transaksi::upload();
        if (!$bukti_transfer) {
            return false;
        }
        foreach ($_POST['pilihJamLap'] as $a) {
            $pieces = explode("_", $a);
            $idJam = $pieces[0];
            $idDetail = $pieces[1];
            $transaksi = Transaksi::create([
                'detail_id' => $idDetail,
                'jam_id' => $idJam,
                'user_id' => $_SESSION['user']['id_user'],
                'tanggal' => $_POST['tanggal'],
                'bukti_transfer' => $bukti_transfer,
            ]);
        }
        echo "<script>
        alert('Transaksi berhasil! Tunggu pesananmu diproses oleh admin');
        window.location.href = '".BASEURL.'"dashboard/pesanan-saya';
        </script>";
    }
}
```

Kemudian controller memanggil function khusus untuk file handling pada transaksi_model, yakni upload(). Berikut fungsinya.

```

1 reference | 0 overrides
static function upload()
{
    $namaFile = $_FILES["bukti_transfer"]["name"];
    $ukuranFile = $_FILES["bukti_transfer"]["size"];
    $error = $_FILES["bukti_transfer"]["error"];
    $tmpName = $_FILES["bukti_transfer"]["tmp_name"];

    //cek apakah ada gambar yang di upload
    if ($error !== 4) {
        // cek apakah yang diupload gambar
        $ekstensiGambarValid = ['jpg', 'jpeg', 'png'];
        $ekstensiGambar = explode('.', $namaFile);
        $ekstensiGambar = strtolower(end($ekstensiGambar));
        if (!in_array($ekstensiGambar, $ekstensiGambarValid)) {
            echo "
            <script>
                alert('Upload hanya gambar, jangan file lain!')
                window.location.href = '' . BASEURL . "dashboard/pesanan-saya";
            </script>
            ";
            return false;
        } else if (in_array($ekstensiGambar, $ekstensiGambarValid)) {
            // cek jika ukurannya terlalu besar
            if ($ukuranFile > 1000000) {
                echo "
                <script>
                    alert('Ukuran gambar terlalu besar!')
                    window.location.href = '' . BASEURL . "dashboard/pesanan-saya";
                </script>
                ";
                return false;
            } else {
                // generate nama file baru
                $namaFileBaru = uniqid();
                $namaFileBaru .= "." . $ekstensiGambar;
                // Lolos pengecekan, gambar siap diupload
                move_uploaded_file($tmpName, 'img/bukti_transaksi/' . $namaFileBaru);
                return $namaFileBaru;
            }
        }
    } else {
        return true;
    }
}

```

Function ini khusus untuk menangani file yang diupload oleh user. Pada function ini ada beberapa pengecekan, pertama yakni mengecek apakah yg diupload adalah gambar atau bukan. Ekstensi yang diperbolehkan yakni jpg, jpeg, dan png. Kemudian dilakukan juga pengecekan ukuran file, Jika ukuran file melebihi 1 MB atau 1.000.000 byte maka akan ditolak. Kemudian jika berhasil melewati 2 kondisi tersebut, maka file yg diupload akan dibuatkan nama file baru menggunakan uniqid() dan disimpan pada path yang telah ditentukan.

3.4 MySQL

Untuk relasi pada MySQL sudah dijelaskan pada bagian ERD. Kami memiliki 6 table yang semuanya saling berelasi. Kemudian untuk data, Pada Users saat ini kami memiliki 11 user dengan role yang berbeda-beda. Berikut gambarnya

		id_user	email	username	password	role
<input type="checkbox"/>	Ubah Salin Hapus	1	admin@gmail.com	admin	\$2y\$10\$VHdxkET/AwPf.07S8FUaL...0kt7TDh9CqY2EidSG2Lg...	admin
<input type="checkbox"/>	Ubah Salin Hapus	2	futsalindo@gmail.com	futsalindo	\$2y\$10\$yI02kh1.TG46mdqEuy2L3ux5/FglXzAeHkmlzwVUaXJ...	owner
<input type="checkbox"/>	Ubah Salin Hapus	3	zonafutsal@gmail.com	zonafutsal	\$2y\$10\$txzNAVeIS8y1OHNC/YC1luHYQWZKlyuEOtQyIOS/eAz...	owner
<input type="checkbox"/>	Ubah Salin Hapus	4	elphasindo@gmail.com	elphasindo	\$2y\$10\$hU4ji1PaATkVZW76WzTRP.O2jW5BtAuks/bM0FjQc9H...	owner
<input type="checkbox"/>	Ubah Salin Hapus	5	lapangan8@gmail.com	lapangan8	\$2y\$10\$9plzcEjvoOall5fLreMva.th0PKWq1C6pAfBEyWAF03...	owner
<input type="checkbox"/>	Ubah Salin Hapus	6	argopuro@gmail.com	argopuro	\$2y\$10\$LC2F8P70kvyFGeAapvs4w.fnHhU6Si83dSkhdCr2j.Q...	owner
<input type="checkbox"/>	Ubah Salin Hapus	7	hamzah@gmail.com	hamzah	\$2y\$10\$XbgXKN9gGr4GTlbv7Jnz4eKi4ilgHZ0PShkgmZ57N2o...	renter
<input type="checkbox"/>	Ubah Salin Hapus	9	ainol@gmail.com	ainol	\$2y\$10\$tdvo/i40WqAlbrPK74Qb2etHLgqjKmnO/y0qZ9Dofnt...	renter
<input type="checkbox"/>	Ubah Salin Hapus	10	edok@gmail.com	edok	\$2y\$10\$MpEc1qjJ.gnHUSG.vLKglOrKcGgtf8Ltp1djfpdSVGi...	renter
<input type="checkbox"/>	Ubah Salin Hapus	11	kenol7910@gmail.com	azam	\$2y\$10\$6beqAYDemC4bCMz/Q/Lr..1XnN08KQhiYGTNTCWT20...	renter

Kemudian untuk data transaksi, saat ini kami memiliki lebih dari 300 dummy data yang telah kami siapkan. Berikut lampirannya.

		id_transaksi	detail_id	jam_id	user_id	tanggal	bukti_transfer	status
<input type="checkbox"/>	Ubah Salin Hapus	310	28	6	11	2024-06-22	6669b02f25b39.jpeg	diterima
<input type="checkbox"/>	Ubah Salin Hapus	309	27	6	11	2024-06-22	6669b02f25b39.jpeg	diterima
<input type="checkbox"/>	Ubah Salin Hapus	308	29	5	11	2024-06-22	6669b02f25b39.jpeg	diterima
<input type="checkbox"/>	Ubah Salin Hapus	307	28	5	11	2024-06-22	6669b02f25b39.jpeg	diterima
<input type="checkbox"/>	Ubah Salin Hapus	306	27	5	11	2024-06-22	6669b02f25b39.jpeg	diterima

3.5 PHP MySQL

- **Create**

Untuk implementasi PHP MySQL Create kami terapkan ketika ada renter yang ingin menambahkan jadwal lapangan. Untuk implementasinya kami buat di transaksi_model yakni sebagai berikut.


```

1 reference | 0 overrides
static function create($data)
{
    global $conn;
    $tanggal = htmlspecialchars($data['tanggal']);
    $idJam = htmlspecialchars($data['jam_id']);
    $idDetail = htmlspecialchars($data['detail_id']);
    $idUser = htmlspecialchars($data['user_id']);
    $status = 'diproses';
    $bukti = $data['bukti_transfer'];

    $sql = "INSERT INTO transaksi(detail_id, jam_id, user_id, tanggal, bukti_transfer, status) VALUES (?, ?, ?, ?, ?, ?)";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param('iiisss', $idDetail, $idJam, $idUser, $tanggal, $bukti, $status);
    $stmt->execute();
    $result = $stmt->affected_rows > 0 ? true : false;
    $stmt->close();
    return $result;
}

```

Untuk function insert yang kami buat sudah menerapkan statement dan bind_param sehingga aman terhadap serangan SQL Injection.

- **Read**

Untuk PHP MySQL Read kami terapkan pada beberapa function, seperti saat ingin selectTransaksiUser, selectTransaksiGedung, dan sebagainya. Berikut contohnya.

```

1 reference | 0 overrides
static function selectTransaksiUser($username)
{
    global $conn;
    $sql = "SELECT id_transaksi, username, tanggal, nama_gedung, jam_sewa, nama_lapangan, bukti_transfer, transaksi.status FROM `transaksi`
        INNER JOIN users ON user_id = id_user
        INNER JOIN detail_gedung ON detail_id = id_detail
        INNER JOIN gedung ON gedung_id = id_gedung
        INNER JOIN jam ON jam_id = id_jam
        WHERE username = '$username' ORDER BY id_transaksi DESC";
    $result = mysqli_query($conn, $sql);
    $data = array();
    if ($result->num_rows > 0) {
        while ($a = $result->fetch_array()) {
            $data[] = $a;
        }
    }
    return $data;
}

```

Pada Read ini, kami memanfaatkan relasi yang telah dibuat menggunakan foreign key. Kami gunakan INNER JOIN agar bisa menggabungkan beberapa tabel sehingga bisa menampilkan hasil yang kami inginkan.

- **Update**

Untuk PHP MySQL Update kami terapkan ketika admin menonaktifkan data gedung atau ketika owner memilih untuk menerima atau menolak transaksi. Berikut contohnya pada model transaksi_model.php

```

class Transaksi
{
    1 reference | 0 overrides
    static function updateTerima($id_transaksi)
    {
        global $conn;
        $status = 'diterima';
        $stmt = $conn->prepare("UPDATE transaksi set status=? WHERE id_transaksi=" . $id_transaksi);
        $stmt->bind_param("s", $status);
        $stmt->execute();
        $result = $stmt->affected_rows > 0 ? true : false;
        $stmt->close();
        return $result;
    }

    1 reference | 0 overrides
    static function updateTolak($id_transaksi)
    {
        global $conn;
        $status = 'ditolak';
        $stmt = $conn->prepare("UPDATE transaksi set status=? WHERE id_transaksi=" . $id_transaksi);
        $stmt->bind_param("s", $status);
        $stmt->execute();
        $result = $stmt->affected_rows > 0 ? true : false;
        $stmt->close();
        return $result;
    }
}

```

Dua function ini berfungsi untuk melakukan update status pada transaksi, untuk menerima atau menolak transaksi yang diajukan oleh user.

- **Delete**

Untuk PHP MySQL Delete kami terapkan di owner ketika ingin menghapus transaksi. Berikut implementasinya.

```

1 reference | 0 overrides
static function delete($id_transaksi)
{
    global $conn;
    $query = "DELETE FROM transaksi WHERE id_transaksi=?";
    $stmt = $conn->prepare($query);
    $stmt->bind_param('i', $id_transaksi);
    $stmt->execute();
    $result = $stmt->affected_rows > 0 ? true : false;
    $stmt->close();
    return $result;
}

```

3.6 Form Validation

- Server Side

Server Side validation kami terapkan pada fitur auth(login dan register). Pertama alur dari View yakni sebagai berikut.

```
<div class="container-form login">
  <form action="{<?> urlpath('login') ?>" method="post">
    <h1>Login Akun</h1>
    <?php if (array_key_exists('username', $errLog)) : ?>
      <p style="color: red; font-size: 12px; font-weight: 600; text-align: left; margin: 0px 0px 0px 0px;"><?> $errLog['username'] ?></p>
    <?php endif ?>
    <input type="text" placeholder="Username" name="username" required />
    <?php if (array_key_exists('password', $errLog)) : ?>
      <p style="color: red; font-size: 12px; font-weight: 600; text-align: left; margin: 0px 0px 0px 0px;"><?> $errLog['password'] ?></p>
    <?php endif ?>
    <input type="password" required placeholder="Password" name="password" />
    <button>Login</button>
  </form>
</div>
```

Form input ini akan menerima inputan dari user berupa username dan password untuk login. Kemudian inputnya akan dikirim ke controller.

```
1 reference | 0 overrides
static function saveLogin() {
    $post = array_map('htmlspecialchars', $_POST);

    $user = User::login([
        'username' => $post['username'],
        'password' => $post['password']
    ]);
    if (array_key_exists('role', $user)) {
        unset($user['password']);
        $_SESSION['user'] = $user;
        header('Location: '.BASEURL.'auth?loginSuccess');
    }
    else {
        view('auth_page/layout', [
            'url' => 'auth',
            'errorLog' => $user
        ]);
    }
}
```

Pada controller tersebut, terdapat variabel \$user yang akan menjadi object dari model User::login() dengan parameter dari input POST user. Kemudian berikut modelnya.

```

class User {
    1 reference | 0 overrides
    static function login($data=[]) {
        extract($data);
        global $conn;
        $error = [];

        $result = $conn->query("SELECT * FROM users WHERE username = '$username'");
        if (mysqli_num_rows($result) === 1) {
            $result = $result->fetch_assoc();
            $hashedPassword = $result['password'];
            $verify = password_verify($password, $hashedPassword);
            if ($verify) { return $result; }
            else { $error += ['password' => 'Password salah!']; }
        }
        else {
            $error += ['username' => 'Username anda tidak ditemukan!'];
        }

        return $error;
    }
}

```

Pada model login ini menangkap \$data yang sudah dikirim oleh controller. Kemudian datanya diolah dalam sql. Jika terdapat error misalnya seperti username tidak ditemukan, atau password salah, maka errornya akan ditangkap dan disimpan pada variable \$error berupa array. Kemudian \$error ini akan direturn kembali pada controller. Kemudian dikirim kembali ke view, dan ditangkap oleh view oleh variabel \$errReg dan \$errLog berikut..

```

You, 2 hours ago | 1 author (You)
<?php
$errorReg = [];
$errorLog = [];

if (isset($errorReg)) {
    $errReg = $errorReg;
}

if (isset($errorLog)) {
    $errLog = $errorLog;
}
?>

<div class="container" id="container">
    <div class="container-form register">
        <form action="<?= urlpath('register') ?>" method="post">
            <h1>Register Akun</h1>
            <input type="hidden" name="role" value="rener">
            <?php if (array_key_exists('email', $errReg)) : ?>

```

- **Client Side**

Untuk Client-Side Validation yang kami terapkan yakni required pada beberapa kolom input dan menggunakan input type="email" agar ketika ada input email user perlu menyertakan '@' agar kolom input bisa diproses. Berikut contohnya.

```
<input type="text" required placeholder="Username" name="username" />
```

```
<input type="email" required placeholder="Email" name="email" />
```

3.7 User Authentication

- **Session**

Session adalah bagian penting untuk authentication karena bisa menyimpan data berupa cookies namun disimpan di server. Sehingga session ini bagus untuk dimanfaatkan untuk pengautorisasian/pemberian role dengan hak akses yang berbeda-beda. Session kami terapkan ketika ada user yang berhasil login. Berikut implementasinya.

```
$user = User::login([
    'username' => $post['username'],
    'password' => $post['password']
]);
if (array_key_exists('role', $user)) {
    unset($user['password']);
    $_SESSION['user'] = $user;
    header('Location: '.BASEURL.'auth?loginSuccess');
}
```

Jadi ketika ada user yang berhasil login, maka buat \$_SESSION berdasarkan array dari \$user yang ditemukan. Kemudian pada main.php yang memiliki semua controller, kami lakukan session_start() agar di setiap page yang ada secara otomatis melakukan session_start()

```
1 <?php You, 7 days ago * first commit
2 include_once 'dash_controller.php';
3 include_once 'transaksi_controller.php';
4 include_once 'auth_controller.php';
5 include_once 'gedung_controller.php';
6
7 session_start();
```

Kemudian ketika user melakukan logout, maka lakukan `session_destroy()` untuk menghancurkan data session yang sudah dibuat. Kemudian kembalikan user ke BASEURL.

```
1 reference | 0 overrides
static function logout() {
    $_SESSION = array();

    if (ini_get("session.use_cookies")) {
        $params = session_get_cookie_params();
        setcookie(session_name(), '', time() - 42000,
            $params["path"], $params["domain"],
            $params["secure"], $params["httponly"]
        );
    }

    session_destroy();
    header('Location: '.BASEURL);
}
```

- Login

Login controller akan memanggil model `User::login` yang ada untuk melakukan pengecekan apakah ada user yang memiliki username dan password yang sesuai dengan input POST user.

```
1 reference | 0 overrides
static function saveLogin() { You, 7 days ago • first
    $post = array_map('htmlspecialchars', $_POST);

    $user = User::login([
        'username' => $post['username'],
        'password' => $post['password']
    ]);
    if (array_key_exists('role', $user)) {
        unset($user['password']);
        $_SESSION['user'] = $user;
        header('Location: '.BASEURL.'auth?loginSuccess');
    }
    else {
        view('auth_page/layout', [
            'url' => 'auth',
            'errorLog' => $user
        ]);
    }
}
```

Kemudian berikut model dari User::login

```
2 references | 0 implementations | You, 7 days ago | 1 author (You)
class User {
    1 reference | 0 overrides
    static function login($data=[]) {
        extract($data);
        global $conn;
        $error = [];

        $result = $conn->query("SELECT * FROM users WHERE username = '$username'");
        if (mysqli_num_rows($result) === 1) {
            $result = $result->fetch_assoc();
            $hashedPassword = $result['password'];
            $verify = password_verify($password, $hashedPassword);
            if ($verify) { return $result; }
            else { $error += ['password' => 'Password salah!']; }
        }
        else {
            $error += ['username' => 'Username anda tidak ditemukan!'];
        }

        return $error;
    }
}
```

- Register

Register controller akan memanggil User::register yang ada untuk melakukan query INSERT INTO agar data yang sudah diinputkan oleh POST user bisa masuk ke dalam database.

```
1 reference | 0 overrides
static function saveRegister() {
    $post = array_map('htmlspecialchars', $_POST);

    $user = User::register([
        'email' => $post['email'],
        'username' => $post['username'],
        'password' => $post['password'],
        'role' => $post['role']
    ]);

    if ($user === true) {
        header('Location: '.BASEURL.'auth?registerSuccess');
    } elseif ($user === false){
        header('Location: '.BASEURL.'auth?registerFailed');
    }
    else {
        // var_dump($user);
        // header('Location: '.BASEURL.'auth');
        view('auth_page/layout', [
            'url' => 'auth',
            'errorReg' => $user
        ]);
    }
}
```

Dan berikut modelnya.

```
1 reference | 0 overrides
static function register($data=[]) {
    extract($data);
    global $conn;
    $error = [];
    $email = strtolower($email);
    $username = strtolower($username);
    $hashedPassword = password_hash($password, PASSWORD_DEFAULT);
    $role = strtolower($role);

    if ($email === "" || $username === "" || $password === ""){
        return false;
    } else {
        $usr = $conn->query("SELECT * FROM users WHERE username = '$username'");
        $eml = $conn->query("SELECT * FROM users WHERE email = '$email'");
        if (mysqli_num_rows($usr) > 0) {
            $error += ['username' => 'Username sudah ada!'];
        }

        if (mysqli_num_rows($eml) > 0) {
            $error += ['email' => 'Email sudah ada!'];
        }

        if (strlen($password) < 8){
            $error += ['password' => 'Password minimal 8 karakter!'];
        }

        if ($error != []) {
            return $error;
        } else {
            $sql = "INSERT INTO users SET username = ?, email = ?, password = ?, role = ?";
            $stmt = $conn->prepare($sql);
            $stmt->bind_param('ssss', $username, $email, $hashedPassword, $role);
            $stmt->execute();

            $result = $stmt->affected_rows > 0 ? true : false;

            return $result;
        }
    }
}
```


3.8 AJAX (Asynchronous JavaScript and XMLHttpRequest)

Untuk penerapan AJAX sendiri, kami menggunakan bantuan library jQuery pada javascript. Kami menggunakan jQuery ini agar syntax yang ditulis tidak terlalu panjang dan lebih memudahkan saat development process. Kemudian untuk implementasi dari AJAX ini yakni saat proses lihat jadwal gedung oleh renter. Saat lihat jadwal gedung, awalnya tabel yang tampil akan kosong, user perlu memasukkan tanggal yang ingin dicek terlebih dahulu, kemudian baru akan muncul tabel jadwal jam dan nomor lapangan. Hal ini diperoleh dengan memanfaatkan AJAX. Seperti berikut code awalnya.

```
<div class="tabel-k">
  <table cellpadding="0">
    <tr>
      <th class="w-[250px] border-2 border-[#ffffff]"></th>
      <?php foreach ($lapangan as $lap) : ?>
        <th class="border-2 border-[#ffffff]"><?= $lap['nama_lapangan'] ?></th>
      <?php endforeach ?>
    </tr>
  </table>
</div>
</div>
```

Bisa dilihat tabel diatas tidak berisi apa-apa awalnya. Hanya berisi th atau table head. Kemudian berikut input tanggal dan tombol cari untuk renter.

```
<form action="" method="POST" id="jadwal-gedung" >
  <input id="gedung" type="hidden" name="gedung" value="<?= $_GET['gedung'] ?>">
  <input id="tanggal" class="carl-data" type="text" value="" name="tanggal" size="10" placeholder="Masukkan Tanggal" onfocus="(this.type='date')" autocomplete="off"
  id="tanggal" />
  <button id="buttonCari" type="submit" class="btnn-carl">Cari Data</button>
</form>
```

Ketika button cari disubmit, maka page ini akan menjalankan script yang sudah disiapkan, yakni dengan menggunakan jQuery AJAX.

```

<script>
$(document).ready(function() {
    $('#create-data').hide();

    $('#buttonCari').click(function(e) {
        e.preventDefault();
        $('#create-data').show();

        var tanggal = $('#tanggal').val();
        var gedung = $('#gedung').val();
        var url = '<?=' + urlpath("dashboard/ajax?gedung=") + '>' + gedung + '&tanggal=' + tanggal;

        $.ajax({
            type: 'GET',
            url: url,
            success: function(response) {
                $('#tabel-k').html(response);
            }
        });
    });
});

```

Code ini berfungsi untuk mengirimkan request ketika ada buttonCari yang di click. Request dikirimkan ke route dashboard/ajax dengan parameter \$_GET untuk gedung dan tanggal agar page dashboard/ajax ini bisa mengolah data yang dikirimkan dan mencari jadwal sesuai dengan tanggal dan gedung yang diminta oleh renter. Kemudian jika request yang dikirim dan sukses, maka response dari page dashboard/ajax ini akan digunakan untuk mengganti/mereplace table kosong yang telah dimiliki di awal, diganti dengan table yang sudah berisi jadwal gedung pada tanggal yang telah dipilih.

3.9 JavaScript

Javascript kami gunakan untuk melakukan manipulasi terhadap element-element html dengan menggunakan DOM pada Javascript. Misal menambahkan class active ketika ada button yang di klik sehingga bisa mempengaruhi class pada styling CSS nya sehingga bisa membuat web lebih menarik akibat animasi dan transisi yang dihasilkan. Berikut contohnya.

```

<script>
const container = document.getElementById('container');
const registerBtn = document.getElementById('register');
const loginBtn = document.getElementById('login');
const searchParams = window.Location.pathname.split('/');

if (searchParams.includes('register')) {
  container.classList.add("active")
} else {
  container.classList.remove("active")
}

registerBtn.addEventListener('click', () => {
  container.classList.add('active');
});

loginBtn.addEventListener('click', () => {
  container.classList.remove('active');
});
</script>

```

3.10 CSS

Untuk CSS sendiri kami gunakan untuk melakukan styling pada elemen HTML. Disini kami menggunakan 2 framework, yakni Bootstrap dan Tailwind. Serta menggunakan native CSS juga. Berikut implementasinya.

```

You, 6 days ago | 1 author (You)
@import url('https://fonts.googleapis.com/css2?family=Montserrat:wght@300;400;500;600;700&display=swap');

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Montserrat', sans-serif;
}

| You, 7 days ago • first commit
body {
  background-color: #c9d6ff;
  background: linear-gradient(to right, #e2e2e2, #aafeff);
  display: flex;
  align-items: center;
  justify-content: center;
  flex-direction: column;
  height: 100vh;
}

.container {
  background-color: #fff;
  border-radius: 30px;
  box-shadow: 0 5px 15px rgba(0, 0, 0, 0.35);
  position: relative;
  overflow: hidden;
  width: 880px;
  max-width: 100%;
  height: 550px;
}

```

CSS Native

```


<div class="kiri mx-[100px] mt-[100px]">
    <div class="main-text flex flex-col gap-[30px] justify-center bg-[#2d9596] border-solid border-[2px] border-[#ffffff80] text-white rounded-[20px] w-[50vw] h-[70vh] backdrop-blur-[20px] p-[100px]">
      <h2 class="text-[40px] font-medium">Boost Your Performance</h2>
      <h1 class="text-[60px] font-bold">Health and Fitness</h1>
      <p class="text-[20px] font-medium">Rent High-Quality Sports Venues to Elevate Your Performance.</p>
      <a href="<?>= unipath('dashboard') ?>" class="hover:text-white"><button class="bg-[#ffffff59] w-[200px] h-[50px] text-[20px] rounded-[10px] decoration-inherit hover:bg-[#2d9596] hover:text-[#ffffff]">Get Started</button></a>
    </div>
  </div>
  <div class="kanan mr-[200px] mt-[100px] flex flex-col gap-[70px]">
    <div class="bg-white p-[10px] rounded-[50%]"></div>
    <div class="bg-white p-[10px] rounded-[50%]"></div>
    <div class="bg-white p-[10px] rounded-[50%]"></div>
  </div>
</div>


```

Tailwind CSS

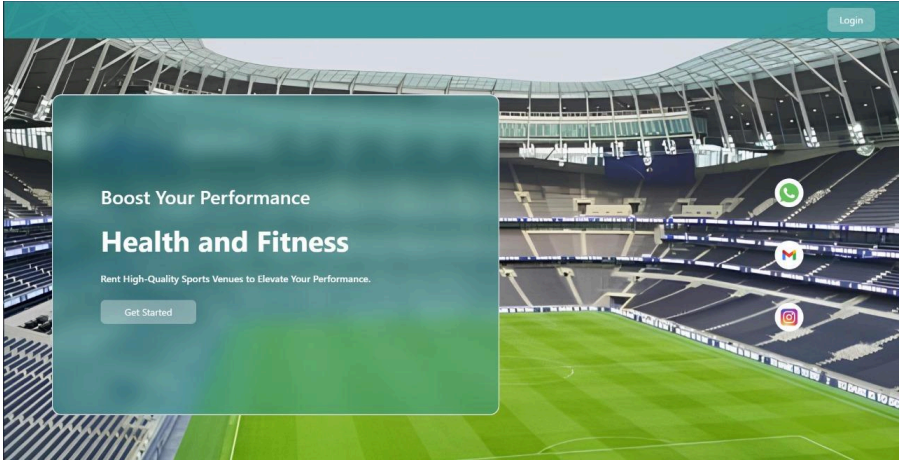
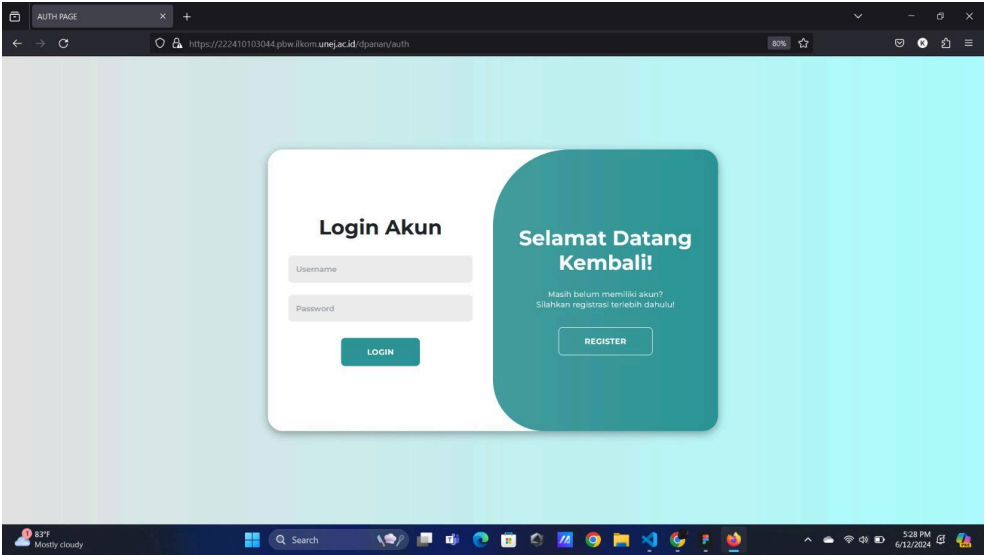
```

<!DOCTYPE html>
<html Lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script src="https://cdn.tailwindcss.com"></script>
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap-icons/font/bootstrap-icons.css" rel="stylesheet">
  <title><?= $title ?? '' ; ?></title>
  <style><?= $style ?? '' ; ?></style>
</head>
<body>
  <?= $body ?? '' ; ?>
</body>
</html>
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.1/dist/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

```

Import CDN Bootstrap dan Tailwind

3.11 User Interface

NO	ROLE
HOMEPAGE	
LOGIN	

1.

ADMIN

Dashboard ADMIN

The screenshot shows the Admin Dashboard for 'Sewa Lapangan'. The left sidebar contains links for 'Dashboard' and 'Data Gedung', and a 'Logout' button at the bottom. The main content area has a greeting 'Halo, Admin' and a section titled 'Lapangan yang Aktif' featuring a photo of an indoor soccer field. Below this is a 'Zona Futsal' section with a description: 'Zona Futsal merupakan gedung olahraga futsal yang memiliki beberapa fasilitas yang lengkap dan memiliki 7 lapangan yang tersedia. Gedung ini terletak di Jl. Hayam Wuruk No. 183, Karang Miuwo, Mangli, Kec. Kaliwates.' The Windows taskbar at the bottom shows the time as 5:39 PM on 6/12/2024.

Data Gedung

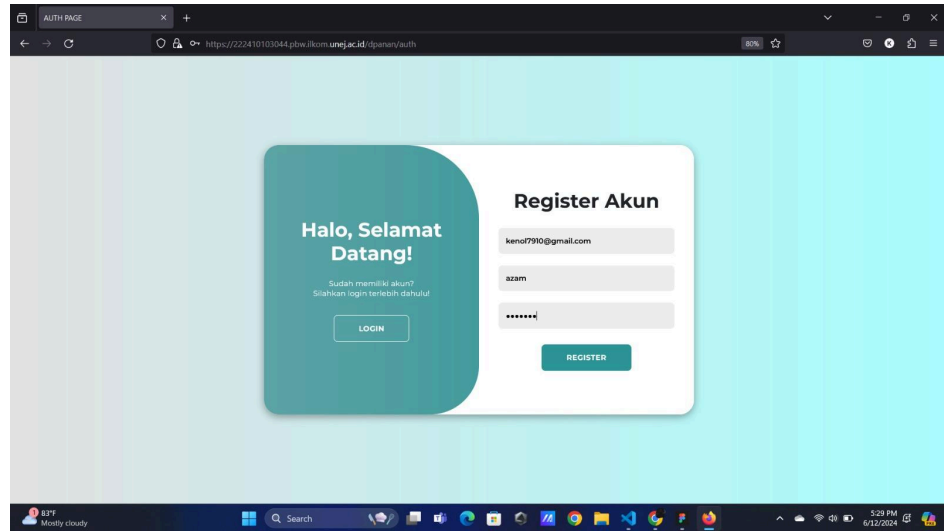
The screenshot shows the 'Data Gedung' table in the Admin Dashboard. The table lists five facilities with their respective details. The left sidebar is the same as in the previous screenshot. The Windows taskbar at the bottom shows the time as 5:39 PM on 6/12/2024.

No.	Nama Gedung	Nama Pemilik	Nomor Telepon	Jumlah Lapangan	Status	Aksi
1	Futsalindo	futsalindo	089787467389	6	active	
2	Zona Futsal	zonafutsal	087467864536	7	active	
3	Elphasindo	elphasindo	087648937462	5	active	
4	Lapangan 8	lapangan8	087964736523	8	active	
5	Argopuro	argopuro	089765473526	4	inactive	

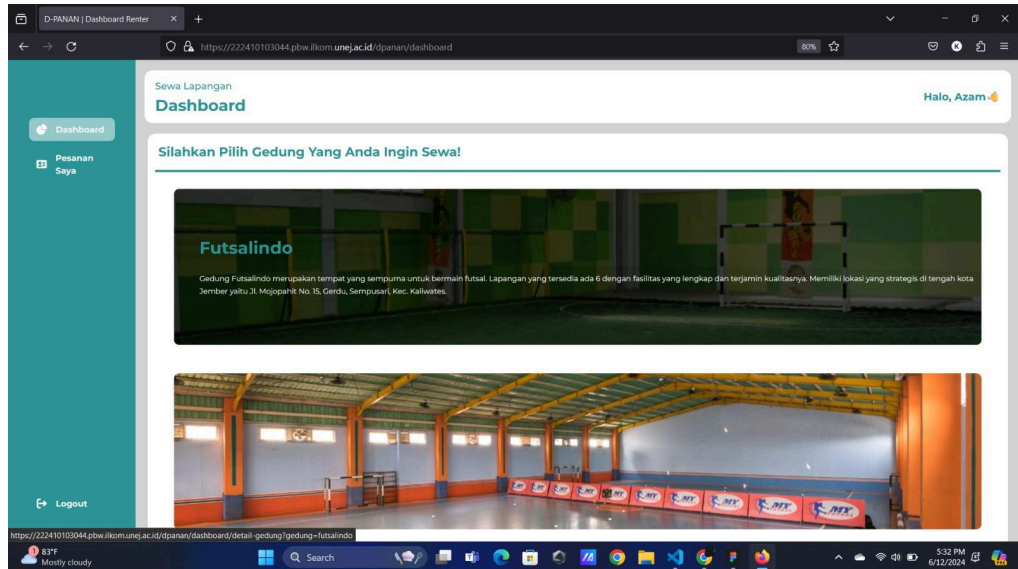
2.

RENTER

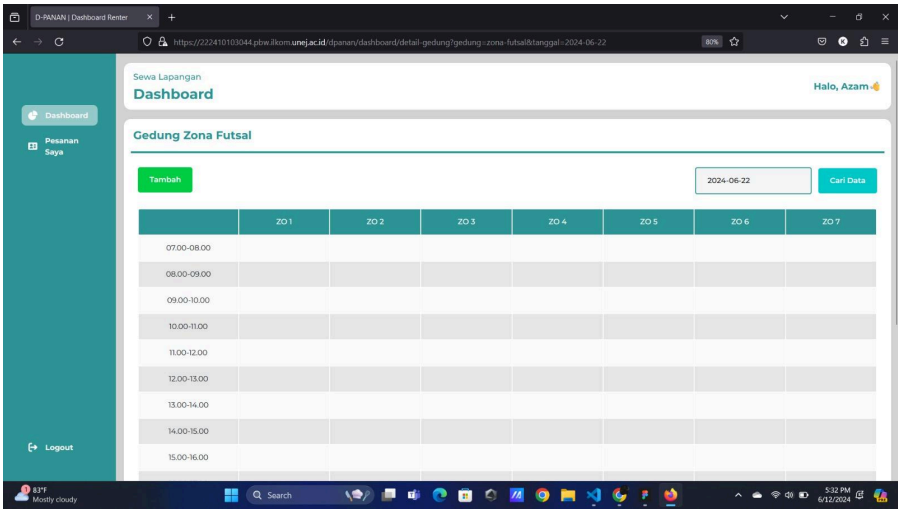
Register



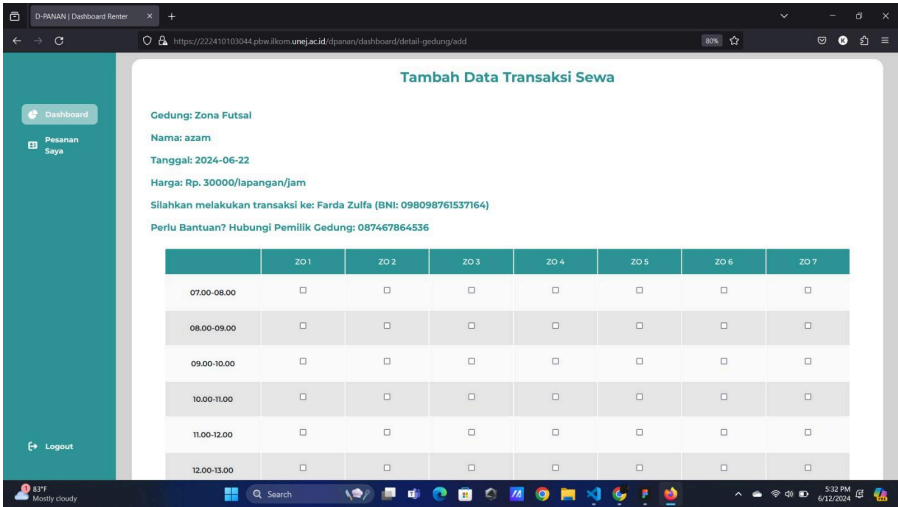
Dashboard RENTER



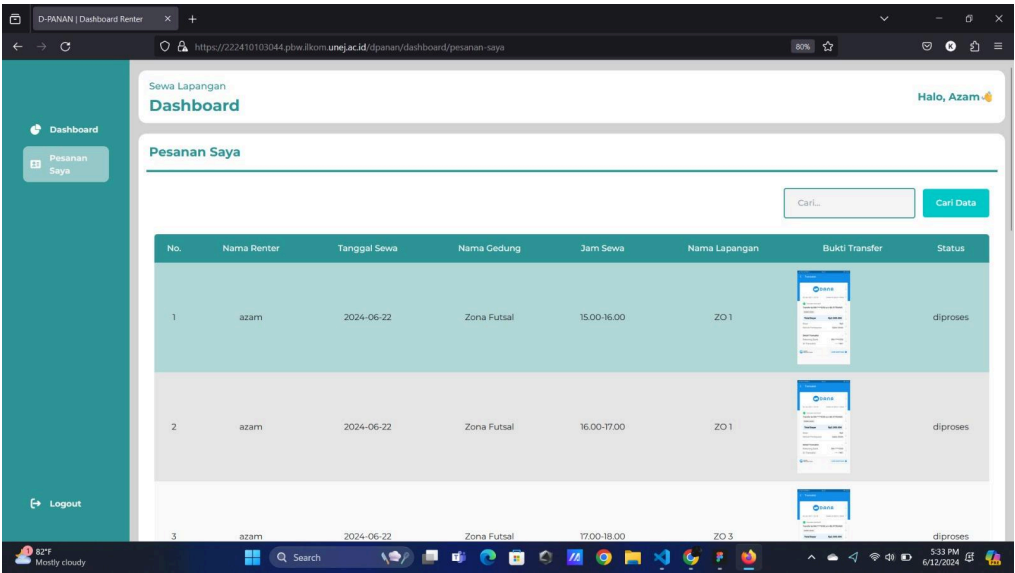
Halaman Melihat Jadwal dan Lapangan



Halaman Tambah Data Transaksi Sewa



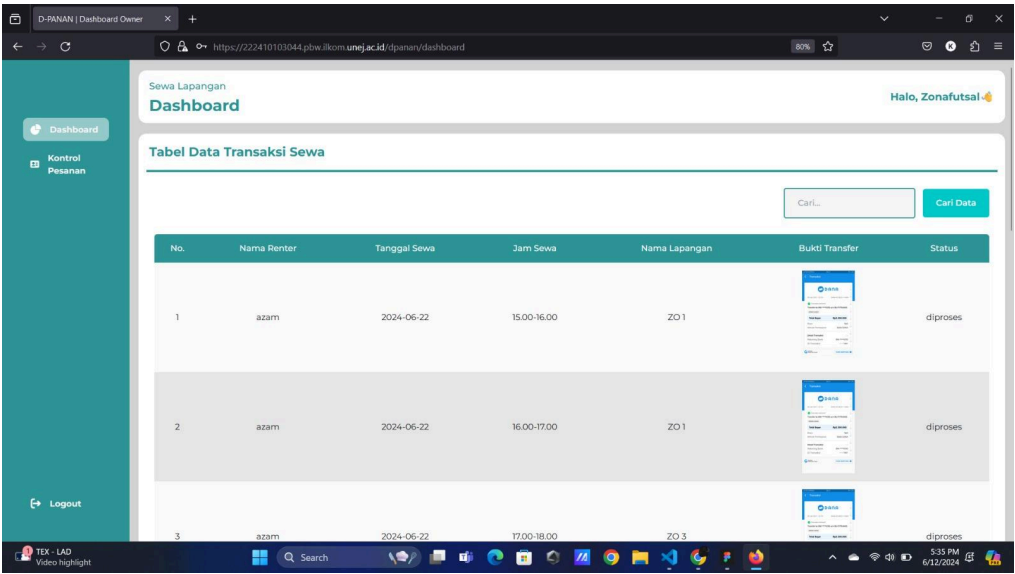
Halaman Pesanan Saya/Riwayat Pesanan



3.

OWNER

Dashboard OWNER



Halaman Kontrol Pesanan

Dashboard

Kontrol Pesanan

Logout

Tabel Data Transaksi Sewa

Cari...

Cari Data

No.	Nama Daster	Tanggal Sewa	Jam Sewa	Nama Lapangan	Bukti Transfer	Status	Terima/Tolak	Aksi
1.	edok	2024-06-27	13.00-14.00	LP 3		diterima	<div><div></div><div></div></div>	<div></div>
2.	edok	2024-06-27	13.00-14.00	LP 2		diterima	<div><div></div><div></div></div>	<div></div>
3.	edok	2024-06-27	13.00-14.00	LP 1		diterima	<div><div></div><div></div></div>	<div></div>

BAB IV PENUTUP

5.1 Kesimpulan

Laporan akhir mata kuliah Pemrograman Web ini menjelaskan tentang pembuatan sistem penyewaan lapangan olahraga berbasis website bernama “D-PANAN”. Laporan ini dilatarbelakangi oleh proses penyewaan lapangan olahraga yang kita ketahui perlu melakukan kunjungan langsung atau melalui telepon. Sistem ini bertujuan untuk memudahkan proses reservasi lapangan bagi penyewa dan pengelolaan transaksi lapangan bagi pengelola lapangan.

Kemudahan akses dan reservasi, keamanan dan transparansi, dan peningkatan efisiensi operasional menjadi keunggulan dari sistem kami. Sistem “D-PANAN” dapat berfungsi sebagai sarana yang berguna dalam mendukung pengelolaan penyewaan lapangan olahraga, literasi keuangan, dan pengembangan industri dalam sektor industri yang bergerak di bidang olahraga melalui pemanfaatan teknologi mutakhir.