

Laporan Pembuatan Web Sederhanan

Halaman login , Halaman Dasbord Admin Dan User

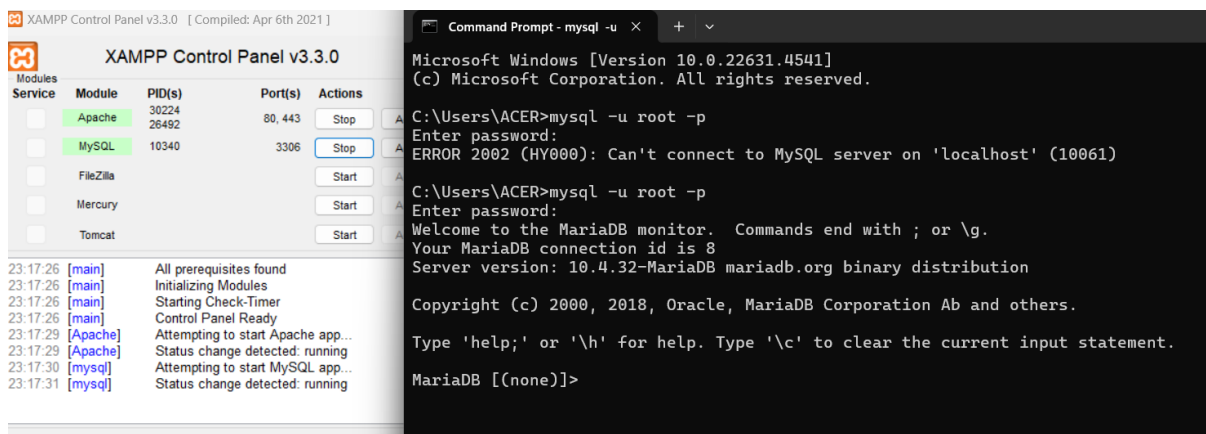
tool yang di gunakan

- visual studio kode
- xampp
- mysql

bahasa perograman yang di gunakan

- php
- html

Pembuatan database menggunakan mysql dengan terminal



Keren saya dalam pembuatan database saya menggunakan xampp maka saya harus menyalakan xampp nanya terlebih dahulu ke meudian baru saya menggunakan command prompt untuk mengakses mysql sebenarnya ada cara lain yakni menggunakan phpMyAdmin yang sudah di sediakan oleh xampp

Untuk masuk ke mysql menggunakan commad rompt dapat menggunakan perintah mysql -u (nama user di mysql kalok melum ada bisa menggunakan root yang merupakan bawan dari mysql) -p (mysql -u root -p) jika menggunakan root saat muncul untuk memasukan password lasung tekan enter saya karena tidak ada password

Saat sudah masuk langkah awal adalah membuat database baru jika belum ada karena saya belum memiliki nya maka saya membuat database baru dengan nama nemo dan sekaligus membuat user baru pada mysql

Perintah untuk membuat user baru pada mysql :

- A. `CREATE USER 'nemo'@'localhost' IDENTIFIED BY 'glady123';`
Pentah ini di gunakan untuk membuat user baru di localhost dan dengan sandi glady123

B. GRANT ALL PRIVILEGES ON mydatabase.* TO 'nemo'@'localhost';

Digunakan untuk memberikan hak menggunakan mysql ke pada user yang baru

Di buat agar dapat menggunakan mysql

C. FLUSH PRIVILEGES;

Perintah ini di gunakan untuk menjalan atau menetapkan user nemo bisa bisa berjalan atau berfungsi atau berjalan

Langkah pembuatan table baru setelah sudah membuat user mysql baru di mysql

Setelah membuat user baru maka selanjutnya login ke user mysql yang baru di buat di saya sudah keluar dari user root dan berpindah ke user nemo kemudian setelah ada di user nemo saya membuat database baru dengan nama web1

Commad yang di gunakan dalam pembuat database

Create database web1;

Perintah tersebut di gunakan untuk membuat database baru setelah sudah di buat masuk ke tahapan pembuatan table untuk menyimpan data yang akan di gunakan user maupun admin untuk login dan juga menyimpan data diri dari user dan admin

Setelah tadi di buat serang di gunakan dengan menggunakan perintah use web1;

Untuk menggunakan database tersebut untuk membuat table, dalam pembuatan table menggunakan perintah ini :

A. CREATE TABLE data_user (

Perintah ini di gunakan untuk membuat table dengan nama data_user

B. id INT AUTO_INCREMENT PRIMARY KEY,

digunakan untuk mebuat table id secara otomatis dan tidak sama

C. nama VARCHAR(255) NOT NULL,

di gunkan untuk membuat table nama di pada databse database dan tidak boleh nol

D. username VARCHAR(255) UNIQUE NOT NULL,

di gunkan untuk membuat table username di pada databse database

E. email VARCHAR(255) UNIQUE NOT NULL,

di gunkan untuk membuat table email di pada databse database

F. password VARCHAR(255) NOT NULL,

di gunkan untuk membuat table pasword di pada databse database

G. alamat TEXT,

di gunkan untuk membuat table alamat di pada databse database

H. role ENUM('admin', 'user') NOT NULL DEFAULT 'user'

untuk membuat role

);

Setelah table sudah di buat kemudian tinggal memasukan user dan data yang di perlukan

INSERT INTO data_user (nama, username, email, password, alamat, role)

(insert into di gunakan untuk memasukan data ke table database yang di inginkan)

VALUES ('figo', 'figodb', 'figo@gmail.com', PASSWORD('figo123'), 'rumah', 'admin');

Tahap pembuatan web halaman login dasbord admin dan user

Sebelum membuat halama halaman login tahap pertama membuat php untuk konesi ke database sebagai berikut ;

```
db.php
1  <?php
2  // Informasi koneksi database
3  $servername = "localhost"; // Alamat server (bisa juga IP server atau nama domain)
4  $username = "nemo"; // Username database
5  $password = "GladYVv!!"; // Password database
6  $dbname = "web1"; // Nama database yang ingin diakses
7
8  // Membuat koneksi
9  $conn = new mysqli(hostname: $servername, username: $username, password: $password, database: $dbname);
10
11 // Cek koneksi
12 if ($conn->connect_error) {
13     die("Koneksi gagal: " . $conn->connect_error);
14 }
15
16 ?>
17
```

Pada gambar di atas merupakan contoh koneksi ke database menggunakan php penjelasan dari program php di atas pada bagian atas di mulai dari nomer 3 sampe dengan nomer 6 merupakan data yang di perlukan untuk dapat terhubung ke database kemudian nomer 9 berfungsi untuk membuat koneksi dengan database pada nomer 12 if \$conn --> connect_error di gunakan untuk menampung error dari koneksi ke mysql jika ada error akan di tampilkan

Jika sudah bisa terhubung ke database maka langkah selajutnya membuat halaman login pada halaman ini untuk login user perlu memasukan data berupa sandi dan username dengan system jika salah satu dari password atau username salah maka login tidak akan di anggap gagal namun jika kedua nya bernilai benar maka akan di arakan ke dasbaor sesuai dengan role yang di miliki jika user akan ke dasbord user dan jika memiliki role admin akan di arakan ke dasbord admin

```

<?php
session_start();
include 'db.php';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $username = ($_POST['username']);
    $password = ($_POST['password']);

    if (empty($username) || empty($password)) {
        $error = "Username dan password tidak boleh kosong!";
    } else {

        $sql = "SELECT * FROM data_user WHERE username = '$username' AND password = '$password'";
        $result = $conn->query(query: $sql);

        if ($result && $result->num_rows === 1) {
            $user = $result->fetch_assoc();

            if ($password === $user['password']) {
                $_SESSION['username'] = $user['username'];
                $_SESSION['role'] = $user['role'];
                $_SESSION['user_id'] = $user['id'];

                if ($user['role'] === 'admin') {
                    header(header: "Location: adminas.php");
                } elseif ($user['role'] === 'user') {
                    header(header: "Location: userdas.php");
                } else {
                    $error = "Role tidak valid!";
                }
                exit();
            } else {
                $error = "Username atau password salah!";
            }
        } else {
            $error = "Username atau password salah!";
        }
    }
}
?>

```

Ini adalah sintak php untuk membuat halaman login selain menggunakan html pada baris pertama terdapat perintah session_star(); yang di gunakan untuk memuali koneksi dengan database dan mencatat log dan catatan tentang user name password keterangan login agak dapat di teruskan ke halaman lain dari web app yang sedang di buat , setelah itu ada juga include yang di gunakan untuk menghubungkan sintak ini dengan database dan begitu juga sebaliknya request method di gunakan untuk menerima atau mendapatkan input dari user yang berupa password dan username yang akan di gunakan untuk login kemudian ada bagian \$post di gunakan untuk mengambil atau menerima inputan dari user

```

if (empty($username) || empty($password)) {
    $error = "Username dan password tidak boleh kosong!";
} else {

```

Bagian ini berfungsi untuk melakukan pengecekan jika ada salah satu dari user atau password tidak di isi maka akan menampilkan error yang di ambil dari html error bawaan

Kemudian jika kedua nya terisi maka akan di teruskan untuk apakah user dan password itu ada di database namun sebelum itu sintak di bawah ini akan mengambil data dari database terlebih dahulu

```
$sql = "SELECT * FROM data_user WHERE username = '$username' AND password = '$password'";
$result = $conn->query(query: $sql);
```

Dengan menggunakan perintah mysql di atas yang di gunakan untuk megambil data user dan password dari database ke mudian akan di bandingkan jika hanya ada satu data nya sama maka akan benar dan user dapat login hal ini di lakukan di sintak berikut

```
if ($result && $result->num_rows === 1) {
    $user = $result->fetch_assoc();
```

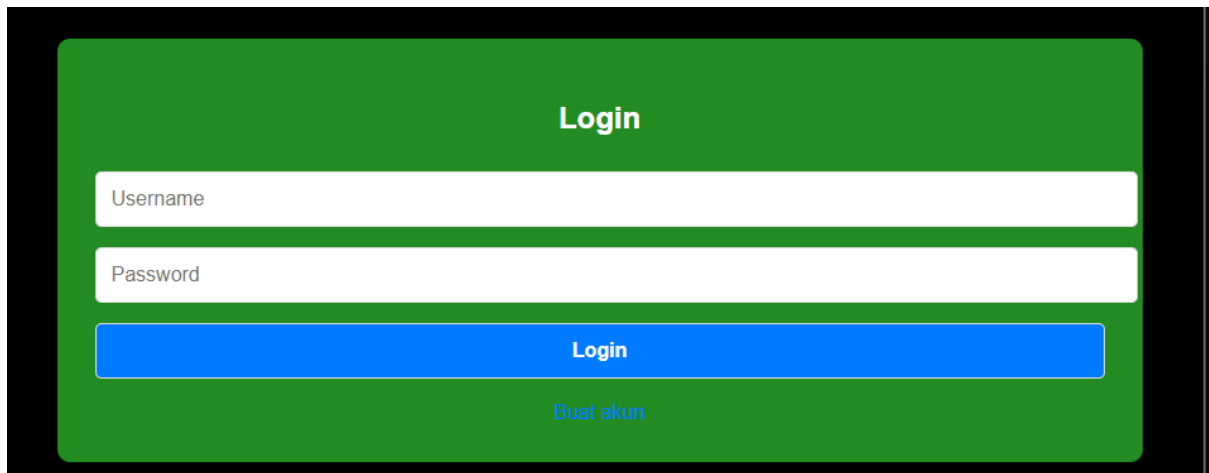
Yang berfungsi untuk melakukan pembadingan jika kedua parameter password dan user name benar maka kan bernilai tru atau satu kemudian akan juga akan di kalukan prosen pengecekan role dari user yang login apa kah dia adalah admin atau user apa bila dia user maka akan di lempar ke dasbord user begitu juga degan admin akan di lempar ke dasbord admin dengan sintak berikut

```
if ($password === $user['password']) {
    $_SESSION['username'] = $user['username'];
    $_SESSION['role'] = $user['role'];
    $_SESSION['user_id'] = $user['id'];

    if ($user['role'] === 'admin') {
        header(header: "Location: admin.php");
    } elseif ($user['role'] === 'user') {
        header(header: "Location: user.php");
    } else {
        $error = "Role tidak valid!";
    }
    exit();
} else {
    $error = "Username atau password salah!";
}
else {
    $error = "Username atau password salah!";
```

Pada sintak ini di gunakan memastikan password yang masuk sesuai dan juga user name kemudian akan di catat pada program php dengan session agar saat berpindah halaman user tidak perlu lagi melakukan verifikasi data lagi karena data login nya sudah di simpan

Kemudian pada sintak ini juga di lakukan pengecekan apa kah yang melakukan login adalah admin atau user yakni pada program user role dengan jika dia memiliki role admin maka akan di arahkan ke dasadmin.php jika bukan maka akan di arahkan ke userdas.php namun



Login

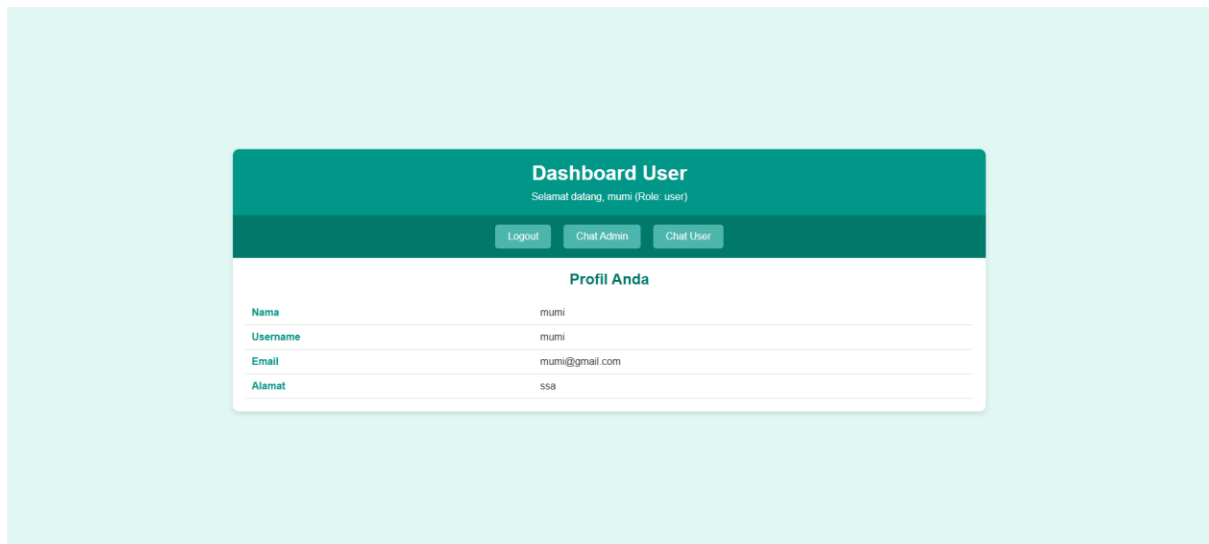
Username

Password

Login

[Buat akun](#)

Ini adalah tampilan dari halaman login nya jika user belum memiliki akun maka akan di arakan ke dasbord untuk membuat akun jika sudah maka akan di lakukan pengarahannya ke dasbord sesuai dengan role yang di miliki misal dia di role user maka akan di arakan ke dasbord user



Dashboard User
Selamat datang, mumi (Role: user)

[Logout](#) [Chat Admin](#) [Chat User](#)

Profil Anda

Nama	mumi
Username	mumi
Email	mumi@gmail.com
Alamat	ssa

Ini adalah sintak dari halaman dasbord user

Bagaimana php dari halaman user

```

<?php
session_start();
require 'db.php';

// Tidak ada pengamanan pada sesi
if (!isset($_SESSION['username'])) {
    header(header: "Location: login.php");
    exit;
}

$username = $_SESSION['username'];

// Query tanpa prepared statement (rentan SQL Injection)
$sql = "SELECT username, role, nama, alamat, email, id FROM data_user WHERE username = '$username'";
$result = $conn->query(query: $sql);

if ($result->num_rows > 0) {
    $user = $result->fetch_assoc();
    $user_name = $user['username']; // Tidak ada escaping (rentan XSS)
    $user_role = $user['role'];
    $user_nama = $user['nama'];
    $user_alamat = $user['alamat'];
    $user_email = $user['email'];
    $user_id = $user['id'];
} else {
    header(header: "Location: login.php");
    exit;
}

// Query untuk mendapatkan ID admin (rentan SQL Injection)
$sql_admin = "SELECT id FROM data_user WHERE role = 'admin' LIMIT 1";
$result_admin = $conn->query(query: $sql_admin);

if ($result_admin->num_rows > 0) {
    $admin = $result_admin->fetch_assoc();
    $admin_id = $admin['id']; // Tidak ada validasi tipe data
} else {
    $admin_id = null; // Admin tidak ditemukan
}
?>

```

Session_start(); di sini untuk memulai seson pada halaman ini dan mengalihkan session dari halaman login ke halaman user

```

if (!isset($_SESSION['username'])) {
    header(header: "Location: login.php");
    exit;
}

```

Sintak di atas di gunakan untuk megalikan jika user terdetesi belum login dengan melakukan pengecekan pada data di session yang ada pada halama login jika sudah maka boleh masuk namun jika belum maka akan di kembalikan ke halaman login lalu exit di gunakan untuk menghentikan sesi pada halaman login

Kemudian pada `$username = $_SESSION['username'];` di gunakan untuk melakukan pegecekan username yang login ke dasbord user jika sudah maka akan di ambilkan data dari database berdasarkan user name yang sama yang ada pada database

```

$sql = "SELECT username, role, nama, alamat, email, id FROM data_user WHERE username = '$username'";
$result = $conn->query(query: $sql);

```

Dengan sintak mysql di atas untuk mengambil data pada database ke mudian akan di simpan pada \$result kemudian dan menyimpan data yang sudah di ambil dari datase akan di simpan di local yaitu pada

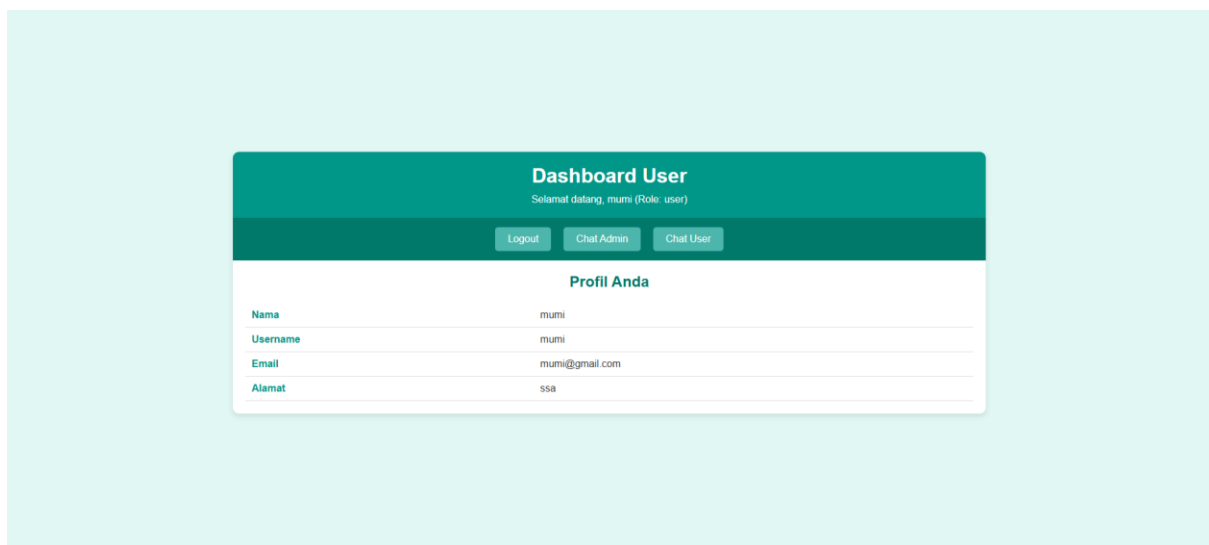
```
$user = $result->fetch_assoc();  
$user_name = $user['username'];  
$user_role = $user['role'];  
$user_nama = $user['nama'];  
$user_alamat = $user['alamat'];  
$user_email = $user['email'];  
$user_id = $user['id'];
```

Kemudian jika data tidak di temukan maka akan di alihkan kembali ke halaman login

```
} else {  
    header(header: "Location: login.php");  
    exit;  
}
```

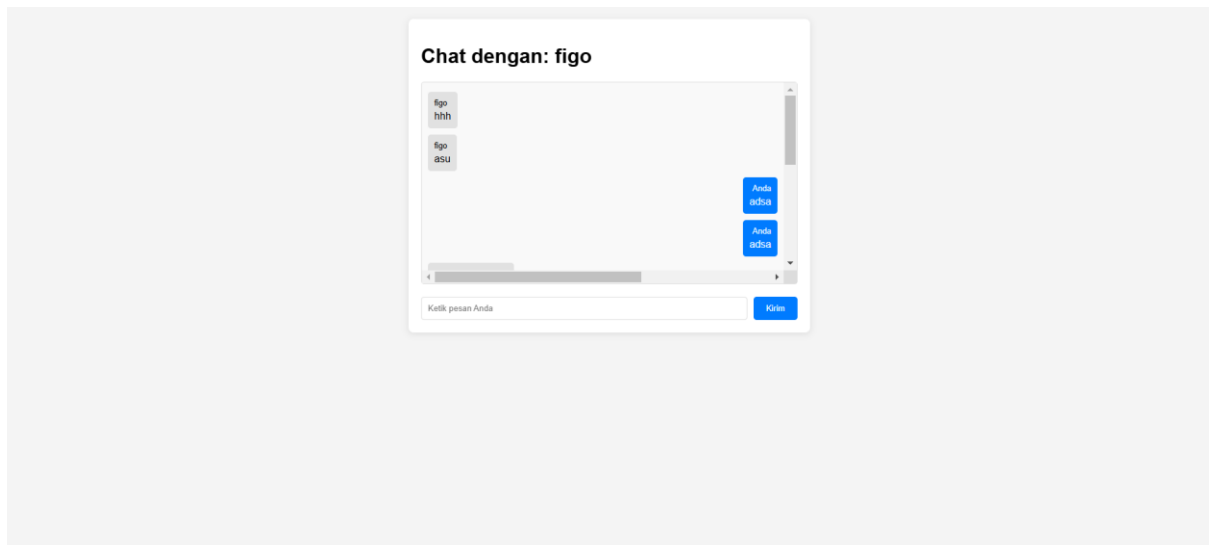
Di atas merupakan sintak untuk megarakan kembali ke dasbord user apa bila data yang ada pada database tidak di temukan

Pada dasbord ini memiliki memiliki fitur chat dengan antar user dan chat pribadi denga admin



Ini adalah tampilan dari dasbord user yang memiliki 2 fitur chat

Ini adalah tampilan dari chat dengan admin



Langkah pertama dalam pembuatan halaman untuk chat ini adalah membuat database baru yang berguna untuk menyimpan data chat dari user atau admin dan menyediakan nomer yang di gunakan untuk terhubung dengan user lain atau admin

CREATE TABLE messages (

(perintah ini di gunakan untuk membuat table baru dengan nama messages)

id INT AUTO_INCREMENT PRIMARY KEY,

(perintah ini untuk membuat kolom untuk di yang berfungsi sebagai primary key agar setiap data yang di simpan memiliki kunci yang di gunakan untuk membedakan yang di buat secara otomatis dengan perintah auto)

sender_id INT NOT NULL,

(sender_id ini di gunakan untuk menyimpan id dari pengirim pesan dengan tambah berbentuk interjer dan tidak boleh kosong)

receiver_id INT NOT NULL,

(pada sintak di mysql di atas di gunkan untuk menyimpan id dari penerima pesan int merupakan tipe datanya kemudian not null itu menandakan bahwa kolom ini tidak boleh kosong)

message TEXT,

(membuat kolom dengan nama message dan dengan tipe text)

timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

(perintah ini di gunakan untuk membuat kolom untuk menyimpan data dari waktu pesan itu di buat

created_at DATETIME DEFAULT CURRENT_TIMESTAMP

pada bagian ini di gunakan untuk membuat tanggal dan waktu pesan di buat otomatis di isikan)

);

Sintak hph untuk halaman chat admin

```
<?php
session_start();
require 'db.php';

// Tidak ada validasi sesi yang memadai
if (!isset($_SESSION['user_id'])) {
    header(header: "Location: login.php");
    exit;
}
```

Pada baris pertama terdapat session_start(); yang berguna untuk meneruskan dari seson halaman sebelumnya ke mudian pada baris berikut nya ada require untuk terhubung dengan database pada if ini di gunakan untuk melakukan pengecekan pada data user name apa kah terdaftar dan sudah login jika sudah maka bisa lanjut ke tahap selanjutnya namun jika belum maka akan di lempar ke bali ke dasbord untuk login

```
$sender_id = $_SESSION['user_id'];

// Query langsung tanpa prepared statement (rentan SQL Injection)
$sql_admin = "SELECT id FROM data_user WHERE role = 'admin' LIMIT 1";
$result_admin = $conn->query(query: $sql_admin);

if ($result_admin && $result_admin->num_rows > 0) {
    $admin = $result_admin->fetch_assoc();
    $receiver_id = $admin['id'];
} else {
    die("Admin tidak ditemukan.");
}
```

Pada bagian awal dari sintak ini di gunakan untuk mengambil user_id dari session dan meyimpan nya pada \$sender_id yang berguna sebagai id pengirim pesan seperti nomer telfon untuk tanda pengenalan , pada baris selajutnya terdapat \$sql_admin = "SELECT id FROM data_user WHERE role = 'admin' LIMIT 1"; di gunakan untuk mengambil id user dengan role admin namun di batasi hanya satu user saja yang di panggil kemudian sintak di bawah nya digunakan untuk menjalankan perintah snitak sql dengan menggunakan koneksi ke database dengan \$conn kemudian output dari database akan di simpan pada \$result_admin

Kemudian pada if akan memeriksa apa pada result_admin berhasil mengambil datanya kemudian pada sintak berikutnya output yang sudah di hasilkan kan di simpan pada \$receiver_id sebagai id admin pada baris berikutnya jika admin tidak di temukan maka sintak akan berhenti dan mengeluarkan pesan error

```
$query_receiver = "SELECT nama FROM data_user WHERE id = '$receiver_id'";
$result_receiver = $conn->query(query: $query_receiver);
$receiver_name = 'Unknown User';
if ($result_receiver && $result_receiver->num_rows > 0) {
    $row_receiver = $result_receiver->fetch_assoc();
    // Tidak ada sanitasi output (rentan XSS)
    $receiver_name = $row_receiver['nama'];
}
```

Pada sintak ini pada bagian awal akan mencari username admin dari database dengan panduan pencarian menggunakan receiver_id Dari admin kemudian pada baris berikutnya di gunakan untuk menjalankan perintah dari sintak mysql di atas kemudian pada baris ke tiga di gunakan untuk melakukan inisialisasi pada variable \$receiver_name dengan nilai default

Kemudian pada if ini di gunakan untuk melakukan pemeriksaan apa kah query berhasil di lakukan dan melakukan pengembalian data pada baris selanjutnya di gunakan untuk mengambil hasil dari query dan di simpan pada \$row_receiver kemudian pada baris selanjutnya akan mengambil nama admin dari array \$row_receiver dan meyimpanya pada \$receiver_name

```
// Tidak ada validasi input (rentan XSS dan SQL Injection)
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $message = $_POST['message'];

    // Query langsung tanpa sanitasi (rentan SQL Injection)
    $query = "INSERT INTO messages (sender_id, receiver_id, message) VALUES ('$sender_id', '$receiver_id', '$message')";
    if (!$conn->query(query: $query)) {
        die("Gagal mengirim pesan: " . $conn->error);
    }
}
```

Pada baris pertama di gunakan untuk mengecek apa request yang datang menggunakan metod post ke mudian pada baris kedua digunakan untuk mengabil pesan dari post lalu di simpan dalam parameter \$message kemudian pada baris ke tiga di gunakan untuk meyimpan pesan yang di dapat dari from ke dalam database

```
$query = "
    SELECT m.*, u.nama as sender_name
    FROM messages m
    LEFT JOIN data_user u ON m.sender_id = u.id
    WHERE (m.sender_id = '$sender_id' AND m.receiver_id = '$receiver_id')
    OR (m.sender_id = '$receiver_id' AND m.receiver_id = '$sender_id')
    ORDER BY m.created_at ASC";
$result = $conn->query(query: $query);

if (!$result) {
    die("Gagal mengambil pesan: " . $conn->error);
}
?>
```

Pada sintak ini mengambil semua pesan yang di buat oleh user dangan user lain yang dia aja berkomunikasi kemudian pesan di urutkan dengan waktu dangan created_at