1. Penjelasan Coding Hallologin.php
2. **<title>Login Page</title>**: Menetapkan judul halaman yang akan ditampilkan di tab browser.
3. **</head>**: Menutup bagian kepala dokumen HTML.
4. **<body>**: Bagian utama dokumen HTML yang berisi konten yang akan ditampilkan kepada pengguna.
5. **<h2>Login</h2>**: Menampilkan heading level 2 yang berisi teks "Login".
6. **<form method="post" action="algologin.php">**: Membuat sebuah formulir dengan metode HTTP POST yang akan mengirim data ke **algologin.php** saat formulir disubmit.
7. **<label for="username">Username:</label>**: Menampilkan label "Username:" untuk input username.
8. **<input type="text" id="username" name="username" required>**: Membuat input teks untuk username dengan atribut **required** yang menandakan input ini wajib diisi.
9. **<br><br>**: Menggunakan tag **<br>** untuk membuat baris baru (line break) di antara elemen-elemen form.
10. **<label for="password">Password:</label>**: Menampilkan label "Password:" untuk input password.
11. **<input type="password" id="password" name="password" required>**: Membuat input password yang tersembunyi dan juga wajib diisi.
12. **<button type="submit" name="login">Login</button>**: Membuat tombol submit untuk mengirimkan data formulir. Ketika tombol ini ditekan, data formulir akan dikirimkan ke **algologin.php**.

2. Penjelasan Coding algologin.php

1. **<?php ... ?>**: Ini adalah tag PHP yang menandakan awal dan akhir dari blok kode PHP di dalam dokumen HTML. Semua kode PHP harus berada di antara tag ini.
2. **session\_start();**: Fungsi **session\_start()** digunakan untuk memulai atau melanjutkan sesi yang ada. Dalam konteks login, ini penting karena Anda kemungkinan akan menyimpan informasi sesi seperti nama pengguna yang login.
3. **if(isset($\_POST['login'])) { ... }**: Ini adalah struktur kontrol **if** yang digunakan untuk mengecek apakah ada data yang dikirimkan melalui metode POST dari formulir login. **$\_POST['login']** adalah nama dari elemen tombol submit pada formulir login.
4. Komentar **// Ambil data dari form**: Ini adalah komentar yang memberikan penjelasan tentang apa yang dilakukan oleh blok kode berikutnya.
5. **$username = $\_POST['username'];** dan **$password = $\_POST['password'];**: Dua baris ini mengambil nilai yang dimasukkan pengguna untuk username dan password dari formulir login yang disubmit.
6. Blok **// Verifikasi username dan password ... } else { echo "wait a minute who are you"; }**: Di dalam blok **if**, username dan password yang dimasukkan oleh pengguna diverifikasi. Dalam kasus ini, username dan password yang benar adalah "jayyidganteng" dan "pakebanget". Jika verifikasi berhasil, maka sesi akan dimulai dan pengguna akan diarahkan ke halaman dashboard **codegambang.html** dengan menggunakan fungsi **header("Location: codegambang.html");**. Jika verifikasi gagal, maka pesan "wait a minute who are you" akan ditampilkan.
7. **exit();**: Ini adalah fungsi PHP yang digunakan untuk menghentikan eksekusi skrip PHP setelah pengalihan header dilakukan. Hal ini penting karena setelah melakukan pengalihan header, Anda tidak ingin skrip PHP terus dieksekusi.

3. Penjelasan Code algologout

1. **<?php ... ?>**: Ini adalah tag PHP yang menandakan awal dan akhir dari blok kode PHP di dalam dokumen.
2. **session\_start();**: Fungsi ini digunakan untuk memulai sesi pada halaman PHP. Sesuai dengan namanya, **session\_start()** digunakan untuk memulai atau melanjutkan sesi yang sudah ada.
3. **session\_unset();**: Fungsi ini digunakan untuk menghapus semua variabel sesi yang disimpan. Ini membantu untuk membersihkan data sesi yang tidak diperlukan lagi.
4. **session\_destroy();**: Fungsi ini digunakan untuk menghapus seluruh sesi yang sedang berjalan. Hal ini termasuk menghapus data sesi dari server dan menghentikan sesi pengguna saat ini.
5. **header("Location: hallogin.php");**: Ini adalah header HTTP yang digunakan untuk mengalihkan pengguna ke halaman **hallogin.php** setelah proses logout selesai. Dengan menggunakan **header("Location: ...")**, Anda mengarahkan browser pengguna ke halaman baru.
6. **exit();**: Fungsi **exit()** digunakan untuk menghentikan eksekusi skrip PHP setelah header HTTP dikirimkan. Hal ini penting agar tidak ada kode tambahan yang dieksekusi setelah melakukan pengalihan header.

Penjelasan dari codegambang.html

Mari kita jelaskan setiap baris dari kode formulir input nilai mahasiswa tersebut:

1. **<h2>Form Input Nilai Mahasiswa</h2>**: Ini adalah elemen HTML yang digunakan untuk menampilkan judul "Form Input Nilai Mahasiswa" dalam ukuran heading level 2 (h2).
2. **<form id="inputForm">**: Ini adalah elemen **<form>** yang digunakan untuk mengelompokkan elemen-elemen input yang terkait dengan pengiriman data. Atribut **id="inputForm"** memberikan ID "inputForm" ke formulir ini, yang dapat digunakan untuk mengidentifikasi formulir secara unik dalam halaman.
3. **<label for="name">Nama:</label>**: Ini adalah elemen **<label>** yang terkait dengan input "name" dengan atribut **for="name"**. Label ini digunakan untuk memberikan keterangan atau petunjuk tentang apa yang diharapkan dari input tersebut (dalam hal ini, nama mahasiswa).
4. **<input type="text" id="name" name="name" required>**: Ini adalah input teks (**<input type="text">**) yang digunakan untuk memasukkan nama mahasiswa. Atribut **id="name"** memberikan ID "name" ke input ini, dan atribut **name="name"** memberikan nama "name" ke input ini. Atribut **required** menandakan bahwa input ini harus diisi sebelum formulir dapat disubmit.
5. **<label for="nim">NIM:</label>** dan **<input type="text" id="nim" name="nim" required>**: Ini adalah elemen label dan input untuk Nomor Induk Mahasiswa (NIM), dengan konfigurasi yang serupa dengan input nama.
6. **<label for="tugas">Total Nilai Tugas:</label>** dan **<input type="number" id="tugas" name="tugas" min="0" required>**: Ini adalah elemen label dan input untuk memasukkan total nilai tugas mahasiswa. Tipe input adalah "number", yang berarti hanya nilai numerik yang diperbolehkan. Atribut **min="0"** menunjukkan bahwa nilai minimum yang diperbolehkan adalah 0.
7. **<label for="pbl">Total Nilai PBL:</label>** dan **<input type="number" id="pbl" name="pbl" min="0" required>**: Ini adalah elemen label dan input untuk memasukkan total nilai Project Based Learning (PBL) mahasiswa, dengan konfigurasi yang serupa dengan input total nilai tugas.
8. **<label for="uts">Nilai UTS:</label>** dan **<input type="number" id="uts" name="uts" min="0" required>**: Ini adalah elemen label dan input untuk memasukkan nilai Ujian Tengah Semester (UTS) mahasiswa, dengan konfigurasi yang serupa dengan input total nilai tugas.
9. **<label for="uas">Nilai UAS:</label>** dan **<input type="number" id="uas" name="uas" min="0" required>**: Ini adalah elemen label dan input untuk memasukkan nilai Ujian Akhir Semester (UAS) mahasiswa, dengan konfigurasi yang serupa dengan input total nilai tugas.
10. **<input type="button" value="Submit" onclick="submitForm()">**: Ini adalah tombol submit (**<input type="button">**) yang digunakan untuk mengirimkan formulir. Ketika tombol ini ditekan, fungsi **submitForm()** akan dipanggil untuk memproses data yang dimasukkan pengguna ke dalam formulir.
11. **<h2>Output Tabel Nilai Mahasiswa</h2>**: Ini adalah judul level 2 (h2) yang menunjukkan bahwa di bawahnya akan ada tabel yang menampilkan data nilai mahasiswa.
12. **<table id="outputTable">**: Ini adalah elemen tabel HTML dengan ID "outputTable". ID ini digunakan untuk mengidentifikasi tabel tersebut secara unik dalam halaman.
13. **<tr>**: Ini adalah tag untuk membuat baris (row) di dalam tabel.
14. **<th>**: Ini adalah tag untuk membuat sel header (kolom header) di dalam tabel. Di dalam tag ini, terdapat teks yang menunjukkan nama kolom atau atribut yang akan ditampilkan di tabel.
15. Setelah tag **<th>** terdapat teks seperti "Nama", "NIM", "Nilai Tugas", dll., yang mewakili nama kolom atau atribut yang akan ditampilkan di tabel.
16. Setelah semua kolom header selesai (ditutup dengan tag **</tr>**), tabel akan siap untuk menampilkan data nilai mahasiswa sesuai dengan format yang telah ditentukan dalam kolom-kolom tersebut.
17. **var form = document.getElementById("inputForm");**: Baris ini mendapatkan referensi ke elemen form dengan id "inputForm" dari halaman HTML. Ini dilakukan dengan menggunakan **document.getElementById()** yang merupakan metode untuk mendapatkan elemen berdasarkan id-nya.
18. **var name = form.elements["name"].value;**: Di sini, kita mendapatkan nilai dari input dengan nama "name" di dalam formulir. **form.elements["name"]** mengambil elemen input dengan atribut **name="name"**, dan **.value** digunakan untuk mendapatkan nilai yang dimasukkan oleh pengguna ke dalam input tersebut.
19. **var nim = form.elements["nim"].value;**: Baris ini melakukan hal yang sama dengan baris sebelumnya, tetapi untuk input dengan nama "nim".
20. **var tugas = parseFloat(form.elements["tugas"].value);**: Disini, kita mendapatkan nilai dari input "tugas" dan mengkonversi nilainya menjadi tipe data float dengan menggunakan **parseFloat()**. Hal ini berguna jika nilai yang dimasukkan adalah dalam format angka desimal, sehingga kita bisa melakukan perhitungan matematika dengan tepat.
21. **var pbl = parseFloat(form.elements["pbl"].value);**: Langkah ini serupa dengan baris sebelumnya, namun untuk input "pbl".
22. **var uts = parseFloat(form.elements["uts"].value);**: Baris ini melakukan hal yang sama dengan baris sebelumnya, tetapi untuk input "uts".
23. **var uas = parseFloat(form.elements["uas"].value);**: Disini, kita mendapatkan nilai dari input "uas" dan mengkonversi nilainya menjadi tipe data float.
24. **var totalNilai = (tugas \* 0.05) + (pbl \* 0.75) + (uts \* 0.10) + (uas \* 0.10);**: Baris ini menghitung total nilai mahasiswa dengan mengalikan nilai dari setiap jenis tugas (tugas, PBL, UTS, dan UAS) dengan bobotnya masing-masing. Misalnya, nilai tugas dikalikan dengan 0.05 (5%), nilai PBL dengan 0.75 (75%), nilai UTS dengan 0.10 (10%), dan nilai UAS dengan 0.10 (10%), kemudian hasilnya dijumlahkan.
25. **var kategoriNilai;**: Ini adalah deklarasi variabel **kategoriNilai** yang akan digunakan untuk menyimpan kategori nilai mahasiswa.
26. **if (totalNilai >= 80) { kategoriNilai = 'A'; }**: Ini adalah kondisi pertama dalam penentuan kategori nilai. Jika totalNilai lebih besar atau sama dengan 80, maka kategoriNilai akan diisi dengan 'A'.
27. **else if (totalNilai >= 75 && totalNilai < 80) { kategoriNilai = 'B+'; }**: Ini adalah kondisi kedua. Jika totalNilai lebih besar atau sama dengan 75 dan kurang dari 80, maka kategoriNilai akan diisi dengan 'B+'.
28. **else if (totalNilai >= 70 && totalNilai < 75) { kategoriNilai = 'B'; }**: Ini adalah kondisi ketiga. Jika totalNilai lebih besar atau sama dengan 70 dan kurang dari 75, maka kategoriNilai akan diisi
29. Kondisi selanjutnya (**else if**) secara berurutan mengecek rentang nilai totalNilai dan mengisi variabel kategoriNilai sesuai dengan skala kategori yang telah ditentukan (C+, C, D+, E).
30. **else { kategoriNilai = 'E'; }**: Jika nilai totalNilai tidak masuk ke dalam kondisi-kondisi sebelumnya, maka kategoriNilai akan diisi dengan 'E', yang menunjukkan kategori nilai terendah.
31. **ar table = document.getElementById("outputTable");**:
    1. Kode ini mengambil referensi ke elemen tabel di halaman HTML dengan ID "outputTable".
    2. Variabel **table** kemudian menyimpan referensi tersebut untuk digunakan selanjutnya dalam skrip.
32. **var newRow = table.insertRow(-1);**:
    1. Baris ini menambahkan baris baru ke dalam tabel dengan menggunakan metode **insertRow()**.
    2. **-1** yang dilewatkan ke dalam metode **insertRow()** mengindikasikan bahwa baris baru akan ditambahkan di akhir tabel.
33. **var cell1 = newRow.insertCell(0);**, **var cell2 = newRow.insertCell(1);**, dst.:
    1. Setelah baris baru ditambahkan, kode ini menambahkan sel-sel baru ke dalam baris tersebut menggunakan metode **insertCell()**.
    2. **insertCell(0)** menambahkan sel pertama (index 0) ke dalam baris, **insertCell(1)** menambahkan sel kedua (index 1), dan seterusnya.
34. **cell1.innerHTML = name;**, **cell2.innerHTML = nim;**, dst.:
    1. Setelah sel-sel baru ditambahkan, kode ini mengisi konten dari masing-masing sel dengan nilai yang sesuai.
    2. **name**, **nim**, **tugas**, **pbl**, **uts**, **uas** adalah variabel yang berisi nilai dari input yang dimasukkan oleh pengguna melalui formulir. Nilai-nilai ini diisikan ke dalam sel-sel yang telah ditambahkan sebelumnya.
    3. **totalNilai.toFixed(2)** digunakan untuk mengonversi nilai **totalNilai** menjadi string dengan dua angka di belakang koma (desimal). Hasilnya kemudian diisikan ke dalam sel **cell7**.
    4. **kategoriNilai** adalah variabel yang berisi kategori nilai yang telah ditentukan berdasarkan nilai total mahasiswa. Nilai ini diisikan ke dalam sel **cell8**.
35. **form.reset();**
    1. Ini adalah baris kode JavaScript yang berfungsi untuk mengatur ulang (reset) nilai-nilai yang telah dimasukkan oleh pengguna ke dalam formulir.
    2. Ketika formulir telah disubmit dan nilai-nilai telah diproses, menggunakan **form.reset();** akan menghapus semua nilai yang ada di formulir, sehingga memungkinkan pengguna untuk mengisi formulir kembali dengan data yang baru.
36. **Elemen <a href="algologout.php">Kalo Capek Udahan Aja</a>**
    1. Ini adalah elemen HTML yang berupa hyperlink (tautan).
    2. **href="algologout.php"** mengarahkan tautan ke halaman **algologout.php** saat tautan tersebut di-klik.
    3. Teks "Kalo Capek Udahan Aja" yang terdapat di antara tag **<a>** dan **</a>** merupakan teks tautan yang ditampilkan kepada pengguna sebagai opsi untuk keluar dari halaman atau aplikasi.
    4. Saat pengguna mengklik tautan tersebut, mereka akan diarahkan ke halaman **algologout.php**, yang dapat digunakan untuk melakukan tindakan logout atau keluar dari sesi pengguna yang sedang aktif.