# BAB IV IMPLMENTASI DAN PENGUJIAN

## 4.1 Spesifikasi

Pada tahap implementasi sistem, diperlukan perangkat pendukung yang memadai agar proses pengembangan dan pengujian dapat berjalan dengan baik. Spesifikasi ini mencakup kebutuhan perangkat lunak dan perangkat keras yang digunakan selama proses pembuatan, pengembangan, dan pengujian sistem pendukung keputusan pemilihan guru terbaik berbasis web dengan metode ARAS. Penjabaran spesifikasi bertujuan untuk memberikan gambaran mengenai lingkungan pengembangan sistem yang digunakan oleh penulis.

### 4.1.1 Spesifikasi Perangkat Keras

Dalam pengembangan sistem, penulis menggunakan perangkat keras untuk mendukung proses pembuatan sistem pendukung keputusan dalam menentukan guru terbaik menggunakan metode ARAS. Perangkat keras ini digunakan selama proses perancangan, implementasi, serta pengujian sistem. Adapun spesifikasi perangkat keras yang digunakan oleh penulis adalah sebagai berikut:

|  |  |  |
| --- | --- | --- |
| No | Perangkat Keras | Spesifikasi |
| 1 | Processor | AMD Ryzen 5 3500U |
| 2 | RAM | 8 GB |
| 3 | SSD | 256 GB |
| 4 | VGA | AMD Radeon |

### 4.1.2 Spesifikasi Perangkat Lunak

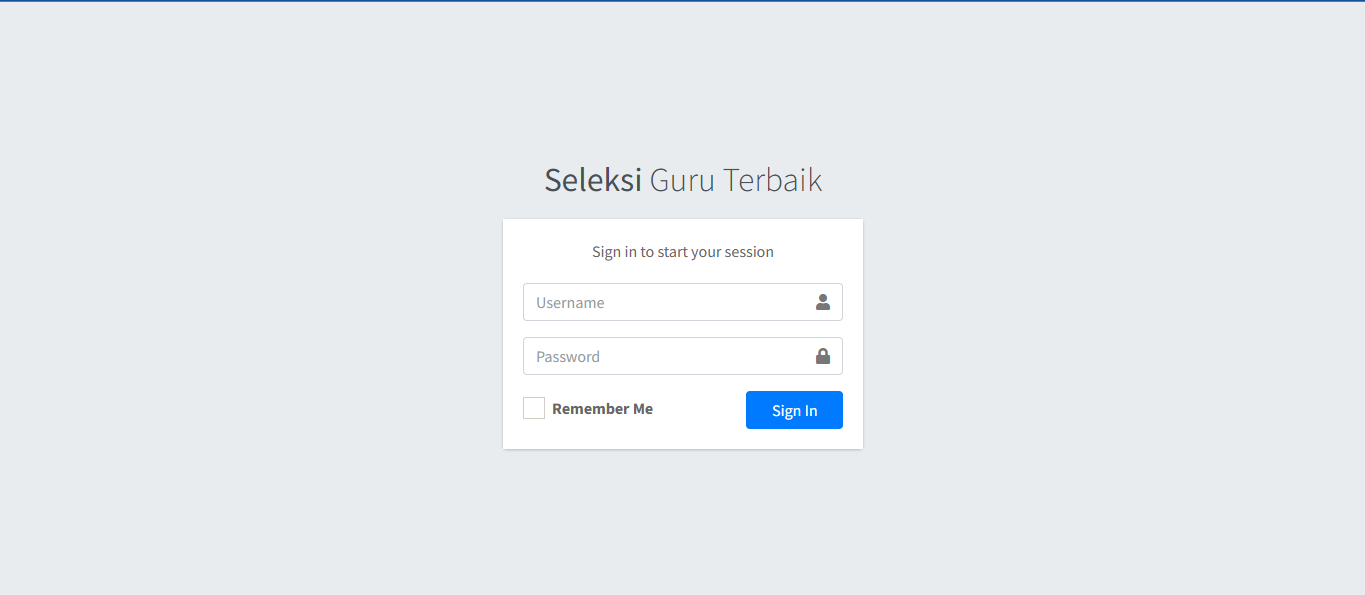
Selain perangkat keras, perangkat lunak juga sangat berperan penting dalam pengembangan sistem. Perangkat lunak digunakan sebagai alat bantu untuk melakukan perancangan antarmuka, pembuatan program, pengelolaan basis data, hingga proses perhitungan metode ARAS. Adapun spesifikasi perangkat lunak yang digunakan dalam pembangunan sistem pendukung keputusan pemilihan guru terbaik ini adalah sebagai berikut:

|  |  |  |
| --- | --- | --- |
| No | Perangkat Lunak | Keterangan |
| 1 | Windows 11 | Sistem operasi utama yang digunakan untuk menjalankan seluruh pengembangan |
| 2 | Visual Studio Code (VSCode) | Code editor untuk menulis dan mengelola file program |
| 3 | XAMPP | Paket server lokal yang terdiri dari Apache, MySQL, dan PHP |
| 4 | HTML | Bahasa markup untuk membuat struktur halaman web |
| 5 | CSS | Bahasa gaya untuk mendesain tampilan antarmuka web |
| 6 | PHP | Bahasa pemrograman backend untuk logika sistem dan koneksi database |
| 7 | Bootstrap | Framework CSS untuk mempercepat desain UI responsif dan konsisten |
| 8 | Draw.io | Aplikasi berbasis web untuk membuat diagram seperti UML dan ERD |

## 4.2 Implementasi Antarmuka

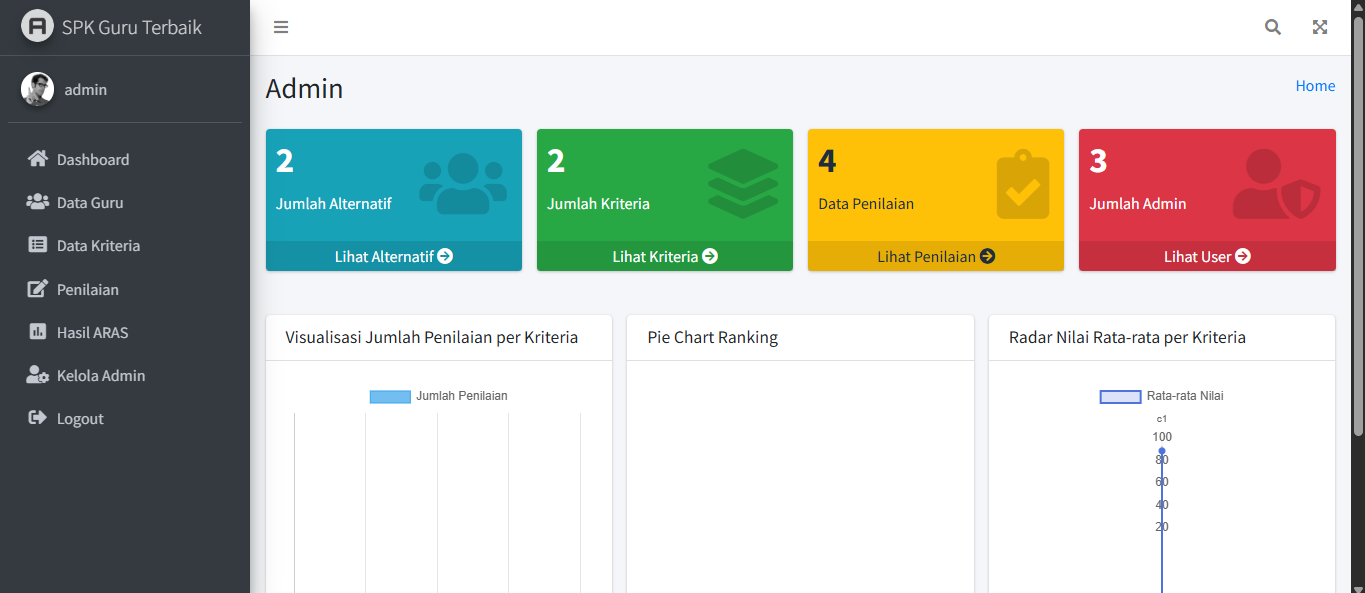
Implementasi antarmuka pada sistem ini meliputi tampilan halaman login, halaman dashboard, halaman data kriteria, halaman data alternatif, halaman penilaian alternatif, halaman hasil perhitungan menggunakan metode ARAS, serta halaman kelola admin. Penerapan antarmuka ini bertujuan untuk merealisasikan rancangan sistem yang telah dibuat sebelumnya dalam bentuk visual, sehingga memudahkan pengguna dalam mengakses, mengelola data, serta melakukan perhitungan untuk pemilihan guru terbaik. Berikut merupakan hasil tampilan antarmuka akhir dari sistem pendukung keputusan yang telah diimplementasikan:

### 4.2.1 Halaman Login



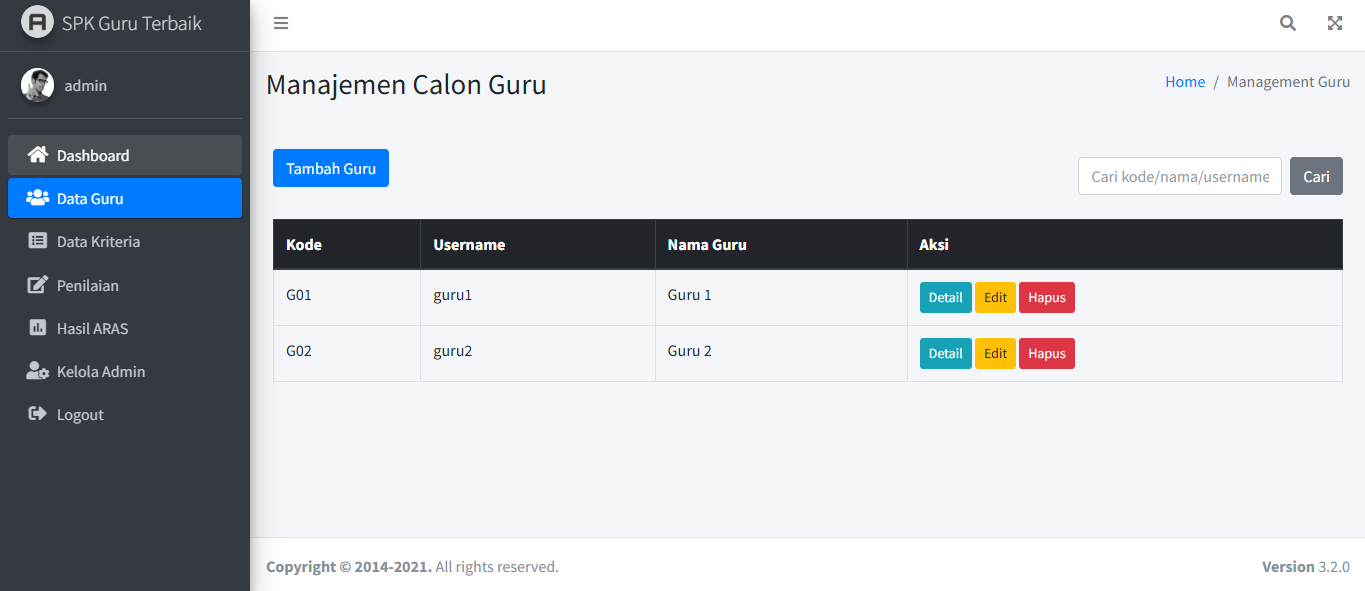
Halaman login merupakan pintu masuk utama bagi admin untuk mengakses sistem pendukung keputusan pemilihan guru terbaik. Pada halaman ini, admin diminta untuk memasukkan username dan password yang telah terdaftar sebelumnya. Tujuan dari halaman login ini adalah untuk menjaga keamanan dan membatasi akses hanya kepada pengguna yang memiliki wewenang. Jika login berhasil, maka admin akan diarahkan menuju halaman dashboard utama. Jika data login salah, sistem akan memberikan notifikasi kesalahan.

### 4.2.2 Halaman Dashboard



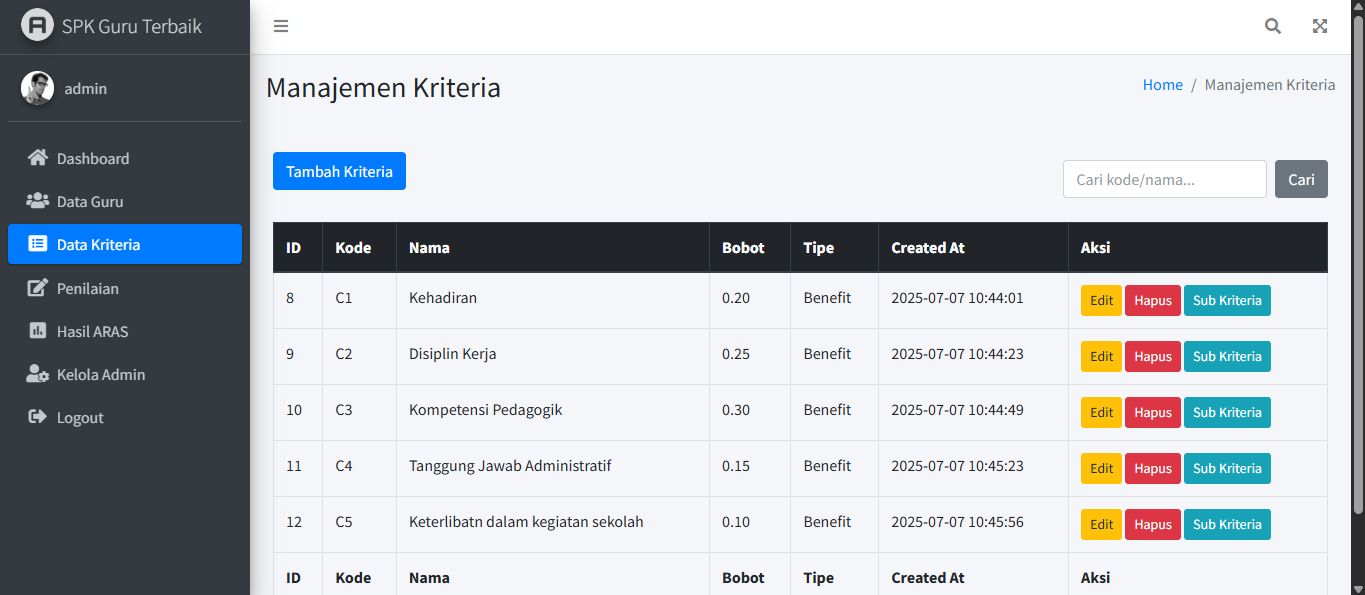
Halaman dashboard merupakan tampilan utama yang ditampilkan setelah admin berhasil login ke dalam sistem. Halaman ini berfungsi sebagai pusat navigasi untuk mengakses seluruh fitur yang tersedia dalam sistem. Pada halaman ini ditampilkan ringkasan informasi penting, seperti jumlah data guru (alternatif), jumlah kriteria, dan akses cepat menuju menu utama lainnya seperti penilaian dan hasil ARAS. Desain dashboard dibuat sederhana dan intuitif agar memudahkan admin dalam mengelola dan memantau data secara keseluruhan.

### 4.2.3 Halaman Data Alternatif



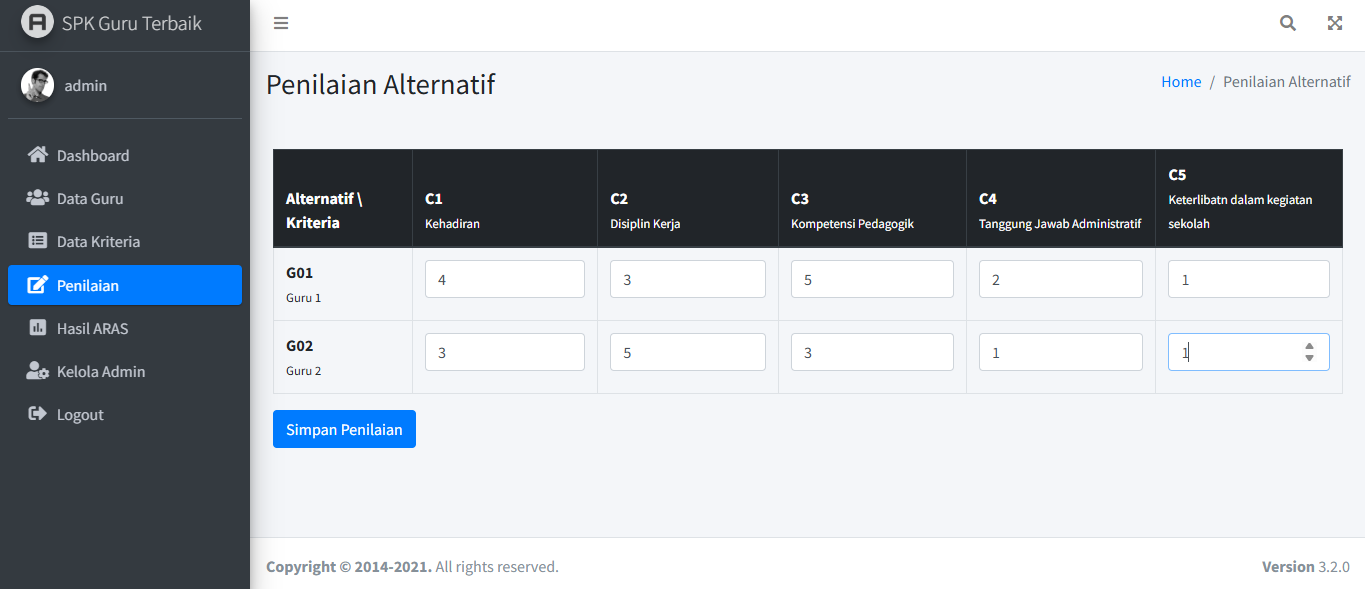
Halaman data alternatif digunakan untuk mengelola daftar guru yang akan menjadi kandidat dalam pemilihan guru terbaik. Dalam halaman ini, admin dapat melakukan input data guru seperti kode dan nama guru, serta mengedit atau menghapus data yang sudah ada. Data alternatif ini akan digunakan dalam proses penilaian dan perhitungan menggunakan metode ARAS. Tampilan halaman dibuat sederhana dan mudah dipahami agar memudahkan proses pengelolaan data alternatif oleh admin.

### 4.2.4 Halaman Data Kriteria



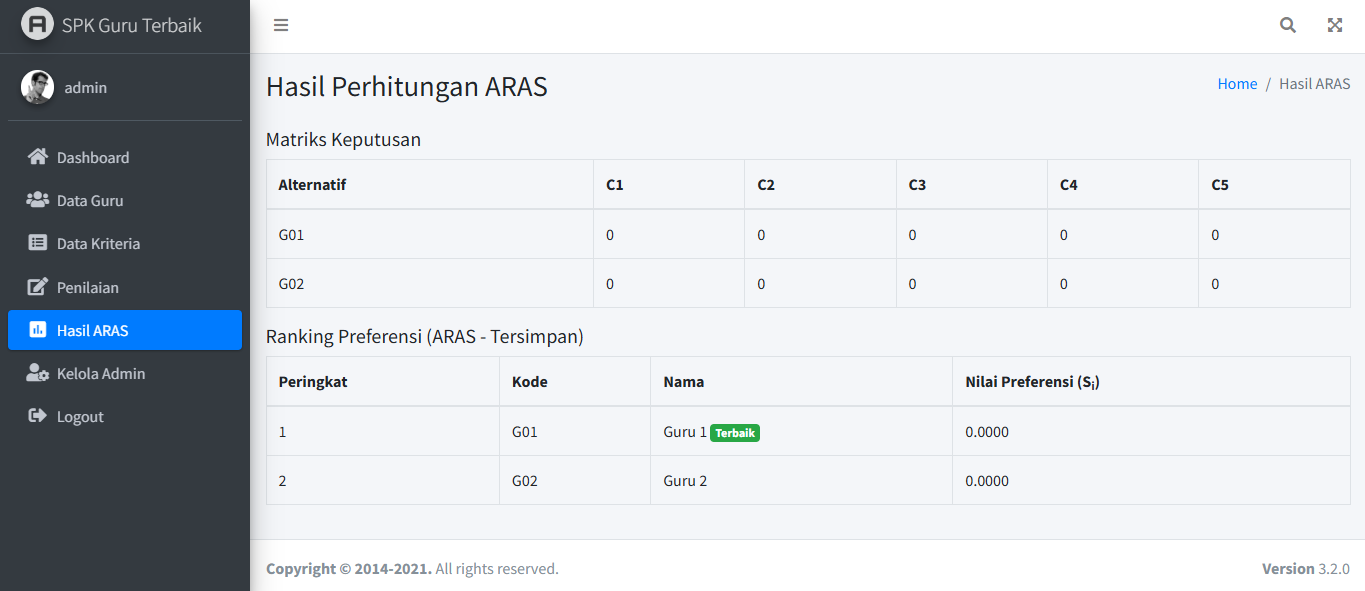
Halaman data kriteria digunakan untuk mengelola kriteria penilaian yang menjadi dasar dalam proses perhitungan metode ARAS. Admin dapat menambahkan kriteria baru dengan memasukkan kode, nama kriteria, bobot, serta tipe kriteria (benefit atau cost). Selain itu, admin juga dapat mengedit dan menghapus kriteria yang sudah ada. Data kriteria ini sangat penting karena akan memengaruhi hasil akhir dalam proses pengambilan keputusan. Desain halaman ini dibuat responsif dan user-friendly untuk memudahkan admin dalam mengelola kriteria yang digunakan.

### 4.2.5 Halaman Penilaian Alternatif



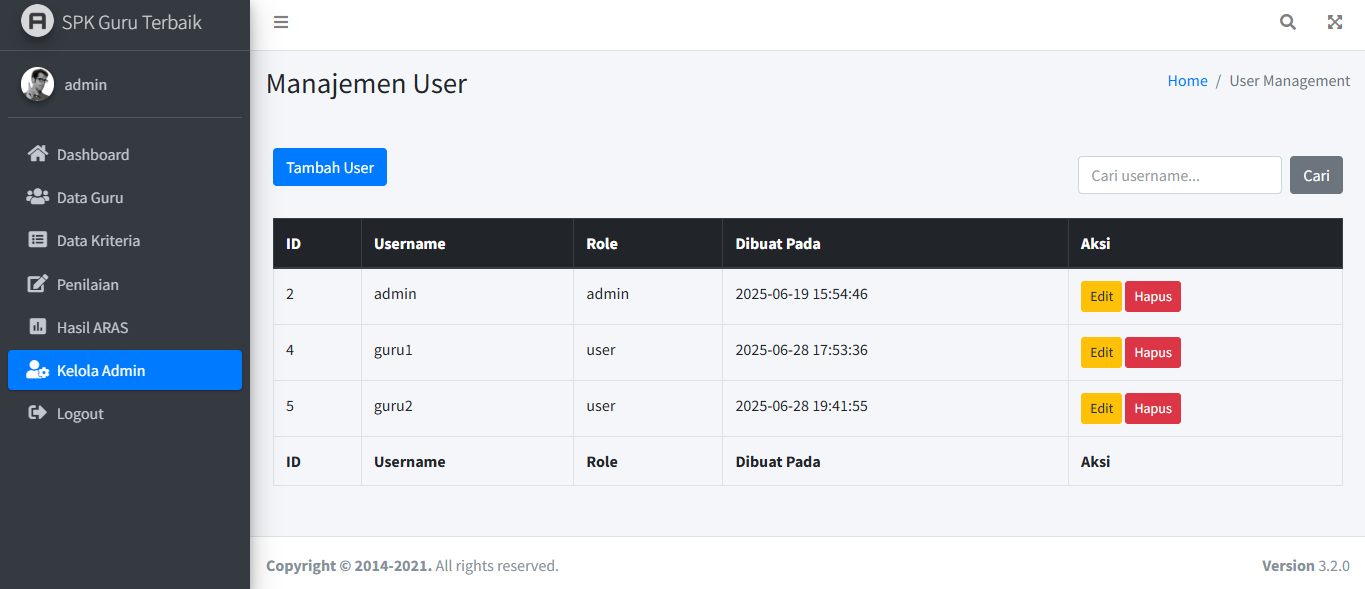
Halaman penilaian alternatif digunakan oleh admin untuk memberikan nilai terhadap setiap guru (alternatif) berdasarkan masing-masing kriteria yang telah ditentukan sebelumnya. Pada halaman ini, admin memilih guru, kemudian memasukkan nilai kriteria yang sesuai, baik berupa skor langsung maupun pemilihan sub-kriteria. Penilaian ini menjadi dasar utama dalam proses perhitungan metode ARAS. Sistem akan menyimpan nilai-nilai tersebut ke dalam database untuk kemudian diolah menjadi nilai preferensi dan ranking. Tampilan halaman dibuat sederhana, sistematis, dan mudah dioperasikan agar proses penilaian berjalan efektif.

### 4.2.6 Halaman Hasil Aras



Halaman hasil ARAS menampilkan output dari perhitungan metode Additive Ratio Assessment (ARAS) yang telah dilakukan berdasarkan data penilaian alternatif terhadap kriteria yang tersedia. Di halaman ini, sistem akan menampilkan nilai preferensi dari masing-masing alternatif (guru) beserta urutan peringkatnya. Nilai preferensi menggambarkan seberapa layak seorang guru dipilih berdasarkan bobot dan tipe kriteria yang telah ditentukan. Tujuan dari halaman ini adalah memberikan informasi keputusan yang akurat dan objektif kepada admin dalam menentukan guru terbaik. Hasil perhitungan ditampilkan dalam bentuk tabel yang mudah dipahami.

### 4.2.7 Halaman Kelola Admin



Halaman kelola admin berfungsi untuk mengatur data pengguna yang memiliki akses ke dalam sistem. Pada halaman ini, admin dapat menambahkan akun baru, mengedit informasi akun, ataupun menghapus akun yang sudah tidak digunakan. Fitur ini penting untuk menjaga keamanan dan kontrol akses terhadap sistem, agar hanya pengguna yang berwenang yang dapat mengakses fitur-fitur penting. Tampilan halaman kelola admin dibuat sederhana dan informatif agar memudahkan pengelolaan akun oleh pengguna yang memiliki hak akses.

## 4.3 Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa seluruh fitur dalam sistem pendukung keputusan pemilihan guru terbaik dapat berjalan dengan baik sesuai dengan kebutuhan dan perancangan yang telah dibuat. Dalam tahap ini digunakan dua metode pengujian, yaitu black-box testing dan white-box testing. Black-box testing digunakan untuk menguji fungsionalitas sistem dari sisi pengguna tanpa memperhatikan struktur internal program, sedangkan white-box testing dilakukan untuk menguji alur logika dalam program secara langsung, seperti percabangan, perulangan, dan alur kontrol. Kedua metode ini digunakan secara komplementer agar pengujian sistem dapat dilakukan secara menyeluruh, baik dari sisi tampilan maupun dari sisi logika pemrogramannya.

### 4.3.1 Pengujian Sistem Black Box

1. **Pengujian Black Box Login**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Skenario Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | Mengisi username dan password dengan data yang valid, lalu menekan tombol "Sign In". | Sistem menampilkan pesan login berhasil dan masuk ke halaman dashboard. | Sesuai | Valid |
| 2 | Mengisi username valid dan password salah, lalu menekan tombol "Sign In". | Sistem menampilkan pesan error "Password salah" atau "Kredensial tidak valid" dan tetap di halaman login. | Sesuai | Valid |
| 3 | Mengisi username yang tidak terdaftar, lalu menekan tombol "Sign In". | Sistem menampilkan pesan error "Username tidak ditemukan" dan tetap di halaman login. | Sesuai | Valid |
| 4 | Mengosongkan username dan password, lalu menekan tombol "Sign In". | Sistem menampilkan pesan validasi pada kedua kolom, seperti "Wajib diisi". | Sesuai | Valid |
| 5 | Mengisi username valid tetapi mengosongkan password, lalu menekan tombol "Sign In". | Sistem menampilkan pesan validasi pada kolom password. | Sesuai | Valid |

1. **Pengujian Black Box Data Guru**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Skenario Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | Melihat Data: Mengakses halaman Manajemen Guru. | Sistem menampilkan tabel berisi daftar semua data guru yang ada. | Sesuai | Valid |
| 2 | Menambah Data: Menekan tombol "Tambah Guru". | Sistem menampilkan halaman atau pop-up form untuk mengisi data guru baru. | Sesuai | Valid |
| 3 | Mengubah Data: Menekan tombol "Edit" pada salah satu baris data guru. | Sistem menampilkan halaman atau pop-up form yang sudah terisi dengan data guru yang dipilih, siap untuk diubah. | Sesuai | Valid |
| 4 | Menghapus Data: Menekan tombol "Hapus" pada salah satu baris data guru. | Sistem menampilkan dialog konfirmasi (misalnya: "Apakah Anda yakin?") sebelum menghapus data. | Sesuai | Valid |
| 5 | Melihat Detail: Menekan tombol "Detail" pada salah satu baris data guru. | Sistem menampilkan halaman atau pop-up yang berisi semua informasi detail dari guru yang dipilih. | Sesuai | Valid |
| 6 | Pencarian Data (Ditemukan): Mengisi kolom pencarian dengan nama guru yang ada (contoh: "Guru 1") dan menekan tombol "Cari". | Sistem menampilkan tabel yang hanya berisi data guru dengan nama "Guru 1". | Sesuai | Valid |
| 7 | Pencarian Data (Tidak Ditemukan): Mengisi kolom pencarian dengan nama guru yang tidak ada (contoh: "Guru 99") dan menekan tombol "Cari". | Sistem menampilkan tabel kosong dengan pesan "Data tidak ditemukan". | Sesuai | Valid |

1. **Pengujian Black Box Data Kriteria**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Skenario Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | Melihat Data: Mengakses halaman Manajemen Kriteria. | Sistem menampilkan tabel berisi daftar semua data kriteria yang ada. | Sesuai | Valid |
| 2 | Menambah Data: Menekan tombol "Tambah Kriteria". | Sistem menampilkan halaman atau form untuk mengisi data kriteria baru (kode, nama, bobot, tipe). | Sesuai | Valid |
| 3 | Mengubah Data: Menekan tombol "Edit" pada salah satu baris data kriteria. | Sistem menampilkan halaman form yang sudah terisi dengan data kriteria yang dipilih, siap untuk diubah. | Sesuai | Valid |
| 4 | Menghapus Data: Menekan tombol "Hapus" pada salah satu baris data kriteria. | Sistem menampilkan dialog konfirmasi ("Apakah Anda yakin?") sebelum menghapus data. | Sesuai | Valid |
| 5 | Navigasi Sub Kriteria: Menekan tombol "Sub Kriteria" pada salah satu baris. | Sistem mengarahkan pengguna ke halaman baru untuk mengelola data sub kriteria dari kriteria yang dipilih. | Sesuai | Valid |
| 6 | Pencarian Data (Ditemukan): Mengisi kolom pencarian dengan nama kriteria yang ada (contoh: "Kehadiran") dan menekan tombol "Cari". | Sistem menampilkan tabel yang hanya berisi data kriteria yang cocok dengan pencarian. | Sesuai | Valid |
| 7 | Pencarian Data (Tidak Ditemukan): Mengisi kolom pencarian dengan nama kriteria yang tidak ada dan menekan tombol "Cari". | Sistem menampilkan tabel kosong dengan pesan "Data tidak ditemukan". | Sesuai | Valid |

1. **Pengujian Black Box Penilaian Alternatif**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Skenario Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | Menyimpan Penilaian (Valid): Mengisi semua kolom penilaian dengan data yang valid (angka atau pilihan dropdown), lalu menekan tombol "Simpan Penilaian". | Sistem berhasil menyimpan semua data nilai ke database dan menampilkan pesan sukses. | Sesuai | Valid |
| 2 | Menyimpan Penilaian (Input Kosong): Membiarkan satu atau lebih kolom penilaian kosong, lalu menekan tombol "Simpan Penilaian". | Sistem menampilkan pesan error validasi, seperti "Semua nilai wajib diisi", dan data tidak tersimpan. | Sesuai | Valid |
| 3 | Menyimpan Penilaian (Input Bukan Angka): Mengisi salah satu kolom penilaian dengan data yang tidak valid (contoh: huruf "abc"), lalu menekan tombol "Simpan Penilaian". | Sistem menampilkan pesan error validasi, seperti "Input harus berupa angka", dan data tidak tersimpan. | Sesuai | Valid |
| 4 | Mengubah Nilai: Mengubah nilai yang sudah ada di salah satu kolom dan menekan tombol "Simpan Penilaian". | Sistem berhasil memperbarui (update) nilai tersebut di database, bukan membuat data baru. | Sesuai | Valid |

1. **Pengujian Black Box Kelola Admin**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Skenario Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | Melihat Data: Mengakses halaman Kelola Admin. | Sistem menampilkan tabel berisi daftar semua data pengguna (admin dan user lain). | Sesuai | Valid |
| 2 | Menambah Data: Menekan tombol "Tambah User". | Sistem menampilkan halaman atau form untuk mengisi data pengguna baru (username, password, role). | Sesuai | Valid |
| 3 | Mengubah Data: Menekan tombol "Edit" pada salah satu baris pengguna. | Sistem menampilkan halaman form yang sudah terisi dengan data pengguna yang dipilih, siap untuk diubah. | Sesuai | Valid |
| 4 | Menghapus Data Pengguna Lain: Menekan tombol "Hapus" pada baris pengguna lain (bukan yang sedang login). | Sistem menampilkan dialog konfirmasi ("Apakah Anda yakin?") sebelum menghapus data. | Sesuai | Valid |
| 5 | Menghapus Akun Sendiri: Mencoba menekan tombol "Hapus" pada baris pengguna yang sedang aktif/login. | Tombol "Hapus" tidak dapat diklik (disabled) atau sistem menampilkan pesan error "Anda tidak dapat menghapus akun sendiri". | Sesuai | Valid |
| 6 | Pencarian Data (Ditemukan): Mengisi kolom pencarian dengan username yang ada (contoh: "guru1") dan menekan tombol "Cari". | Sistem menampilkan tabel yang hanya berisi data pengguna yang cocok dengan pencarian. | Sesuai | Valid |

1. **Pengujian Black Box Logout**

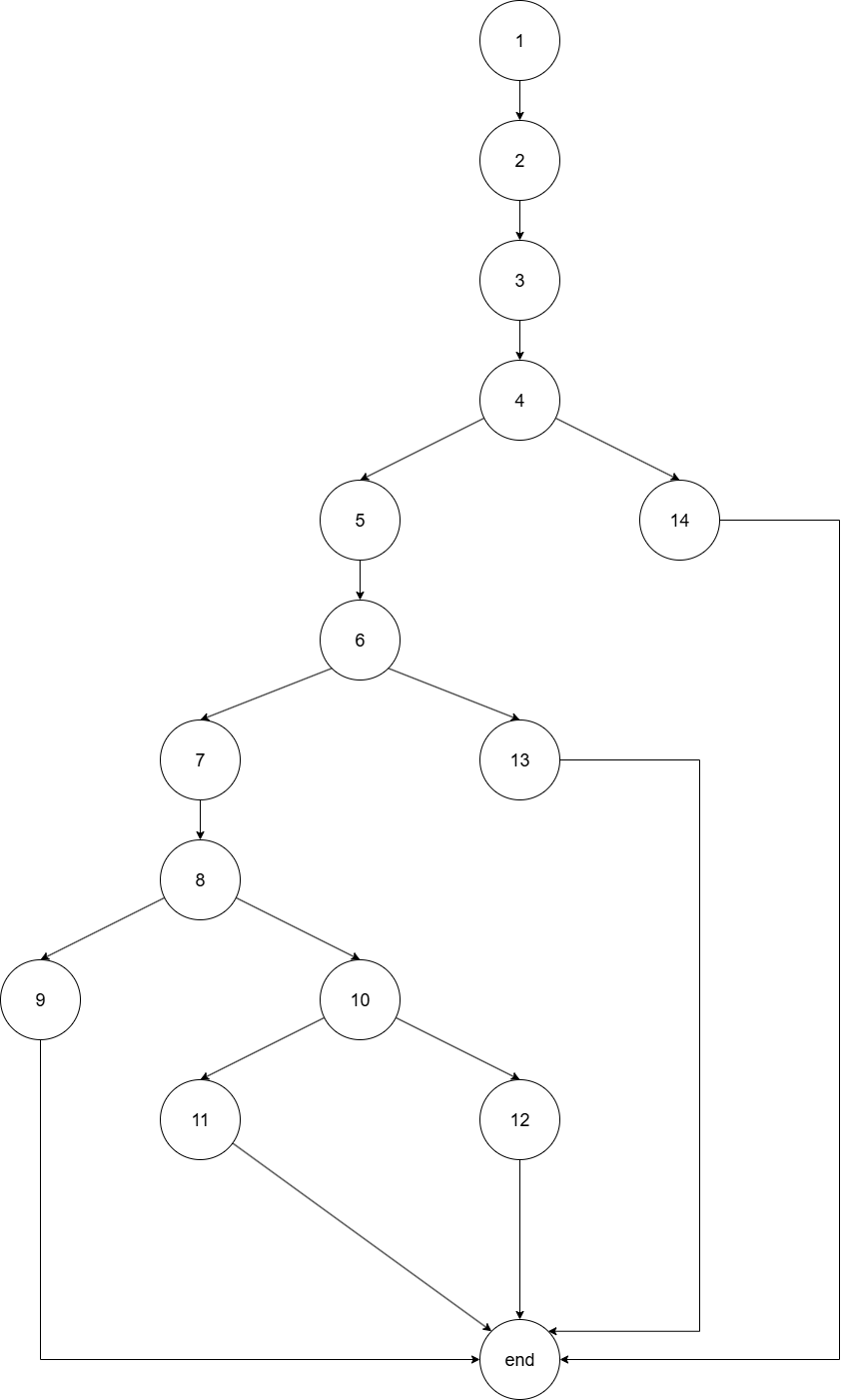
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Skenario Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | Logout Normal: Menekan tombol "Logout" pada sidebar saat sedang login. | Sistem mengakhiri sesi pengguna dan langsung mengarahkan (redirect) ke halaman login. | Sesuai | Valid |
| 2 | Akses Setelah Logout: Setelah logout, mencoba mengakses halaman dashboard kembali (misalnya dengan menekan tombol "Back" di browser). | Sistem menolak akses dan mengembalikan pengguna ke halaman login. | Sesuai | Valid |

### 4.3.2 Pengujian Sistem White Box

Setelah melakukan pengujian black box untuk memvalidasi fungsionalitas sistem dari sudut pandang pengguna, tahap pengujian selanjutnya adalah pengujian white box. Berbeda dengan pengujian black box, pengujian white box berfokus pada analisis dan pengujian struktur internal dari kode program. Pengujian ini dilakukan untuk memastikan bahwa semua jalur logika, percabangan kondisional (if/else), perulangan (loop), dan operasi internal dalam sebuah modul atau fungsi telah berjalan sesuai dengan yang dirancang. Tujuan utama dari pengujian white box adalah untuk memverifikasi alur kerja internal, menemukan potensi *bug* atau *error* dalam logika kode, dan menjamin bahwa setiap baris kode memberikan kontribusi yang benar terhadap hasil akhir dari sebuah proses. Pada tahap ini, pengujian akan difokuskan pada salah satu modul krusial sistem, yaitu modul perhitungan metode ARAS, untuk menelusuri dan memvalidasi alur logikanya secara mendetail.

1. **Pengujian White Box Login**

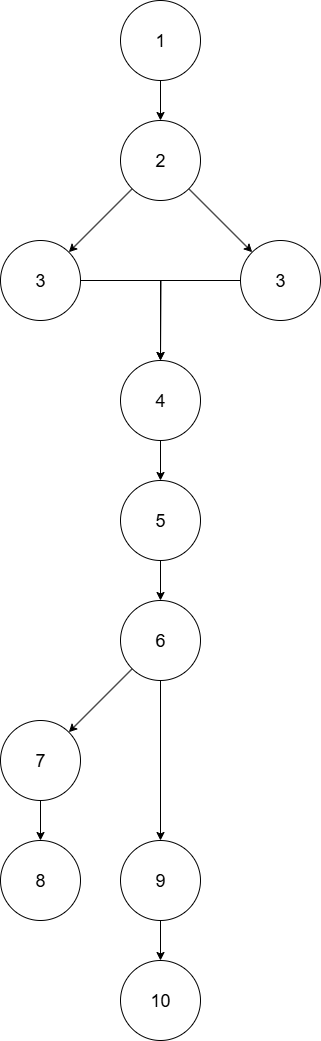
|  |  |
| --- | --- |
| **Node** | **Source Code** |
| 1 | session\_start();  include '../config/config.php';  include '../config/koneksi.php'; |
| 2 | if ($\_SERVER['REQUEST\_METHOD'] == 'POST') { |
| 3 | $username = $\_POST['username'];  $password = $\_POST['password'];  $sql = "SELECT \* FROM users WHERE username = ?";  $stmt = $koneksidb->prepare($sql);  $stmt->bind\_param("s", $username);  $stmt->execute();  $result = $stmt->get\_result(); |
| 4 | if ($result->num\_rows > 0) { |
| 5 | $user = $result->fetch\_assoc(); |
| 6 | if (password\_verify($password, $user['password'])) { |
| 7 | $\_SESSION['user'] = [ 'id' => $user['id'], 'username' => $user['username'], 'role' => $user['role'] ]; |
| 8 | if ($user['role'] == 'admin') { |
| 9 | $\_SESSION['login\_success'] = 'Selamat datang, ' . $user['username'] . '!';  header("Location: " . base\_url('dashboard/admin/')); |
| 10 | } else if ($user['role'] == 'user') { |
| 11 | $\_SESSION['login\_success'] = 'Selamat datang, ' . $user['username'] . '!';  header("Location: " . base\_url('dashboard/user/')); |
| 12 | } else {  $\_SESSION['login\_error'] = 'Role tidak dikenali!';  header("Location: login.php");  exit;  } |
| 13 | } else {  $\_SESSION['login\_error'] = 'Password salah!';  header("Location: login.php");  exit;  } |
| 14 | } else {  $\_SESSION['login\_error'] = 'Username tidak ditemukan!';  header("Location: login.php");  exit;  } |
| 15 | } |



|  |  |  |
| --- | --- | --- |
| **Jalur** | **Independent Path** | **Keterangan** |
| 1 | 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 | Login berhasil sebagai admin |
| 2 | 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 10 → 11 | Login berhasil sebagai user |
| 3 | 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 10 → 12 | Role tidak dikenali → login gagal |
| 4 | 1 → 2 → 3 → 4 → 5 → 6 → 13 | Password salah → login gagal |
| 5 | 1 → 2 → 3 → 4 → 14 | Username tidak ditemukan → gagal |
| 6 | 1 → 2 | Kasus saat request bukan POST |

1. **Pengujian White Box Data Guru**

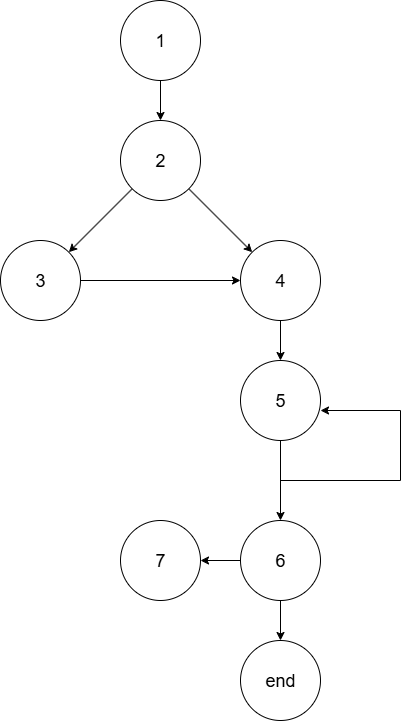
|  |  |
| --- | --- |
| **Node** | **Source Code** |
| 1 | <?php  $search = isset($\_GET['search']) ? mysqli\_real\_escape\_string($koneksidb, $\_GET['search']) : '';  $query = "SELECT a.\*, u.username  FROM alternatives a  JOIN users u ON a.user\_id = u.id"; |
| 2 | if (!empty($search)) {  $query .= " WHERE a.name LIKE '%$search%' OR a.code LIKE '%$search%' OR u.username LIKE '%$search%'";  } |
| 3 | $alternatif = mysqli\_query($koneksidb, $query);  ?> |
| 4 | <div class="container mt-4">  <div class="d-flex justify-content-between align-items-center mb-3">  <a href="create.php" class="btn btn-primary mb-3">Tambah Guru</a> |
| 5 | <form class="form-inline" method="GET">  <input type="text" name="search" class="form-control" placeholder="Cari kode/nama/username..."  value="<?= isset($\_GET['search']) ? htmlspecialchars($\_GET['search']) : '' ?>">  <button type="submit" class="btn btn-secondary ml-2">Cari</button>  </form>  </div>  <table class="table table-bordered table-hover">  <thead class="thead-dark">  <tr>  <th>Kode</th>  <th>Username</th>  <th>Nama Guru</th>  <th>Aksi</th>  </tr>  </thead> |
| 6 | <tbody>  <?php while ($row = mysqli\_fetch\_assoc($alternatif)): ?> |
| 7 | <tr>  <td><?= htmlspecialchars($row['code']) ?></td>  <td><?= htmlspecialchars($row['username']) ?></td>  <td><?= htmlspecialchars($row['name']) ?></td>  <td>  <button class="btn btn-sm btn-info" data-toggle="modal" data-target="#detailModal<?= $row['id'] ?>"> Detail  </button>  <a href="edit.php?id=<?= $row['id'] ?>" class="btn btn-sm btn-warning">Edit</a>  <a href="#" class="btn btn-sm btn-danger" onclick="confirmDelete(<?= $row['id'] ?>)">Hapus</a>  </td>  </tr> |
| 8 | <div class="modal fade" id="detailModal<?= $row['id'] ?>" tabindex="-1" role="dialog" aria-labelledby="modalLabel<?= $row['id'] ?>" aria-hidden="true">  <div class="modal-dialog modal-dialog-centered" role="document">  <div class="modal-content">  <div class="modal-header">  <h5 class="modal-title" id="modalLabel<?= $row['id'] ?>">Detail Guru</h5>  <button type="button" class="close" data-dismiss="modal" aria-label="Tutup">  <span aria-hidden="true">&times;</span>  </button>  </div>  <div class="modal-body">  <p><strong>Kode:</strong> <?= htmlspecialchars($row['code']) ?></p>  <p><strong>Username:</strong> <?= htmlspecialchars($row['username']) ?></p>  <p><strong>Nama:</strong> <?= htmlspecialchars($row['name']) ?></p>  <p><strong>Alamat:</strong> <?= htmlspecialchars($row['alamat']) ?></p>  <p><strong>No HP:</strong> <?= htmlspecialchars($row['no\_hp']) ?></p>  <p><strong>Umur:</strong> <?= htmlspecialchars($row['umur']) ?> tahun</p>  <p><strong>Created At:</strong> <?= htmlspecialchars($row['created\_at']) ?></p>  </div>  <div class="modal-footer">  <button type="button" class="btn btn-secondary" data-dismiss="modal">Tutup</button>  </div>  </div>  </div>  </div>  <?php endwhile; ?>  </tbody>  </table>  </div> |
| 9 | <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>  <script>  function confirmDelete(id) {  const baseUrl = "<?= base\_url('dashboard/admin/alternatif/delete.php') ?>";  Swal.fire({  title: 'Yakin ingin menghapus?',  text: "Data ini tidak dapat dikembalikan!",  icon: 'warning',  showCancelButton: true,  confirmButtonColor: '#d33',  cancelButtonColor: '#3085d6',  confirmButtonText: 'Ya, hapus!',  cancelButtonText: 'Batal'  }).then((result) => {  if (result.isConfirmed) {  window.location.href = baseUrl + '?id=' + id;  }  })  }  </script> |
| 10 | <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js"></script> |



|  |  |  |
| --- | --- | --- |
| **Jalur** | **Independent Path** | **Keterangan** |
| 1 | 1 → 2 (false) → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 10 | Eksekusi dilakukan saat tidak ada input pencarian, data ditampilkan semua |
| 2 | 1 → 2 (true) → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 10 | Eksekusi dilakukan saat ada input pencarian, query ditambahkan kondisi pencarian |

1. **Pengujian White Box Data Kriteria**

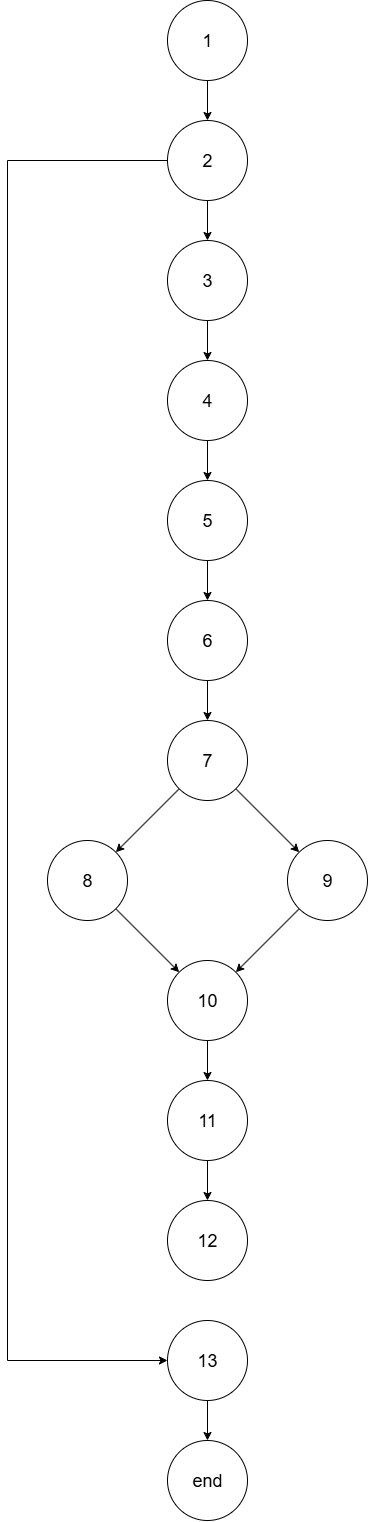
|  |  |
| --- | --- |
| **Node** | **Source Code** |
| 1 | <?php  $search = isset($\_GET['search']) ? mysqli\_real\_escape\_string($koneksidb, $\_GET['search']) : '';  $query = "SELECT \* FROM criteria"; |
| 2 | if (!empty($search)) { |
| 3 | $query .= " WHERE name LIKE '%$search%' OR code LIKE '%$search%'";  } |
| 4 | $kriteria = mysqli\_query($koneksidb, $query);  ?>  <div class="container mt-4">  <div class="d-flex justify-content-between align-items-center mb-3">  <a href="create.php" class="btn btn-primary mb-3">Tambah Kriteria</a>  <form class="form-inline" method="GET">  <input type="text" name="search" class="form-control" placeholder="Cari kode/nama..."  value="<?= isset($\_GET['search']) ? htmlspecialchars($\_GET['search']) : '' ?>">  <button type="submit" class="btn btn-secondary ml-2">Cari</button>  </form>  </div>  <table class="table table-bordered table-hover">  <thead class="thead-dark">  <tr>  <th>ID</th>  <th>Kode</th>  <th>Nama</th>  <th>Bobot</th>  <th>Tipe</th>  <th>Created At</th>  <th>Aksi</th>  </tr>  </thead> |
| 5 | <tbody>  <?php while ($row = mysqli\_fetch\_assoc($kriteria)): ?>  <tr>  <td><?= $row['id'] ?></td>  <td><?= htmlspecialchars($row['code']) ?></td>  <td><?= htmlspecialchars($row['name']) ?></td>  <td><?= $row['weight'] ?></td>  <td><?= ucfirst($row['type']) ?></td>  <td><?= $row['created\_at'] ?></td>  <td>  <a href="edit.php?id=<?= $row['id'] ?>" class="btn btn-sm btn-warning">Edit</a>  <a href="#" class="btn btn-sm btn-danger" onclick="confirmDelete(<?= $row['id'] ?>)">Hapus</a>  <a href="<?= base\_url('dashboard/admin/subkriteria/index.php?criterion\_id=' . $row['id']) ?>"  class="btn btn-sm btn-info">Sub Kriteria</a>  </td>  </tr>  <?php endwhile; ?>  </tbody>  <tfoot>  <tr>  <th>ID</th>  <th>Kode</th>  <th>Nama</th>  <th>Bobot</th>  <th>Tipe</th>  <th>Created At</th>  <th>Aksi</th>  </tr>  </tfoot>  </table>  </div> |
| 6 | <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>  <script>  function confirmDelete(id) {  const baseUrl = "<?= base\_url('dashboard/admin/kriteria/delete.php') ?>";  Swal.fire({  title: 'Yakin ingin menghapus?',  text: "Data ini tidak dapat dikembalikan!",  icon: 'warning',  showCancelButton: true,  confirmButtonColor: '#d33',  cancelButtonColor: '#3085d6',  confirmButtonText: 'Ya, hapus!',  cancelButtonText: 'Batal' |
| 7 | }).then((result) => {  if (result.isConfirmed) {  window.location.href = baseUrl + '?id=' + id;  }  })  }  </script> |



|  |  |  |
| --- | --- | --- |
| **Jalur** | **Independent Path** | **Keterangan** |
| 1 | 1 → 2 (false) → 4 → 5 → 6 → 7 | Tidak ada pencarian, data tampil, tidak hapus |
| 2 | 1 → 2 (true) → 3 → 4 → 5 → 6 → 7 | Ada pencarian, data tampil, tidak hapus |
| 3 | 1 → 2 (true) → 3 → 4 → 5 (loop ≥ 2x) → 6 → 7 | Ada pencarian, data lebih dari 1, ditampilkan semua |
| 4 | 1 → 2 → 4 → 5 → 6 → 7 (konfirmasi hapus) → Redirect | Proses hapus dikonfirmasi, data dihapus |

1. **Pengujian White Box Penilaian Alternatif**

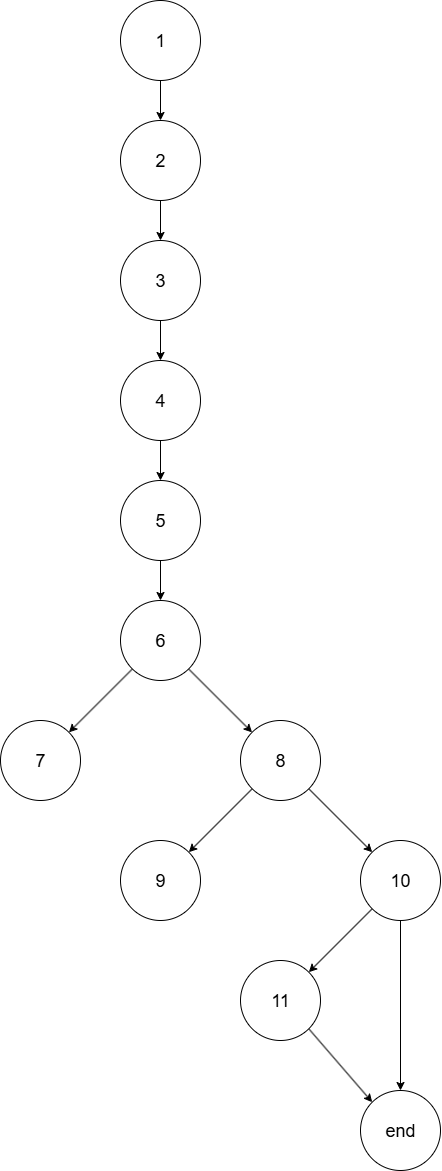
|  |  |
| --- | --- |
| **Node** | **Source Code** |
| 1 | <?php  $alternatif = mysqli\_query($koneksidb, "SELECT \* FROM alternatives ORDER BY code ASC");  $kriteria = mysqli\_query($koneksidb, "SELECT \* FROM criteria ORDER BY code ASC"); |
| 2 | if ($\_SERVER['REQUEST\_METHOD'] === 'POST') { |
| 3 | foreach ($\_POST['nilai'] as $alt\_id => $krit) { |
| 4 | foreach ($krit as $krit\_id => $data) { |
| 5 | $nilai = isset($data['score']) ? intval($data['score']) : null;  $sub\_id = isset($data['sub\_criterion\_id']) ? (int) $data['sub\_criterion\_id'] : null; |
| 6 | $cek = mysqli\_query($koneksidb, "SELECT id FROM scores WHERE alternative\_id = $alt\_id AND criterion\_id = $krit\_id"); |
| 7 | if (mysqli\_num\_rows($cek) > 0) { |
| 8 | mysqli\_query($koneksidb, "UPDATE scores SET score = " . ($nilai ?? 'NULL') . ", sub\_criterion\_id = " . ($sub\_id ?: 'NULL') . " WHERE alternative\_id = $alt\_id AND criterion\_id = $krit\_id"); |
| 9 | } else {  mysqli\_query($koneksidb, "INSERT INTO scores (alternative\_id, criterion\_id, score, sub\_criterion\_id) VALUES ($alt\_id, $krit\_id, " . ($nilai ?? 'NULL') . ", " . ($sub\_id ?: 'NULL') . ")");  } |
| 10 | } |
| 11 | } |
| 12 | header("Location: index.php?success=simpan");  exit; |
| 13 | } |



|  |  |  |
| --- | --- | --- |
| **Jalur** | **Independent Path** | **Keterangan** |
| 1 | 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 10 → 11 → 12 | Jalur ketika data ditemukan dan dilakukan UPDATE ke database. |
| 2 | 1 → 2 → 3 → 4 → 5 → 6 → 7 → 9 → 10 → 11 → 12 | Jalur ketika data belum ada dan dilakukan INSERT ke database. |
| 3 | 1 → 2 → 13 | Jalur ketika method bukan POST sehingga tidak terjadi pemrosesan data. |

1. **Pengujian White Box Kelola Admin**

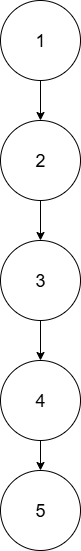
|  |  |
| --- | --- |
| **Node** | **Source Code** |
| 1 | <?php  include '../../../config/config.php';  include '../../../config/koneksi.php';  session\_start(); |
| 2 | cek\_admin(); |
| 3 | $page\_title = 'Manajemen User';  $breadcrumbs = [  ['title' => 'Home', 'link' => base\_url('dashboard/admin')],  ['title' => 'User Management']  ]; |
| 4 | $content = base\_path('dashboard/admin/user/home.php'); |
| 5 | include base\_path('layout/main.php');  ?> |
| 6 | <?php if (isset($\_GET['success']) && $\_GET['success'] == 'delete'): ?> |
| 7 | <script>  Swal.fire({  title: 'Berhasil!',  text: 'User berhasil dihapus.',  icon: 'success',  confirmButtonText: 'Oke'  })  </script>  <?php endif; ?> |
| 8 | <?php if (isset($\_GET['success']) && $\_GET['success'] == 'simpan'): ?> |
| 9 | <script>  Swal.fire({  title: 'Berhasil!',  text: 'User berhasil ditambah.',  icon: 'success',  confirmButtonText: 'Oke'  })  </script>  <?php endif; ?> |
| 10 | <?php if (isset($\_GET['success']) && $\_GET['success'] == 'update'): ?> |
| 11 | <script>  Swal.fire({  title: 'Berhasil!',  text: 'User berhasil diubah.',  icon: 'success',  confirmButtonText: 'Oke'  })  </script>  <?php endif; ?> |

****

|  |  |  |
| --- | --- | --- |
| **Jalur** | **Independent Path** | **Keterangan** |
| 1 | 1 → 2 → 3 → 4 → 5 → 6 (false) → 8 (false) → 10 (false) → END | |  | | --- | |  |  |  | | --- | | Jalur tanpa notifikasi sukses |   UPDATE ke database. |
| 2 | 1 → 2 → 3 → 4 → 5 → 6 (true) → 7 → END | Jalur ketika ?success=delete, menampilkan notifikasi "User berhasil dihapus" |
| 3 | 1 → 2 → 3 → 4 → 5 → 6 (false) → 8 (true) → 9 → END | Jalur ketika ?success=simpan, menampilkan notifikasi "User berhasil ditambah" |

1. **Pengujian White Box Logout**

|  |  |
| --- | --- |
| **Node** | **Source Code** |
| 1 | <?php  session\_start(); |
| 2 | session\_destroy(); |
| 3 | include '../config/config.php'; |
| 4 | header('Location: ' . base\_url('login/login.php')); |
| 5 | exit;  ?> |



|  |  |  |
| --- | --- | --- |
| **Jalur** | **Independent Path** | **Keterangan** |
| 1 | 1 → 2 → 3 → 4 → 5 | |  | | --- | |  |   Jalur tunggal logout: memulai sesi, menghancurkan sesi, mengarahkan user ke halaman login |