**SISTEM PENUNJANG KEPUTUSAN UNTUK PEMILIHAN GURU TERBAIK MENGGUNAKAN METODE ADDITIVE RATIO ASSESSMENT (ARAS) BERBASIS WEB (STUDI KASUS: SDI IBNU QOYYIM ILMI)**

SKRIPSI

Oleh:

Muhamad Thoriq Maulidka

211011400576

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS PAMULANG**

**TANGERANG SELATAN**

**2025**

**SISTEM PENUNJANG KEPUTUSAN UNTUK PEMILIHAN GURU TERBAIK MENGGUNAKAN METODE ADDITIVE RATIO ASSESSMENT (ARAS) BERBASIS WEB (STUDI KASUS: SDI IBNU QOYYIM ILMI)**

**SKRIPSI**

****Diajukan Untuk Melengkapi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer

Oleh:

Muhamad Thoriq Maulidka

211011400576

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS PAMULANG**

**TANGERANG SELATAN**

**2025**

# LEMBAR PERNYATAAN

# LEMBAR PERSETUJUAN

# LEMBAR PENGESAHAN

# *ABSTRACT*

# ABSTRAK

# KATA PENGANTAR

Puji syukur Alhamdulillah kehadirat Allah *subahanahu wa ta’alaa* yang telah melimpahkan segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan skripsi yang merupakan salah satu persyaratan untuk menyelesaikan program studi strata satu (S1) pada program studi Teknik Informatika di Universitas Pamulang. Dengan segala keterbatasan, penulis menyadari pula bahwa skripsi ini tidak akan terwujud tanpa bantuan, bimbingan, dan dorongan dari berbagai pihak. Untuk itu, dengan segala kerendahan hati, penulis menyampaikan ucapan terima kasih kepada:

1. Bapak Dr. Pranoto, S.E., M.M., selaku Ketua Yayasan Sasmita Jaya.
2. Bapak Dr. E. Nurzaman, M.Si. selaku Rektor Universitas Pamulang.
3. Bapak Yan Mitha Djaksana, S.Kom., M.Kom. Dekan Fakultas Ilmu Komputer Universitas Pamulang.
4. Bapak Dr. Eng., Ahmad Musyafa, S.Kom., M.Kom. selaku Ketua Program Studi Teknik Informatika Universitas Pamulang.
5. Bapak Dr. Eng., Ahmad Musyafa, S.Kom., M.Kom. selaku pembimbing skripsi pada Program Studi Teknik Informatika di Universitas Pamulang.
6. Kedua Orang Tua yang saya cintai yang telah memberikan baik dukungan moral maupun materi yang sangat melimpah dalam proses penulisan skripsi, sehingga penulis sangat termotivasi untuk segera menyelesaikan penulisan skripsi ini.
7. Teman-Teman seperjuangan kelas **TPLP008** Reguler A angkatan 2021 yang telah memberikan semangat dan dukungannya dalam pembuatan skripsi ini.
8. Serta semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam penyusunan skripsi ini.

Kritik dan saran sangat penulis harapkan guna terciptanya sebuah karya ilmiah yang berkualitas yang berguna kelak bagi penulis sendiri maupun bagi pembaca.

Akhir kata penulis berharap skripsi ini dapat bermanfaat bagi semua pihak terutama bagi penulis sendiri baik sebagai bahan karya tulis berupa informasi, perbandingan maupun dasar untuk penelitian materi lebih lanjut.

Pamulang, 19 September 2024

Penulis

# DAFTAR ISI

[LEMBAR PERNYATAAN III](#_Toc202712308)

[LEMBAR PERSETUJUAN IV](#_Toc202712309)

[LEMBAR PENGESAHAN V](#_Toc202712310)

[*ABSTRACT* VI](#_Toc202712311)

[ABSTRAK VII](#_Toc202712312)

[KATA PENGANTAR VIII](#_Toc202712313)

[DAFTAR ISI X](#_Toc202712314)

[DAFTAR GAMBAR XIII](#_Toc202712315)

[DAFTAR TABEL XIV](#_Toc202712316)

[BAB I PENDAHULUAN 1](#_Toc202712317)

[1.1 Latar Belakang 1](#_Toc202712318)

[1.2 Identifikasi Masalah 3](#_Toc202712319)

[1.3 Rumusan Masalah 4](#_Toc202712320)

[1.4 Batasan Masalah 4](#_Toc202712321)

[1.5 Tujuan Penelitian 5](#_Toc202712322)

[1.6 Manfaat Penelitian 5](#_Toc202712323)

[1.7 Sistematika Penulisan 7](#_Toc202712324)

[BAB II LANDASAN TEORI 9](#_Toc202712325)

[2.1 Penelitian Relevan 9](#_Toc202712326)

[2.2 Tinjauan Pustaka 15](#_Toc202712327)

[2.2.1 Definisi Sistem 15](#_Toc202712328)

[2.2.2 Aplikasi 16](#_Toc202712329)

[2.2.3 Pengujian Perangkat Lunak 17](#_Toc202712330)

[2.2.4 Sistem Penunjang Keputusan 20](#_Toc202712331)

[2.2.5 Pendidikan 21](#_Toc202712332)

[2.2.6 Pengertian Metode *Additive Ratio Assessment (ARAS)* 22](#_Toc202712333)

[2.2.7 Aplikasi Berbasis *Web* 26](#_Toc202712334)

[2.3 UML 27](#_Toc202712335)

[2.3.1 Use Case Diagram 28](#_Toc202712336)

[2.3.2 Activity Diagram 29](#_Toc202712337)

[2.3.3 Sequence Diagram 30](#_Toc202712338)

[2.3.4 Class Diagram 32](#_Toc202712339)

[2.4 Perancangan Basis Data 33](#_Toc202712340)

[2.4.2 Basis Data 34](#_Toc202712341)

[2.4.2*Entity Relationship Diagram (ERD)* 34](#_Toc202712342)

[2.4.3 *Logical Record Structure (LRS)* 36](#_Toc202712343)

[2.4.4 Normalisasi 37](#_Toc202712344)

[2.5 Aplikasi Pendukung 38](#_Toc202712345)

[2.5.1 XAMPP 39](#_Toc202712346)

[2.5.2 HTML 39](#_Toc202712347)

[2.5.3 CSS 41](#_Toc202712348)

[2.5.4 PHP 42](#_Toc202712349)

[2.5.5 Visual Studio Code 43](#_Toc202712350)

[2.5.6 Bootstrap 43](#_Toc202712351)

[2.5.7 Draw.io 43](#_Toc202712352)

[2.6 Kerangka Pemikiran 44](#_Toc202712353)

[BAB III ANALISA DAN PERANCANGAN SISTEM 46](#_Toc202712354)

[3.1 Analisa Sistem 46](#_Toc202712355)

[3.1.1 Analisa Sistem Berjalan 46](#_Toc202712356)

[3.1.2 Analisa Sistem Usulan 49](#_Toc202712357)

[3.1.3 Penerapan Perhitungan Motode ARAS(BLM DAPET DATA) 50](#_Toc202712358)

[3.2 Perancangan Basis Data 50](#_Toc202712359)

[3.2.1 Entity Relationship Diagram (ERD) 51](#_Toc202712360)

[3.2.2 Transformasi ERD ke Logical Record Structure (LRS) 53](#_Toc202712361)

[3.2.3 Logical Record Structure (LRS) 53](#_Toc202712362)

[3.2.4 Normalisasi(SKIP) 54](#_Toc202712363)

[3.2.5 Spesifikasi Basis Data 54](#_Toc202712364)

[3.3 Perancangan Unified Modeling Language (UML) 59](#_Toc202712365)

[3.3.1 Use Case Diagram 59](#_Toc202712366)

[3.3.2 Activity Diagram 64](#_Toc202712367)

[3.3.3 Sequence Diagram 76](#_Toc202712368)

[3.3.4 Class Diagram 85](#_Toc202712369)

[3.4 User Interface 87](#_Toc202712370)

[DAFTAR PUSTAKA 94](#_Toc202712371)

# DAFTAR GAMBAR

# DAFTAR TABEL

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Dalam dunia pendidikan, guru merupakan komponen utama yang memegang peranan penting dalam proses pembelajaran dan pencapaian tujuan pendidikan. Kualitas dan kompetensi seorang guru sangat menentukan mutu pendidikan di suatu lembaga. Oleh karena itu, pemilihan guru terbaik menjadi langkah strategis dalam meningkatkan kualitas lembaga pendidikan, khususnya di tingkat sekolah dasar. Namun demikian, proses pemilihan guru terbaik sering kali dilakukan secara subjektif, tanpa adanya standar penilaian yang terstruktur dan terukur, sehingga berpotensi menimbulkan ketidakadilan atau kesalahan dalam pengambilan keputusan.

SDI Ibnu Qoyyim Ilmi sebagai salah satu institusi pendidikan dasar berbasis Islam tentunya memiliki kebutuhan untuk mengevaluasi dan mengapresiasi kinerja guru secara objektif. Namun, proses pemilihan guru terbaik di sekolah ini masih dilakukan secara manual dan belum memiliki acuan penilaian yang terdokumentasi dengan baik. Hal ini menyebabkan proses pemilihan cenderung subjektif, rawan terjadi ketidaktepatan dalam penginputan data, serta berisiko menghasilkan keputusan yang tidak mencerminkan kinerja guru secara menyeluruh dan akurat.

Selain itu, proses pemilihan guru terbaik selama ini masih dilakukan secara manual, tanpa adanya sistem terotomatisasi yang dapat menyimpan dan mengelola data penilaian secara terstruktur. Hal ini menyebabkan proses evaluasi menjadi kurang efisien, memakan waktu yang lama, dan rawan terhadap subjektivitas dalam pengambilan keputusan. Ketiadaan sistem digital juga menyulitkan pihak sekolah untuk melakukan evaluasi secara berkelanjutan dan membandingkan kinerja antar guru secara menyeluruh berdasarkan data yang valid. Oleh karena itu, dibutuhkan suatu sistem yang mampu membantu proses pengambilan keputusan dengan mempertimbangkan berbagai aspek penilaian secara sistematis dan transparan.

Sistem penunjang keputusan adalah salah satu komponen dalam sistem informasi berbasis komputer yang berfungsi untuk membantu proses pengambilan keputusan dalam suatu organisasi atau perusahaan. Sistem ini mengolah data menjadi informasi yang dapat dimanfaatkan oleh pengguna dalam menentukan keputusan. Sistem Penunjang Keputusan memungkinkan proses pengambilan keputusan yang lebih cepat dan lebih akurat dengan mengandalkan data objektif daripada sekadar intuisi atau preferensi subjektif. Sistem ini memberikan wawasan serta rekomendasi kepada pengambil keputusan berdasarkan analisis permasalahan, riwayat tindakan yang telah dilakukan, dampak dari tindakan tersebut, serta informasi kontekstual yang relevan (Andoyo et al., 2021).

Sistem Penunjang Keputusan memerlukan metode perhitungan yang akurat untuk menghasilkan keputusan yang optimal. Salah satu metode yang dapat digunakan adalah ARAS, yang merupakan teknik pengambilan keputusan multikriteria berbasis perangkingan. Metode ini menggunakan konsep *utility degree*, di mana setiap alternatif dibandingkan berdasarkan nilai indeks keseluruhannya terhadap nilai indeks dari alternatif terbaik (Damayanti, 2020).

Metode ARAS mengoptimalkan perhitungan dengan menjumlahkan semua kriteria secara maksimal, sehingga hanya dengan satu kali perhitungan, peringkat setiap alternatif dapat diperoleh secara akurat. Keunggulan inilah yang membuat metode ARAS sering digunakan dalam Sistem Penunjang Keputusan, karena mampu menentukan variabel dengan tepat serta menghasilkan peringkat secara efisien dalam satu tahap perhitungan (Saputra & Supriatin, 2022).

Berdasarkan uraian permasalahan di atas maka peneliti bermaksud untuk melakukan penelitian dengan judul "**SISTEM PENUNJANG KEPUTUSAN UNTUK PEMILIHAN GURU TERBAIK MENGGUNAKAN METODE *ADDITIVE RATIO ASSESSMENT (ARAS)* BERBASIS *WEB* (STUDI KASUS: SDI IBNU QOYYIM ILMI)**".

## 1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah diuraikan, diperoleh identifikasi masalah sebagai berikut:

1. Proses pemilihan guru terbaik di SDI Ibnu Qoyyim Ilmi masih dilakukan secara manual, sehingga rentan terhadap subjektivitas penilai dan berpotensi menyebabkan kesalahan input data maupun ketidaksesuaian hasil dengan kinerja guru yang sebenarnya.
2. Ketiadaan sistem pendukung keputusan yang terstruktur dan terotomatisasi menyebabkan proses evaluasi dan seleksi guru terbaik menjadi kurang efisien, memakan waktu, dan tidak dapat menjamin objektivitas hasil keputusan.

## 1.3 Rumusan Masalah

Berdasarkan identifikasi masalah di atas, rumusan masalah yang akan diteliti dalam penelitian ini adalah sebagai berikut:

1. Bagaimana mengembangkan suatu sistem penunjang keputusan berbasis web yang dapat membantu pihak sekolah dalam memilih guru terbaik berdasarkan kriteria seperti kedisiplinan, pengalaman mengajar, tanggung jawab, perilaku, dan komitmen terhadap tugas?
2. Bagaimana metode *Additive Ratio Assessment (ARAS)* dapat diimplementasikan untuk menilai dan membandingkan kinerja guru berdasarkan kriteria yang telah ditentukan?

## 1.4 Batasan Masalah

Dalam penelitian ini, terdapat beberapa batasan yang perlu diperhatikan untuk menjaga fokus dan ruang lingkup penelitian, antara lain:

1. Penelitian ini hanya akan berfokus pada proses pemilihan guru terbaik di SDI Ibnu Qoyyim Ilmi.
2. Penelitian ini akan menggunakan metode *Additive Ratio Assessment (ARAS)* sebagai metode utama dalam pengambilan keputusan multikriteria.
3. Kriteria yang digunakan dalam penilaian guru terbaik meliputi kedisiplinan, pengalaman mengajar, tanggung jawab, perilaku, dan komitmen terhadap tugas.
4. Sistem pendukung keputusan yang dikembangkan dibuat berbasis *web* dan hanya dapat diakses oleh pihak internal sekolah.
5. Data penilaian guru diperoleh dari pihak sekolah melalui observasi, wawancara, atau dokumen resmi yang relevan.

## 1.5 Tujuan Penelitian

Adapun tujuan penelitian yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Mengembangkan suatu sistem penunjang keputusan berbasis *web* yang dapat membantu pihak sekolah dalam memilih guru terbaik berdasarkan kriteria seperti kedisiplinan, pengalaman mengajar, tanggung jawab, perilaku, dan komitmen terhadap tugas.
2. Menerapkan metode *Additive Ratio Assessment (ARAS)* untuk menilai dan membandingkan kinerja guru berdasarkan kriteria yang telah ditentukan.

## 1.6 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat, baik secara teoretis maupun praktis, sebagai berikut:

1. Manfaat Teoretis

Penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan ilmu pengetahuan, khususnya di bidang sistem pendukung keputusan dan penerapan metode *Additive Ratio Assessment (ARAS).* Selain itu, penelitian ini juga dapat menjadi referensi tambahan dalam studi mengenai pengambilan keputusan multikriteria berbasis *web* di lingkungan pendidikan.

1. Manfaat Praktis
   1. Bagi Manajemen Sekolah (SDI Ibnu Qoyyim Ilmi)

Penelitian ini dapat menjadi alat bantu dalam proses pemilihan guru terbaik secara objektif dan terukur berdasarkan kriteria yang telah ditentukan. Sistem yang dikembangkan juga dapat mendukung efisiensi dan transparansi dalam pengambilan keputusan oleh pihak manajemen sekolah.

* 1. Bagi Guru

Penelitian ini dapat mendorong motivasi guru untuk meningkatkan kinerja dan kualitas kerja mereka berdasarkan aspek-aspek penilaian yang dijadikan dasar dalam sistem.

* 1. Bagi Peneliti Lain

Penelitian ini dapat menjadi acuan atau referensi dalam pengembangan sistem pendukung keputusan sejenis, khususnya yang menggunakan metode *ARAS* atau yang ditujukan untuk evaluasi kinerja tenaga pendidik.

* 1. Bagi Universitas

Penelitian ini dapat menjadi referensi dan bahan ajar dalam pengembangan ilmu pengetahuan di bidang sistem penunjang keputusan, khususnya yang menerapkan metode *Additive Ratio Assessment (ARAS)*. Selain itu, hasil penelitian ini dapat menjadi contoh penerapan keilmuan Teknik Informatika dalam menyelesaikan permasalahan nyata di lingkungan pendidikan, sehingga memperkuat kontribusi universitas dalam mendukung penelitian terapan dan pengabdian kepada masyarakat.

## 1.7 Sistematika Penulisan

Skripsi ini disusun dalam beberapa bab, di mana setiap bab berisi penjelasan singkat yang memperjelas proses penelitian. Struktur ini dirancang agar pembahasan lebih sistematis dan fokus pada topik permasalahan. Skripsi ini terdiri dari lima bab, yaitu:

**BAB I PENDAHULUAN**

Bab ini berisi pembahasan mengenai latar belakang masalah, identifikasi masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, serta sistematika penulisan.

**BAB II TINJAUAN PUSTAKA**

Bab ini menjelaskan penelitian yang relevan, landasan teori, perangkat lunak pendukung dan bahasa pemrograman yang digunakan, konsep pengujian sistem, serta kerangka pemikiran yang mendasari penelitian.

**BAB III ANALISA DAN PERANCANGAN**

Bab ini berisi analisis sistem, perancangan basis data, pembuatan diagram *Unified Modeling Language (UML)*, perhitungan menggunakan metode *ARAS*, serta desain antarmuka pengguna *(user interface)*.

**BAB IV IMPLEMENTASI DAN PENGUJIAN**

Bab ini menjelaskan tentang spesifikasi, implementasi program, serta pengujian sistem menggunakan metode *black box* pada sistem yang telah dikembangkan. Hasil penelitian disajikan dalam bentuk teks, gambar, atau tabel data yang telah dianalisis, dengan setiap hasil penelitian disertai pembahasan untuk mencapai kesimpulan.

**BAB V PENUTUP**

Bab penutup berisi kesimpulan dan saran dari hasil penelitian. Kesimpulan menyajikan ringkasan mengenai temuan penelitian dan analisis data yang berkaitan dengan masalah yang dihadapi serta pembuktian hipotesis. Saran memberikan pandangan peneliti mengenai kemungkinan pengembangan dan penerapan hasil penelitian di masa depan. Kesimpulan dan saran disajikan dalam sub bab yang terpisah.

# BAB II LANDASAN TEORI

## 2.1 Penelitian Relevan

1. Penelitian yang dilakukan oleh(Mohamad Alif Wahidin, Siti Ariani, Virgiyanti, M. Erwanto, 2024),dengan judul **“Metode *Additive Ratio Assessment (Aras)* Untuk Menentukan Material Furniture Pada PT. Cirebon Furniture”.** penelitian ini dilakukan untuk membantu PT Cirebon Furniture dalam menentukan material furniture terbaik dengan menggunakan metode *Additive Ratio Assessment (ARAS)*. Metode *ARAS* dipilih karena kemampuannya dalam menilai alternatif berdasarkan berbagai kriteria, termasuk biaya, kualitas, ketersediaan, dan dampak lingkungan. Penelitian ini melibatkan pengumpulan data kriteria secara menyeluruh, normalisasi nilai, dan pembobotan berbasis entropi untuk menghasilkan peringkat material yang paling optimal. Hasil dari implementasi sistem berbasis *ARAS* menunjukkan peningkatan efisiensi dalam proses pengambilan keputusan, dengan rekomendasi yang lebih objektif dan transparan dibandingkan metode manual sebelumnya. Pendekatan ini juga dapat direplikasi oleh produsen lain untuk mendukung pengambilan keputusan berbasis data yang lebih terstruktur​.
2. Penelitian yang dilakukan oleh(Marwa Sulehu dan Liza Handayani, 2019)dengan judul **“Sistem Pendukung Keputusan Pemilihan Guru Terbaik Menggunakan Metode *Additive Ratio Assessment (ARAS)*”.** Penelitian ini mengembangkan Sistem Pendukung Keputusan (SPK) untuk pemilihan guru terbaik di SD Al-Washliyah Medan menggunakan metode *Additive Ratio Assessment (ARAS)*, yang mampu mengolah berbagai kriteria seperti absensi, prestasi, pendidikan terakhir, tingkat kegagalan siswa, dan kepribadian dengan bobot masing-masing, sehingga menghasilkan perankingan guru secara objektif dan efisien; metode *ARAS* mempermudah proses seleksi dengan membandingkan alternatif berdasarkan nilai kriteria yang telah dinormalisasi dan dibobot, sehingga keputusan pemilihan guru terbaik dapat dilakukan lebih cepat dan akurat, memberikan manfaat motivasi bagi guru serta meningkatkan kualitas pendidikan di sekolah tersebut.
3. Penelitian yang dilakukan oleh(Alpin Alpandi, 2020)dengan judul **“Sistem Pendukung Keputusan Penilaian Guru Terbaik Dengan Metode *Additive Ratio Assessment (ARAS)*”.** Penelitian ini membahas pengembangan sistem pendukung keputusan untuk penilaian guru terbaik di SD Negeri 060815 Medan menggunakan metode *Additive Ratio Assessment (ARAS)*, yang bertujuan meningkatkan objektivitas dan akurasi dalam proses evaluasi kinerja guru. Sistem ini menilai guru berdasarkan lima kriteria utama, yaitu kedisiplinan, sikap, prestasi, tanggung jawab, dan pengalaman, dengan bobot yang telah ditentukan untuk masing-masing kriteria, sehingga hasil penilaian lebih terstruktur dan terhindar dari subjektivitas yang sering muncul pada penilaian manual1. Proses *ARAS* meliputi pembentukan matriks keputusan, normalisasi data, pemberian bobot, perhitungan nilai optimalisasi, hingga penentuan peringkat akhir, yang akhirnya memberikan rekomendasi guru terbaik berdasarkan nilai tertinggi dari hasil perhitungan1. Hasil implementasi menunjukkan bahwa sistem ini mampu membantu pihak sekolah dalam memilih guru terbaik secara lebih adil dan transparan, serta dapat dijadikan acuan dalam upaya peningkatan profesionalisme dan kualitas pengajaran di sekolah tersebut.
4. Penelitian yang dilakukan oleh(Hari Setiyani, 2022)dengan judul **“Sistem Penerapan Metode *Additive Ratio Assesment (ARAS)* Pada Sistem Pemilihan Tempat Kursus Bahasa Inggris Online”.** Penelitian ini bertujuan membangun sistem pendukung keputusan (SPK) untuk membantu memilih tempat kursus Bahasa Inggris online terbaik dengan menggunakan metode *ARAS*. Metode *ARAS* dipilih karena mampu melakukan perangkingan alternatif berdasarkan beberapa kriteria, seperti durasi kursus, tingkatan pembelajaran, ketersediaan sertifikat, metode pembelajaran, dan harga. Sistem dikembangkan dengan metodologi *Extreme Programming (XP)* dan diuji menggunakan black box testing, yang hasilnya menunjukkan sistem berfungsi 100% sesuai harapan dan perhitungan otomatisnya akurat seperti perhitungan manual. Studi kasus menunjukkan *ARAS* dapat menentukan alternatif kursus terbaik, dengan hasil *English Academy* sebagai pilihan teratas, diikuti *Golden English*, *Engoo*, dan *Lister*. Sistem ini dapat diimplementasikan untuk membantu orang tua atau calon peserta kursus dalam memilih kursus Bahasa Inggris online secara cepat dan tepat sesuai kebutuhan.
5. Penelitian yang dilakukan oleh( Chrisnatanius Tarigan, Erika Fahmi Ginting, dan Rendy Syahputra, 2022)dengan judul **“Sistem Pendukung Keputusan Dalam Menentukan Kinerja Pengajar Dengan Metode *Additive Ratio Assessment (ARAS)*”.** Penelitian ini bertujuan untuk membantu proses penilaian kinerja tenaga pengajar di SMP Negeri 3 Satu Atap Selesai yang sebelumnya dilakukan secara manual dan cenderung subjektif. Dengan menerapkan metode *ARAS*, sistem pendukung keputusan ini menilai kinerja guru berdasarkan lima kriteria utama, yaitu pedagogik, kepribadian, sosial, profesional, dan tanggung jawab, yang masing-masing memiliki bobot berbeda sesuai tingkat kepentingannya. Proses penilaian dilakukan melalui tahapan pengumpulan data, normalisasi, pembobotan, hingga perangkingan alternatif menggunakan algoritma *ARAS*. Hasil implementasi sistem menunjukkan bahwa metode ini mampu memberikan hasil perangkingan yang objektif dan akurat, memudahkan kepala sekolah dalam menentukan guru terbaik, yang pada penelitian ini diperoleh oleh Novrini, S.Pd. Sistem ini diharapkan dapat meningkatkan kualitas evaluasi kinerja guru dan mendukung pengambilan keputusan yang lebih efektif di lingkungan sekolah.
6. Penelitian yang dilakukan oleh(Agung Deni Wahyudi, 2024)**,** dengan judul **“Analisis Kepuasan Terhadap Pelayanan Supplier Menggunakan Metode *A New Additive Ratio Assessment (ARAS)*”** penelitian ini menggunakan metode *A New Additive Ratio Assessment (ARAS)* untuk menganalisis tingkat kepuasan perusahaan terhadap pelayanan supplier. Fokus penelitian adalah mengevaluasi supplier berdasarkan beberapa kriteria, seperti harga produk, kualitas, waktu pengiriman, ketepatan waktu, responsivitas, dan komunikasi. Data dikumpulkan melalui survei dan analisis historis, lalu diolah menggunakan metode *ARAS* yang memungkinkan pengambilan keputusan secara objektif. Hasil akhirnya adalah peringkat supplier berdasarkan skor kepuasan, dengan supplier "Berkah Sejahtera" menempati peringkat tertinggi. Metode *ARAS* terbukti efektif dalam memberikan penilaian yang sistematis dan terukur, sehingga membantu perusahaan dalam pengelolaan rantai pasok yang lebih efisien.
7. Penelitian yang dilakukan oleh(Syafrida Hafni Sahir dan Siti Apsah Panjaitan, 2019)**,** dengan judul **“Analisis Penerapan Metode *Additive Ratio Assessment (ARAS)* Pada Pemberian Insentif Sales Penjualan Guna Mendukung Keputusan Manajemen”** penelitian ini menganalisis penerapan *metode Additive Ratio Assessment (ARAS)* dalam pemberian insentif bagi tenaga penjualan (sales) untuk mendukung pengambilan keputusan manajemen di sebuah perusahaan. *ARAS* digunakan untuk mengevaluasi kinerja sales berdasarkan berbagai kriteria, seperti jumlah penjualan, tanggung jawab, inisiatif, kejujuran, dan disiplin. Proses analisis melibatkan pembentukan matriks keputusan, normalisasi, pembobotan kriteria, hingga perangkingan alternatif untuk menentukan sales yang layak menerima insentif. Hasil penelitian menunjukkan bahwa metode *ARAS* mampu memberikan hasil yang lebih objektif, akurat, dan dapat dipertanggungjawabkan, sehingga mempermudah manajemen dalam mengalokasikan insentif secara adil dan efisien. Metode ini dinilai efektif untuk meningkatkan motivasi dan kinerja tenaga penjualan, sekaligus menjadi solusi bagi subjektivitas dalam proses pengambilan keputusan insentif.
8. Penelitian yang dilakukan oleh (Darma Saputra Situmeang, Darjat Saripurna, Mhd. Zulfansyuri Siambaton, 2022)**,** dengan judul **“Analisis Penilaian Kualitas Jenis Pelayanan Terbaik dengan Metode *Aras* pada Kantor Dinas Kependudukan dan Pencatatan Sipil”** penelitian ini menganalisis kualitas pelayanan di Kantor Dinas Kependudukan dan Pencatatan Sipil (Disdukcapil) Kabupaten Tapanuli Tengah menggunakan metode *Additive Ratio Assessment (ARAS)*. Tujuan utama penelitian adalah untuk meningkatkan efisiensi dan efektivitas pelayanan kepada masyarakat, dengan fokus pada berbagai keluhan yang diterima terkait layanan yang lambat, prosedur yang rumit, dan kurangnya transparansi. Dari 25 responden yang dinilai, layanan perkawinan muncul sebagai yang terbaik dengan nilai utilitas 0,895. Penelitian ini juga mengidentifikasi enam kriteria penilaian, termasuk persyaratan layanan dan kompetensi pelaksana, serta memberikan rekomendasi untuk peningkatan pelayanan berdasarkan hasil analisis tersebut. Hasil penelitian diharapkan dapat membantu Disdukcapil dalam memahami dan memperbaiki kualitas layanan yang diberikan kepada masyarakat.
9. Penelitian yang dilakukan oleh(Bayu Pangestu, Kosim, Asep Kosasih, 2022)**,** dengan judul **“*Application of Additive Ratio Assessment (ARAS) Method for the Selection of Youth Red Cross Chairperson at* SMA Negeri 1 Lebakwangi Kuningan”** penelitian ini bertujuan untuk membangun sistem pendukung keputusan dalam pemilihan ketua Palang Merah Remaja (PMR) di SMA Negeri 1 Lebakwangi Kuningan menggunakan metode *Additive Ratio Assessment (ARAS).* Metode *ARAS* dipilih karena kemampuannya dalam menangani masalah pengambilan keputusan multi-kriteria dengan membandingkan nilai indeks keseluruhan alternatif. Hasil penelitian menunjukkan bahwa sistem ini dapat membantu panitia pemilihan dalam menilai dan memilih calon ketua PMR secara lebih akurat dan sistematis, mengatasi masalah yang sering muncul pada proses pemungutan suara manual yang dapat menghasilkan data tidak akurat. Dengan implementasi sistem berbasis *web*, proses pemilihan menjadi lebih efisien dan terorganisir, serta memberikan manfaat bagi institusi pendidikan dalam meningkatkan efektivitas pengambilan keputusan.

## 2.2 Tinjauan Pustaka

Berikut adalah tinjauan pustaka yang digunakan oleh penulis untuk mendukung penelitian ini.

### 2.2.1 Definisi Sistem

Menurut (Geograf, 2023), sistem adalah sebuah kesatuan yang tersusun dari elemen-elemen yang saling terhubung dan memengaruhi satu sama lain. Setiap elemen memiliki fungsi khusus yang mendukung pencapaian tujuan bersama. Keterkaitan antar elemen ini membentuk pola kerja yang teratur dan terorganisir.

Menurut (Manurian et al., 2020), sistem merupakan kumpulan prosedur yang saling berkaitan dan membentuk suatu jaringan kerja yang dirancang secara terpadu, dengan tujuan untuk menjalankan aktivitas tertentu atau mencapai sasaran yang telah ditetapkan, khususnya dalam pelaksanaan kegiatan utama suatu organisasi atau perusahaan.

Menurut (Purnama et al., 2020), sistem merupakan sekumpulan elemen yang saling terhubung dan berinteraksi, serta membentuk suatu kesatuan utuh untuk mencapai tujuan tertentu. Sistem memiliki sejumlah karakteristik, antara lain: komponen penyusun, batas sistem, lingkungan eksternal, hubungan antar bagian, input, output, proses pengolahan, dan tujuan yang ingin dicapai.

Berdasarkan beberapa pendapat para ahli, dapat disimpulkan bahwa **sistem merupakan suatu kesatuan yang terdiri dari elemen-elemen atau komponen-komponen yang saling terhubung dan berinteraksi,** membentuk pola kerja yang teratur dan terorganisir untuk mencapai tujuan tertentu. Sistem tidak hanya mencakup komponen internal, tetapi juga mempertimbangkan batas sistem, pengaruh dari lingkungan eksternal, serta adanya alur proses mulai dari input, pengolahan, hingga menghasilkan output. Dengan kata lain, sistem dirancang secara terpadu untuk menjalankan fungsi atau aktivitas tertentu, baik dalam konteks organisasi, perusahaan, maupun bidang lainnya.

### 2.2.2 Aplikasi

Menurut (M. Adlan Al Hawari Nasution, Siswanto, dan Eko Suryana, 2023) aplikasi merupakan suatu perangkat lunak (*software*) atau program komputer yang beroperasi pada sistem yang dibuat serta dikembangkan untuk melakukan perintah tertentu, Istilah aplikasi sendiri diambil dari bahasa Inggris “*Application*” yang dapat diartikan sebagai penerapan atau penggunaan. Secara harfiah, aplikasi merupakan suatu penerapan perangkat lunak atau *software* yang dikembangkan untuk tujuan melakukan tugas-tugas tertentu.

Menurut (Tandon dan Sharma, 2022) aplikasi perangkat lunak merupakan kumpulan program yang dirancang untuk menyelesaikan tugas-tugas spesifik, seperti pengelolaan dokumen, perhitungan data, penyajian informasi, atau sistem layanan tertentu seperti perpustakaan dan pemesanan tiket kereta. Biasanya, setiap aplikasi hanya difokuskan untuk satu fungsi utama dan tidak bisa digunakan untuk fungsi lain. Sebagai contoh, software manajemen perpustakaan tidak dapat difungsikan untuk pemesanan tiket kereta api, begitu pula pengolah kata tidak dimanfaatkan sebagai *spreadsheet*.

### 2.2.3 Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah suatu proses pengeksekusian program yang bertujuan untuk menemukan kesalahan. Pengujian sebaiknya menemukan kesalahan yang tidak disengaja dan pengujian dinyatakan sukses jika berhasil memperbaiki kesalahan tersebut. Selain itu, pengujian juga bertujuan untuk menunjukkan kesesuaian fungsi-fungsi perangkat lunak dengan spesifikasinya. Sebuah perangkat lunak dinyatakan gagal, jika perangkat lunak tersebut tidak memenuhi spesifikasi (Komarudin, 2016).

Secara umum, terdapat dua metode utama dalam pengujian perangkat lunak, yaitu *Black Box Testing* dan *White Box Testing*.

1. *Black Box Testing*

Blackbox-testingmerupakan salah satu metode untuk menguji perangkat lunak yang telah dibangun, baik pengujian pada unit-unit kecil maupun hasil yang telah terintegrasi untuk menguji fungsional perangkat lunak.Pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program untuk mengetahui apakah fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan (Syarif & Pratama, 2021).

Ada beberapa jenis pengujian black box testing, Berikut adalah 10 tipe pengujian dari metode Black Box menurut (Hanifah et al., 2016):

1. *Equivalence Partitioning* : Membagi inputan menjadi kelas data yang dapat digunakan untuk menggenerasi kasus uji.
2. *Boundary Value Analysis / Limit Testing* : Mengijinkan untuk menyeleksi kasus uji yang menguji batasan nilai input, Merupakan komplemen dari *Equivalence Partitioning*.
3. *Comparison Testing* : Uji setiap versi dengan data yang sama untuk memastikan semua versi menghasilkan keluaran yang sama.
4. *Sample Testing* : Melibatkan beberapa nilai yang terpilih dari sebuah kelas ekivalen.
5. *Robustness Testing* : Data input dipilih diluar spesifikasi yang telah didefinisikan, Tujuan dari pengujian ini adalah membuktikan bahwa tidak ada kesalahan jika masukan tidak valid
6. *Behavior Testing* : Hasil uji tidak dapat dievaluasi jika hanya melakukan pengujian sekali, tapi dapat dievaluasi jika pengujian dilakukan beberapa kali, misalnya pada pengujian struktur data stack.
7. *Performance Testing* : Mengevaluasi kemampuan program untuk beroperasi dengan benar dipandang dari sisi acuan kebutuhan misalnya : aliran data, ukuran pemakaian memori, kecepatan eksekusi.
8. *Requirement Testing* : Spesifikasi kebutuhan yang terasosiasi dengan perangkat lunak diidentifikasi pada tahap spesifikasi kebutuhan dan desain.
9. *Endurance Testing* : Melibatkan kasus uji yang diulang-ulang dengan jumlah tertentu.
10. *Cause – Effect Relationship Testing* : Bagi-bagi spesifikasi kebutuhan menjadi bagian yang memiliki kemungkinan kerja.
11. *White Box Testing*

Pengujian *white box* adalah salah satu cara untuk menguji suatu aplikasi atau software dengan cara meneliti dan menganalisa kode dari program yang dibuat ada yang salah atau tidak. Jika model yang sudah dihasilkan berupa output yang tidak sesuai dengan yang diharapkan maka akan dikompilasi ulang dan dicek kembali kode- kode tersebut hingga sesuai dengan yang diharapkan. Adapun langkah penyelesaian *white box* adalah sebagai berikut (Suprapti et al., 2017):

1. Menganalisa sistem berdasarkan alur *flowchart* sistem informasi penjualan.
2. Membuat *flow graph* berdasarkan alur *flowchart*.
3. Menentukan jalur independen berdasarkan gambar *flow graph.*
4. Menghitung kompleksitas siklomatis berdasarkan jalur independen yang dilalui. Untuk menghitung kompleksitas siklomatis ada 3 cara yaitu:
   1. Jumlah region grafik alir sesuai dengan kompleksitas siklomatis
   2. Kompleksitas siklomatis V(G) untuk grafik alir G ditentukan sebagai V(G)=E- N+2 dimana E adalah jumlah edge grafik alir dan N adalah jumlah simpul grafik alir
   3. Kompleksitas siklomatis V(G) untuk grafik alir G ditentukan sebagai V(G)=P+1 dimana P adalah jumlah simpul predikat yang diisikan dalam grafik alir G.
5. Melakukan *test case*.

### 2.2.4 Sistem Penunjang Keputusan

Sistem Penunjang Keputusan (SPK) adalah suatu sistem informasi interaktif yang dirancang untuk memberikan dukungan dalam proses pengambilan keputusan yang kompleks. Dengan menyediakan akses yang mudah terhadap data, model-model analitis, serta kemampuan manipulasi data, SPK memfasilitasi pengambilan keputusan yang lebih baik, terutama dalam situasi yang bersifat semi-terstruktur atau tidak terstruktur. Dalam konteks seperti ini, di mana solusi optimal tidak selalu jelas, SPK berperan sebagai alat bantu yang berharga untuk mengevaluasi berbagai alternatif dan mencapai keputusan yang rasional (Hairani, 2021).

Dalam buku “SISTEM PENDUKUNG KEPUTUSAN Konsep, Implementasi & Pengembangan” menjelaskan Sistem Pendukung Keputusan (DSS) mendorong keputusan yang lebih cepat dan lebih cerdas berdasarkan data objektif, bukan berdasarkan kriteria subjektif atau naluri pribadi. Sistem pendukung keputusan menawarkan wawasan dan tindakan yang diusulkan kepada pembuat keputusan berdasarkan diagnosisi masalah, tindakan sebelumnya yang diambil, hasil dari tindakan tersebut dan info kontekstual relevan lainnya (Andoyo et al., 2021).

SPK merupakan sistem yang membantu, pengambil keputusan dalam proses penyelesaian masalah dengan menyediakan data, model, dan algoritma yang sesuai dengan kebutuhan pengambil keputusan. SPK juga dapat membantu pengambil keputusan dalam menyelesaikan masalah yang kompleks dan tidak terstruktur dengan cara memberikan informasi yang terstruktur dan terorganisir secara logis (Sarwandi et al., 2023).

### 2.2.5 Pendidikan

Pendidikan merupakan proses belajar yang dirancang untuk mengembangkan potensi peserta didik melalui beragam pendekatan, strategi, dan metode. Proses ini dapat melibatkan teknik seperti ceramah, diskusi, maupun eksperimen, yang masing-masing memiliki keunggulan dan keterbatasan dalam mendukung keberhasilan kegiatan pembelajaran secara menyeluruh (Ramdani et al., 2023).

Pendidikan merupakan proses memanusiakan individu agar tumbuh menjadi anggota masyarakat yang bermoral, berpikir kritis, dan memiliki kontribusi positif. Proses ini menekankan pengembangan pengetahuan, keterampilan, dan sikap yang esensial bagi kemajuan diri serta pembangunan bangsa, sebagaimana dibahas dalam kajian tentang ilmu pendidikan (Habsy et al., 2024).

1. Sekolah

Menurut (KBBI, n.d.) sekolah adalah institusi atau tempat yang digunakan untuk kegiatan belajar-mengajar, di mana proses pemberian dan penerimaan pelajaran berlangsung. Berdasarkan tingkatannya, sekolah dapat dibedakan menjadi sekolah dasar, menengah, dan tinggi. Sementara berdasarkan jurusannya, terdapat berbagai jenis seperti sekolah kejuruan, sekolah keguruan, teknik, pertanian, dan lainnya.

1. Guru

Menurut (Diadra Dewina Alti, Uli Makmun Hasibuan, dan Halimah Tusadiah, 2024) tutor(guru) merupakan seseorang yang memberikan dukungan tambahan kepada siswa untuk membantu mereka memahami materi pelajaran. Tutor(guru) juga membimbing dalam penerapan strategi belajar yang efektif, seperti penyusunan jadwal belajar, teknik membaca, pencatatan, hingga metode mengingat, yang secara keseluruhan bertujuan meningkatkan efektivitas proses belajar dan prestasi akademik siswa.

### 2.2.6 Pengertian Metode *Additive Ratio Assessment (ARAS)*

*Additive Ratio Assessment (ARAS)* merupakan metode yang digunakan untuk melakukan perankingan kriteria. Secara umum, metode ini berfokus pada perbandingan antar alternatif guna memperoleh hasil yang optimal. Dalam *ARAS*, perbandingan dilakukan antara jumlah skor kriteria yang telah dinormalisasi dan dibobot, yang mewakili alternatif yang sedang dievaluasi, dengan jumlah skor kriteria yang sama untuk alternatif optimal. Proses ini menghasilkan derajat optimalitas yang mencerminkan sejauh mana alternatif yang dibandingkan mencapai kriteria ideal (Pradana et al., 2024).

Kelebihan metode *ARAS* adalah dapat menghasilkan perankingan dengan membandingkan nilai alternatif pada setiap kriteria untuk menghasilkan alternatif yang paling optimal atau sering disebut dengan alternatif ideal. Kelebihan lainnya bahwa proses penentuan keputusan seringkali kompleks sebab didasari atas argumen pengambilan keputusan sehingga proses penentuan keputusan bisa dipahami secara sederhana dengan menggunakan konsep perbandingan relatif (Sarwandi et al., 2023).

Metode *ARAS* mengoptimalkan perhitungan dengan menjumlahkan semua kriteria secara maksimal, sehingga hanya dengan satu kali perhitungan, peringkat setiap alternatif dapat diperoleh secara akurat (Saputra & Supriatin, 2022).

Dalam melakukan perangkingan, metode *ARAS* memiliki beberapa tahapan penyelesaian perhitungan sebagai berikut:

1. Membuat *Decision Making Matrix (DMM)*

Membuat matriks pengambilan keputusan (X) berdasarkan nilai alternatif (m) terhadap kriteria (n), yaitu:

Dimana:

1. Normalisasi *Decision Making Matrix (DMM)*

Normalisasi *DMM* dimulai dengan menentukan nilai setiap alternatif pada setiap kriteria agar dapat dibandingkan dengan nilai alternatif lainnya. Nilai alternatif dan kriteria yang telah dinormalisasi yaitu overline kemudian disusun untuk memudahkan dalam melakukan normalisasi matriks overline . Terdapat proses pembuatan matriks sebagai berikut:

Dimana:

1. Perhitungan Normalisasi Terbobot

Nilai normalisasi terbobot diperoleh dengan mengalikan nilai matriks normalisasi dengan nilai bobot kriteria . Nilai bobot kriteria sudah ditentukan oleh pengambil keputusan. Perhitungan normalisasi terbobot menggunakan persamaan rumus berikut:

Dimana:

dari kriteria

1. Perhitungan Nilai Fungsi Optimalisasi

Perhitungan fungsi optimalisasi diperoleh dengan menghitung penjumlahan nilai alternatif pada matriks normalisasi terbobot, menggunakan persamaan berikut:

Dimana:

1. Perhitungan Nilai Akhir Perangkingan

Pada tahap akhir perhitungan pada metode ARAS adalah menentukan *K*i pada interval (0,1) yang merupakan nilai akhir peringkat altenatif, perhitungan nilai *K*i dihitung dengan cara melakukan pembagian antara nilai fungsi alternatif (*S*i) dengan fungsi alternatif optimal (*S0*). Perhitungan nilai akhir menggunakan rumus berikut:

Dimana:

(Sarwandi et al., 2023).

### 2.2.7 Aplikasi Berbasis *Web*

*Website* merupakan serangkaian halaman *web* yang dirancang khusus sebagai media informasi terkait kepentingan tertentu, semisal pada instansi, organisasi, bisnis, dan masih banyak lagi. *Web* merupakan sistem server internet yang mendukung dokumen dengan format *HTML (Hyper Text Markup Language)* dan mendukung file dalam bentuk teks, grafik, gambar, video, dan audio. *Website* memiliki beberapa fungsi, diantaranya sebagai media promosi, media pemasaran, media informasi, media Pendidikan, dan media komunikasi. Sebuah *website* selain bersifat informatif juga harus bersifat komunikatif dan interaktif kepada pengguna, hal ini berkaitan dengan kualitas *website* yang dibangun (Hendra & Riti, 2023).

Dalam beberapa dekade terakhir, perkembangan web telah mengalami kemajuan yang sangat signifikan dan memberikan pengaruh besar terhadap berbagai bidang, seperti masyarakat, dunia usaha, pendidikan, pemerintahan, hiburan, industri, hingga kehidupan pribadi. Penggunaan web dalam pengembangan perangkat lunak menawarkan sejumlah keunggulan, seperti penghematan biaya instalasi, pembaruan fitur secara otomatis bagi pengguna, serta kemudahan akses dari berbagai perangkat yang terhubung ke internet. Meski demikian, web juga memiliki kelemahan, terutama karena rawan terhadap kesalahan dan proses pengujiannya yang cukup kompleks, sehingga dibutuhkan strategi pengujian yang tepat dan memadai (Pamungkas & Rochimah, 2019).

## 2.3 UML

UML adalah bahasa untuk menspesifikasi, memvisualisasi membangun dan mendokumentasikan artifacts (bagian dari informasi yang digunakan untuk dihasilkan oleh proses pembuatan perangkat lunak, artifact tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. Selain itu UML adalah bahasa pemodelan yang menggunakan konsep orientasi object. UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera rational software corps. Uml menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai perspektif. Uml tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan (Destriana et al., 2022). Beberapa jenis diagram UML yang umum digunakan dalam proses pengembangan perangkat lunak antara lain sebagai berikut:

### 2.3.1 Use Case Diagram

Use case diagram merupakan hasil dari proses analisis dan perancangan sistem yang bertujuan untuk menggambarkan kebutuhan sistem yang akan diimplementasikan oleh pengguna. Diagram ini menjadi representasi interaksi antara sistem dengan aktor (pengguna), sehingga sangat penting untuk menentukan bentuk abstraksi yang sesuai dengan kebutuhan sistem. Use case diagram menggambarkan bagaimana sistem digunakan melalui skenario interaksi yang umum terjadi antara admin dan user (Ramdany et al., 2024).

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Keterangan** |
|  | Actor | Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan use case. |
|  | Use Case | Abstraksi dan interaksi antara sistem dan aktor. |
|  | Association | Abstraksi dari penghubung antara aktor dan use case. |
|  | Generalization | Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan use case. |
| <<include>> | Include | Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya. |
| <<extend>> | Extend | Menunjukkan bahwa suatu use case merupakan tambahan fungsionalitas dari use case lainnya jika terpenuhi. |

### 2.3.2 Activity Diagram

Activity diagram merepresentasikan alur fungsional dalam suatu sistem informasi. Diagram ini secara menyeluruh menunjukkan di mana alur kerja (workflow) dimulai, berakhir, aktivitas apa saja yang berlangsung selama proses tersebut, serta urutan dari setiap aktivitas. Selain itu, activity diagram juga mendukung pemodelan proses secara paralel. Bagi yang terbiasa dengan pendekatan analisis dan desain terstruktur, diagram ini menggabungkan konsep dasar dari data flow diagram (DFD) dan flowchart (Ramdany et al., 2024).

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Keterangan** |
|  | Status Awal | Sebuah diagram aktivitas memiliki sebuah status awal. |
|  | Aktivitas | Aktivitas yang dilakukan sistem, aktivitas yang biasanya diawali dengan kata kerja. |
|  | Percabangan | Percabangan dimana ada pilihan aktivitas yang lebih dari satu. |
|  | Penggabungan | Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu. |
|  | Status Akhir | Status akhir yang dilakukan sistem sebuah diagram aktivitas memiliki sebuah diagram aktivitas memiliki sebuah status akhir. |
|  | Swimlane | Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |

### 2.3.3 Sequence Diagram

Sequence diagram memvisualisasikan alur pesan (message) yang dikirimkan antar objek seiring waktu dalam sebuah use case. Diagram ini menunjukkan keterlibatan seluruh objek yang terlibat dalam skenario tertentu, serta memperlihatkan urutan interaksi antar objek tersebut. Menurut pendapat lain, sequence diagram juga merepresentasikan bentuk kolaborasi dinamis antara objek-objek dalam sistem, dengan menampilkan bagaimana pesan dikomunikasikan serta bagaimana interaksi berlangsung selama proses tersebut (Narulita et al., 2024).

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Keterangan** |
|  | Entity Class | Gambaran sistem sebagai landasan dalam menyusun basis data. |
|  | Boundary Class | Menangani komunikasi antar lingkungan sistem. |
|  | Control Class | Bertanggung jawab terhadap kelaskelas objek yang berisi logika. |
|  | Recursive | Pesan untuk dirinya. |
|  | Activition | Mewakili proses durasi aktivasi sebuah durasi. |
|  | Life Line | Komponen yang digambarkan garis putus terhubung dengan objek. |

### 2.3.4 Class Diagram

Class diagram merupakan representasi visual dari hubungan antar kelas serta penjabaran detail setiap kelas dalam model desain suatu sistem. Diagram ini menampilkan aturan-aturan dan tanggung jawab masing-masing entitas yang berperan dalam menentukan perilaku sistem. Dengan kata lain, class diagram menggambarkan struktur sistem secara keseluruhan berdasarkan jenis-jenis kelas yang dibentuk. Selain itu, class diagram juga merepresentasikan alur kerja basis data dalam sistem yang akan dikembangkan. Class diagram terdiri dari kumpulan kelas beserta relasinya, di mana setiap class identik dengan entity yang digambarkan dalam bentuk persegi. Bagian atas persegi berisi nama class, diikuti atribut pada bagian tengah, dan metode (operasi) di bagian bawah. Class sendiri merupakan spesifikasi yang ketika diinstansiasi akan menghasilkan objek, dan menjadi inti dalam pengembangan serta desain berbasis objek (Ramdany et al., 2024).

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Keterangan** |
|  | Generalization | Menunjukkan hubungan pewarisan antar class (subclass mewarisi dari superclass). |
|  | Class | Representasi objek yang memiliki atribut dan metode (blok utama OOP). |
|  | Aggregation | Hubungan "bagian dari" yang lemah; objek bagian tetap ada jika objek utama dihapus. |
|  | Dependency | Hubungan ketergantungan sementara antar class. Perubahan class lain bisa memengaruhi. |
|  | Association | Hubungan tetap antar class; menunjukkan koneksi antar objek. |
|  | Composition | Hubungan "bagian dari" yang kuat; objek bagian ikut hilang jika objek utama dihapus. |

## 2.4 Perancangan Basis Data

Pada aplikasi berbasis web dinamis yang banyak digunakan harus menerapkan sistem basis data yang dapat membantu pengelolaan data yang digunakan oleh aplikasi. Penerapan basis data baik itu yang berupa *entity relational database (ERD)* yang sudah dikenal sejak lama, maupun dengan bentuk *object relational database*, ataupun lainnya yang banyak dikembangkan saat ini seperti *NoSQL*, maupun *NewSql* memerlukan perencanaan dan desain yang akurat dalam pengembangannya. Sehingga Ketika basis data diterapkan dalam sistem informasi yang berjalan, maka basis data akan menjadi pendukung sistem tersebut (Pradipta et al., 2022).

### 2.4.2 Basis Data

Basis data dapat diartikan sebagai kumpulan data tentang objek/kejadian yang saling terkait. Di sini, data adalah fakta yang menggambarkan objek, seperti manusia dan hewan, yang dapat direkam dan memiliki makna tersirat. Data direkam dalam bentuk angka, huruf, simbol, gambar, suara, atau kombinasinya. Sebuah sistem basis data berisi kumpulan basis data didalam sistem yang mungkin tidak terkait satu sama lain, tetapi terkait sebagai satu kesatuan sistem yang didukung oleh komponen lainnya. Tujuan utama dari basis data adalah untuk menyediakan lingkungan yang terorganisir dan efisien untuk menyimpan, mengelola, dan mengakses data (Ihksan et al., 2023).

Manfaat menggunakan *database* adalah satu *database* bisa digunakan bersama-sama dalam suatu perusahaan sehingga satu *database* bisa menjadi kontrol atau terpusat, *database* terpusat menyebabkan pengurangan biaya perangkat karena tiap divisi atau bidang dalam suatu perusahaan dapat mengakses *database* dari *server* pusat (Ihksan et al., 2023).

### 2.4.2*Entity Relationship Diagram (ERD)*

*Entity-Relationship Diagram (ERD)* adalah teknik yang digunakan untuk tahap dasar dalam membuat database. *ERD* merupakan salah satu teknik merancang database yang paling banyak digunakan. *ERD* berdasar pada model entity-relationship. Data pada model entity-relationship yang direpresentasikan visual disebut dengan *ERD*. *ERD* ini memrepresentasikan bagaimana entitas saling terkait antara satu dengan yang lainnnya dalam database. Dengan dibuat *ERD* dipercaya dapat membantu perancang dalam menganalisis database yang dibuat. Namun terkadang masih banyak kesalahan yang terjadi saat mendesain *ERD*. Banyak hal-hal yang harus diperhatikan dalam merancang *ERD* pada pembuatan database agar *ERD* yang dibuat sesuai dengan ketentuan (’Afiifah et al., 2022).

Notasi standar yang umum digunakan dalam *ERD* antara lain (Aditiyawarman, 2016):

1. *Entity*   
   *Entity* adalah suatu objek yang dapat diidentifikasi yang terdapat dalam suatu sistem yang akan dibuatkan basis datanya. Suatu objek harus dapat dibedakan dengan objek lainnya. Objek bisa berupa orang, bagian, benda atau konsep, bisa abstrak (tidak ada wujudnya) dan bisa saja fisik (ada wujudnya). Entitas digambarkan dalam bentuk persegi panjang.
2. *Attribute*   
   Menyatakan elemen data atau karakteristik yang dimiliki oleh entitas. Digambarkan dengan bentuk elips.
3. *Relationship*   
   Menyatakan hubungan antara satu entitas dengan entitas lainnya. Simbol relationship digambarkan dalam bentuk diamond atau belah ketupat.
4. *Cardinality*  
   Menyatakan jumlah maksimum entitas pada himpunan entitas yang berelasi dengan entitas pada himpunan entitas yang lain.

Jenis hubungan yang teradi antara satu entitas dengan entitas yang lainnya dalam basis data terdiri dari (Aditiyawarman, 2016):

1. Satu ke satu

Setiap entitas pada himpunan entitas A berhubungan dengan paling banyak satu entitas pada himpunan entitas B, dan sebaliknya setiap entitas pada himpunan entitas B berhubungan dengan paling banyak satu entitas pada himpunan entitas A.

1. Satu ke banyak

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, dan tidak sebaliknya dimana setiap entitas pada himpunan entitas B hanya dapat berhubungan dengan paling banyak satu entitas pada himpunan entitas A.

1. Banyak ke satu

Setiap entitas pada himpunan entitas A hanya dapat berhubungan dengan paling banyak satu entitas pada himpunan entitas B, dan tidak sebaliknya dimana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A.

1. Banyak ke banyak

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, dan sebaliknya setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A.

### 2.4.3 *Logical Record Structure (LRS)*

*Logical Record Structure (LRS)* adalah penggambaran skema basis data yang memberikan detail lebih rinci mengenai hubungan antara entitas dalam perancangan basis data. *LRS* merupakan representasi dari struktur record-record pada tabel-tabel yang terbentuk dari hasil relasi antar himpunan entitas.

*LRS* dapat dibentuk berdasarkan diagram *Entity-Relationship Diagram (ERD)* yang telah dibuat sebelumnya. Proses konversi *ERD* ke *LRS* bertujuan untuk membentuk data-data dari hubungan entitas ke dalam model berbasis *record*. Setelah *ERD* dikonversi ke bentuk LRS, hasil akhirnya adalah sebuah diagram yang mampu menggambarkan basis data yang akan digunakan dalam sistem.

Struktur LRS terdiri dari tipe *record*, yang direpresentasikan dalam bentuk persegi dan berisi atribut atau *field* yang diperlukan. Selain itu, *LRS* juga menunjukkan hubungan antar *record* dalam basis data. Dari diagram *LRS* ini, rancangan tabel relasi dapat dibuat, yang kemudian dapat melalui proses normalisasi (jika diperlukan) untuk menghasilkan rancangan tabel relasi yang lebih optimal dan siap digunakan dalam sistem.

### 2.4.4 Normalisasi

Normalisasi adalah suatu teknik untuk menghasilkan sekumpulan relasi/tabel yang memiliki karakteristik tertentu, untuk memenuhi kebutuhan organisasi. Proses normalisasi diperlukan dalam membentuk tabel-tabel yang normal (Khomsi Pane et al., 2022).

1. Bentuk Normal Tahap 1 (1NF)

Bentuk normal yang pertama atau 1NF mensyaratkan beberapa kondisi dalam sebuah *database*, berikut adalah fungsi dari bentuk normal pertama ini.

1. Menghilangkan duplikasi kolom dari tabel yang sama.
2. Buat tabel terpisah untuk masing-masing kelompok data terkait dan mengidentifikasi setiap baris dengan kolom yang unik (*primary key*).
3. Bentuk Normal Tahap 2 (2NF)

Syarat untuk menerapkan normalisasi bentuk kedua ini adalah data telah dibentuk dalam 1NF, berikut adalah beberapa fungsi normalisasi 2NF.

1. Menghapus beberapa *subset* data yang ada pada tabel dan menempatkan mereka pada tabel terpisah.
2. Menciptakan hubungan antara tabel baru dan tabel lama dengan menciptakan *foreign key.*
3. Tidak ada atribut dalam tabel yang secara fungsional bergantung pada *candidate key* tabel tersebut.
4. Bentuk Normal Tahap 3 (3NF)

Normalisasi *database* dalam bentuk 3NF bertujuan untuk menghilangkan seluruh atribut atau *field* yang tidak berhubungan dengan *primary key*. Dengan demikian tidak ada ketergantungan transitif pada setiap kandidat *key*. Syarat dari bentuk normal ketiga atau 3NF adalah;

1. Memenuhi semua persyaratan dari bentuk normal kedua.

Menghapus kolom yang tidak tergantung pada primary key.

## 2.5 Aplikasi Pendukung

Dalam proses pembangunan sistem penunjang keputusan berbasis web, diperlukan berbagai aplikasi dan teknologi untuk menunjang pengembangan dan implementasi sistem secara optimal. Setiap perangkat lunak maupun bahasa pemrograman yang digunakan memiliki peran penting dalam membentuk arsitektur, antarmuka, hingga fungsionalitas sistem. Oleh karena itu, pada sub-bab ini akan dibahas beberapa aplikasi dan teknologi utama yang digunakan dalam penelitian ini, yaitu XAMPP sebagai server lokal, HTML dan CSS sebagai dasar tampilan antarmuka, PHP sebagai bahasa pemrograman sisi server, Visual Studio Code sebagai editor kode utama, serta Bootstrap sebagai framework untuk mendesain antarmuka yang responsif dan modern.

### 2.5.1 XAMPP

*XAMPP* adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*),yang terdiri atas program *Apache HTTP Server, MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Nama *XAMPP* merupakan singkatan dari X (empat sistem operasi apapun), *Apache, MySQL,PHP dan Perl*. Program ini tersedia dalam *GNU General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis (Siregar et al., 2021).

### 2.5.2 HTML

HTML adalah bahasa markah standar yang digunakan untuk membuat dan menyusun konten di World Wide Web. Bahasa ini menyediakan serangkaian elemen atau tag yang menentukan bagian-bagian berbeda dari sebuah dokumen web. Berikut penjelasan rinci mengenai HTML dan pentingnya dalam menyusun dokumen web (Joshi et al., 2022):

1. Struktur Dokumen

HTML menetapkan struktur keseluruhan sebuah dokumen melalui tag pembuka <html> dan penutup </html>. Di dalam struktur ini terdapat dua bagian utama, yaitu <head> (berisi informasi meta, referensi gaya, dan sebagainya) serta <body> (berisi konten yang ditampilkan).

1. Penandaan Teks

HTML menggunakan tag-tag tertentu untuk menandai dan menyusun konten teks. Tag umum antara lain <p> untuk paragraf, <h1> hingga <h6> untuk judul, <strong> untuk teks tebal, <em> untuk penekanan, dan <a> untuk tautan.

1. Daftar dan Tabel

HTML mendukung daftar terurut (<ol>), daftar tak terurut (<ul>), dan daftar definisi (<dl>) untuk mengorganisasi informasi. Elemen <table> digunakan untuk menampilkan data dalam bentuk baris dan kolom.

1. Formulir dan Elemen Input

Formulir (<form>) merupakan komponen utama dalam interaksi pengguna. HTML menyediakan berbagai elemen input seperti kotak teks (<input type="text">), kotak centang (<input type="checkbox">), tombol radio (<input type="radio">), dan lainnya.

1. Gambar dan Multimedia

Gambar dapat disisipkan menggunakan tag <img>. HTML5 juga memperkenalkan tag <video> dan <audio> untuk memudahkan integrasi konten multimedia secara langsung di halaman web.

### 2.5.3 CSS

CSS (Cascading Style Sheets) merupakan bahasa pemrograman yang digunakan untuk mengatur tampilan visual dari dokumen web berbasis HTML maupun XML. Perannya sangat krusial dalam mengontrol estetika halaman web, mulai dari warna, ukuran, tata letak, hingga posisi elemen pada layar. Berikut adalah penjelasan detail mengenai komponen-komponen utama dalam CSS (Joshi et al., 2022):

1. Selector

Salah satu komponen utama dalam CSS adalah selector, yang berfungsi untuk memilih elemen HTML tertentu yang ingin diberikan gaya. Selector ini memungkinkan kontrol yang fleksibel terhadap tampilan melalui identifikasi elemen berdasarkan tag, class, ID, atau struktur kombinasi yang kompleks.

1. Cascade

Konsep cascading menjelaskan bagaimana prioritas aturan styling diterapkan, khususnya saat terdapat beberapa aturan yang saling bertentangan. CSS akan menentukan gaya yang berlaku berdasarkan spesifisitas, pewarisan dari elemen induk, dan urutan penulisan.

1. Box Model

CSS juga menerapkan konsep box model, yang mengatur setiap elemen dalam struktur konten, meliputi konten utama, padding (jarak dalam), border (batas tepi), dan margin (jarak luar). Hal ini sangat penting dalam penyusunan layout yang presisi dan teratur.

1. Layout

Terakhir, fitur seperti Flexbox dan CSS Grid memberikan fleksibilitas tinggi dalam pengaturan tata letak halaman web, baik untuk tampilan sederhana maupun desain kompleks yang responsif terhadap berbagai perangkat.

Dengan demikian, CSS merupakan bagian integral dari pengembangan web modern karena memungkinkan desain yang menarik, terstruktur, dan mendukung kenyamanan pengguna secara visual.

### 2.5.4 PHP

**PHP** adalah salah satu bahasa pemrograman berbasis web yang dirancang khusus untuk membangun aplikasi dinamis di internet. Bahasa ini pertama kali dikembangkan oleh **Rasmus Lerdorf**, seorang anggota tim pengembang perangkat lunak Apache, dan dirilis pada akhir tahun 1994. Awalnya, PHP hanya digunakan untuk mencatat jumlah pengunjung di situs web pribadi miliknya. Seiring waktu, PHP berkembang menjadi bahasa pemrograman yang populer karena bersifat **open-source**, dapat digunakan secara **gratis**, dan memiliki **kemudahan dalam proses pembelajaran,** sehingga dapat diakses oleh berbagai kalangan, baik pemula maupun profesional dalam pengembangan web (Hidayat et al., 2019).

### 2.5.5 Visual Studio Code

*Visual Studio (VS) Code* merupakan *Integrated Development Environment (IDE)* yang ringan, dikembangkan dan didukung oleh *Microsoft*, serta tersedia secara gratis untuk penggunaan pribadi maupun komersial. Fitur utama dari *VS Code* adalah dukungan ekstensi, yang memungkinkan pengguna menambahkan bahasa pemrograman, *debugger*, dan berbagai alat lainnya ke dalam instalasi mereka guna mendukung proses pengembangan lebih lanjut. Selain ekstensi standar yang dirilis oleh *Microsoft*, terdapat banyak ekstensi lainnya yang tersedia di *VS Code Extension Marketplace*, yang dikembangkan oleh organisasi pihak ketiga maupun individu (Tan et al., 2024).

### 2.5.6 Bootstrap

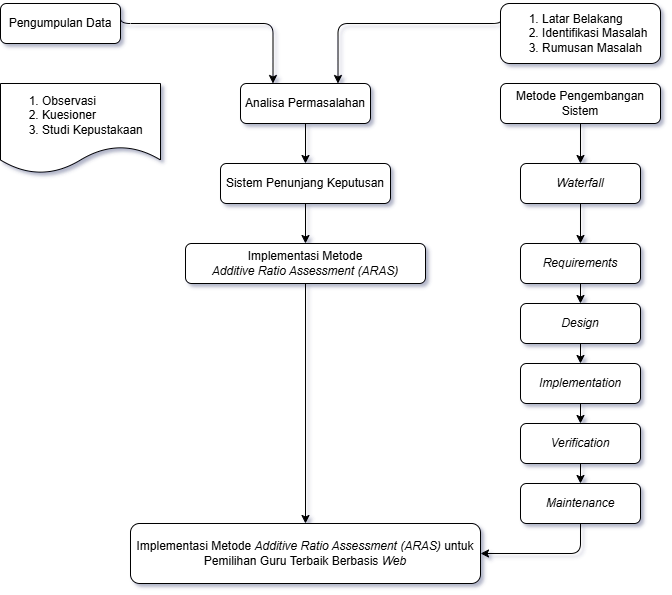
Bootstrap merupakan salah satu framework front-end berbasis HTML, CSS, dan JavaScript yang banyak digunakan dalam pengembangan aplikasi web yang responsif dan mendukung tampilan seluler. Framework ini bersifat open-source dan dapat digunakan secara gratis, sehingga banyak dimanfaatkan oleh pengembang untuk mempercepat pembuatan antarmuka pengguna. Bootstrap menyediakan berbagai template desain siap pakai seperti elemen tipografi, tombol, formulir, tabel, navigasi, modals, hingga carousel gambar yang memudahkan proses pembangunan tampilan web yang modern dan dinamis (Shahu Gaikwad & Adkar, 2019).

### 2.5.7 Draw.io

Draw.io merupakan perangkat lunak yang digunakan untuk mensimulasikan jaringan komputer. Aplikasi ini berfungsi untuk merancang dan memvisualisasikan bentuk jaringan komputer secara virtual. Selain mendukung simulasi, Draw.io juga menyediakan fitur pengaturan tata letak (layout) dan kolaborasi, serta memudahkan dalam pemahaman konsep teknologi yang kompleks. Perangkat lunak ini dilengkapi dengan berbagai elemen yang umum digunakan dalam struktur jaringan, sehingga memudahkan pengguna dalam membuat simulasi jaringan komputer di perangkat mereka. Melalui simulasi ini, pengguna dapat memahami cara kerja tiap komponen jaringan serta proses pengiriman data dari satu komputer ke komputer lainnya (Hendrawan et al., 2023).

## 2.6 Kerangka Pemikiran

Berikut adalah kerangka pemikiran dalam penelitian yang berjudul “Sistem Penunjang Keputusan untuk Pemilihan Guru Terbaik Menggunakan Metode *Additive Ratio Assessment (ARAS)* Berbasis *Web* (Studi Kasus: SDI Ibnu Qoyyim Ilmi)”ini:



# BAB III ANALISA DAN PERANCANGAN SISTEM

## 3.1 Analisa Sistem

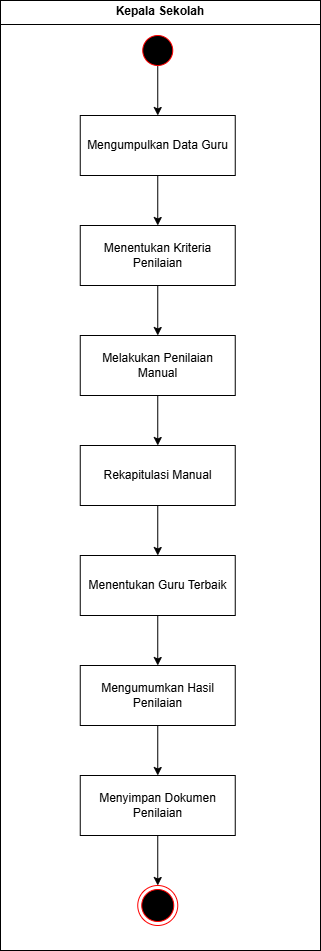
Analisa sistem merupakan tahap awal dalam proses pengembangan sistem yang bertujuan untuk memahami kondisi sistem saat ini dan menentukan kebutuhan sistem yang akan dibangun. Tahapan ini penting dilakukan agar perancangan sistem dapat dilakukan secara tepat sasaran dan sesuai dengan permasalahan yang ada di lapangan. Dalam subbab ini, akan dijelaskan mengenai analisa sistem yang sedang berjalan di SDI Ibnu Qoyyim Ilmi serta sistem usulan yang akan dikembangkan untuk mendukung proses pemilihan guru terbaik. Selain itu, akan dijelaskan pula penerapan perhitungan menggunakan metode Additive Ratio Assessment (ARAS) sebagai metode yang digunakan dalam sistem pendukung keputusan yang dirancang.

### 3.1.1 Analisa Sistem Berjalan

Dalam pelaksanaan sistem berjalan, proses pemilihan guru terbaik di SDI Ibnu Qoyyim Ilmi dilakukan secara manual oleh kepala sekolah. Langkah pertama dimulai dengan pengumpulan data dari setiap guru, meliputi data kehadiran, catatan kedisiplinan, partisipasi dalam kegiatan sekolah, serta hasil laporan pembelajaran. Data ini diperoleh melalui absensi manual, laporan kegiatan, serta pengamatan langsung terhadap aktivitas guru.

Setelah data terkumpul, langkah selanjutnya adalah menentukan kriteria penilaian yang akan digunakan. Kriteria yang dijadikan acuan terdiri dari lima aspek, yaitu kehadiran, disiplin kerja, kompetensi pedagogik, tanggung jawab administratif, dan keterlibatan dalam kegiatan sekolah. Masing-masing guru kemudian dinilai berdasarkan kriteria tersebut menggunakan skala penilaian dari 1 sampai 5. Penilaian ini bersifat subjektif dan dicatat secara manual menggunakan lembar penilaian kertas atau file Excel sederhana.

Selanjutnya, nilai dari tiap guru direkap dan proses perhitungan dilakukan secara manual untuk memperoleh total skor keseluruhan. Guru yang memperoleh nilai tertinggi kemudian dipilih sebagai guru terbaik. Hasil dari proses ini diumumkan secara internal, biasanya melalui rapat atau kegiatan sekolah. Setelah itu, seluruh dokumen penilaian disimpan dalam bentuk fisik, tanpa adanya sistem basis data terpusat atau dokumentasi digital. Hal ini membuat proses pencatatan dan evaluasi menjadi kurang efisien serta sulit untuk ditelusuri kembali dalam jangka panjang.

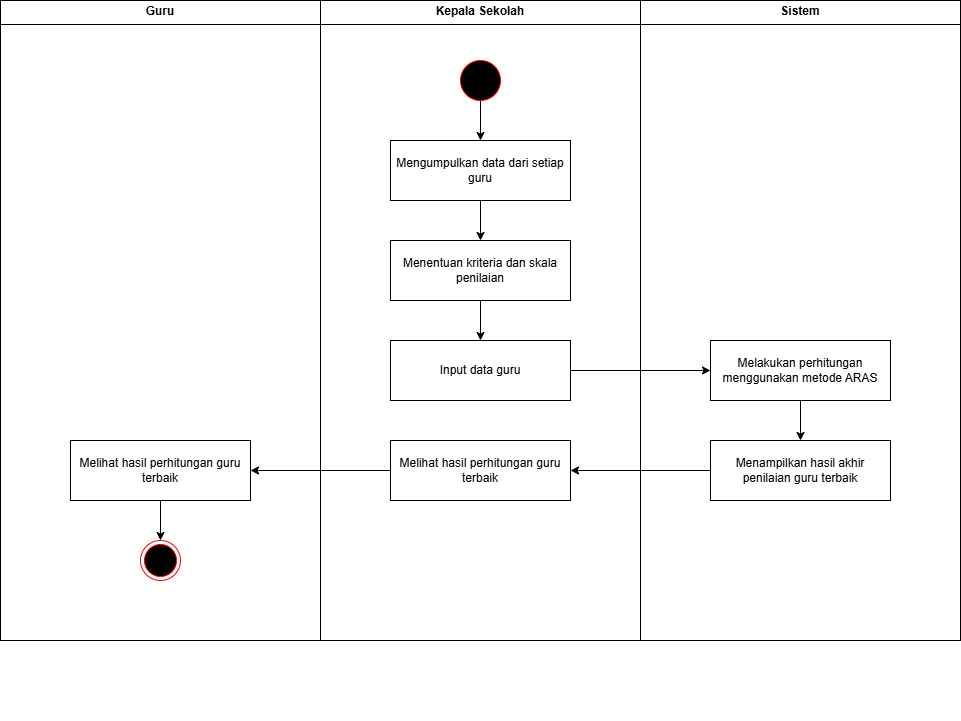


Namun, metode penilaian manual ini memiliki beberapa kekurangan, seperti potensi subjektivitas, rawan kesalahan perhitungan, dan kurang efisien dalam pengumpulan serta pengolahan data. Selain itu, tidak adanya sistem terintegrasi membuat proses evaluasi sulit untuk dilakukan secara konsisten dan terdokumentasi dengan baik. Oleh karena itu, diperlukan sebuah sistem pendukung keputusan yang dapat membantu dalam proses pemilihan guru terbaik secara objektif, akurat, dan efisien.

### 3.1.2 Analisa Sistem Usulan

Sistem usulan yang akan dikembangkan bertujuan untuk menggantikan proses manual dalam pemilihan guru terbaik dengan sistem berbasis web yang lebih terstruktur dan efisien. Sistem ini dirancang untuk mendukung pengambilan keputusan secara objektif dengan memanfaatkan metode *Additive Ratio Assessment* (ARAS) sebagai dasar perhitungan. Pengguna utama dalam sistem ini adalah admin (sebagai operator sistem), kepala sekolah, dan guru sebagai entitas yang dinilai.

Melalui sistem ini, data penilaian guru akan diinput ke dalam aplikasi berdasarkan kriteria-kriteria tertentu yang telah ditentukan sebelumnya, seperti kehadiran, disiplin kerja, kompetensi pedagogik, tanggung jawab administratif, dan keterlibatan dalam kegiatan sekolah. Setiap kriteria memiliki bobot tertentu sesuai dengan tingkat kepentingannya, dan sistem akan melakukan perhitungan otomatis untuk menghasilkan nilai akhir setiap guru. Nilai tersebut kemudian diurutkan untuk menampilkan peringkat guru terbaik.



Dengan adanya sistem ini, proses penilaian menjadi lebih transparan, akurat, dan terdokumentasi dengan baik dalam basis data. Selain itu, sistem juga meminimalkan kesalahan perhitungan, mempercepat proses evaluasi, serta memudahkan pihak sekolah dalam melakukan analisis performa guru secara berkala.

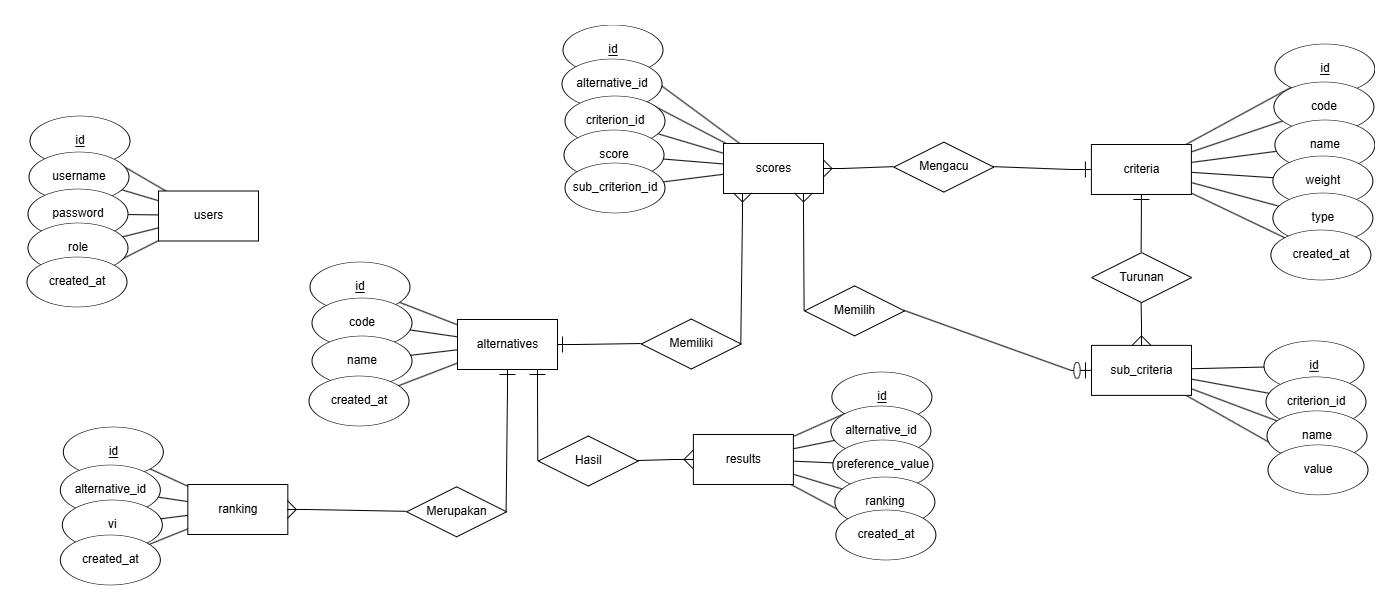
### 3.1.3 Penerapan Perhitungan Motode ARAS(BLM DAPET DATA)

## 3.2 Perancangan Basis Data

Perancangan basis data merupakan tahap yang sangat penting dalam pembangunan sistem informasi karena menyangkut bagaimana data disimpan, dikelola, dan diakses secara efisien. Dalam sistem pendukung keputusan pemilihan guru terbaik berbasis web ini, perancangan basis data dilakukan agar seluruh proses pengolahan data dapat dilakukan secara sistematis dan terintegrasi. Perancangan basis data meliputi penyusunan Entity Relationship Diagram (ERD), transformasi ERD ke bentuk Logical Record Structure (LRS), proses normalisasi, hingga spesifikasi basis data secara keseluruhan. Tujuan dari perancangan ini adalah untuk menghasilkan struktur data yang optimal, minim redundansi, dan mudah dikembangkan.

### 3.2.1 Entity Relationship Diagram (ERD)

Pada perancangan basis data Sistem Pendukung Keputusan Berbasis Web Untuk Pemilihan Guru Terbaik di SDI Ibnu Qoyyim Ilmi Menggunakan Metode Additive Ratio Assessment (ARAS) ini terdapat 7 tabel utama, yaitu users, alternatives, criteria, sub\_criteria, scores, ranking, dan results. Tabel-tabel ini saling berelasi satu sama lain untuk membentuk sistem yang terintegrasi dalam proses penilaian guru. Adapun gambaran relasi antar entitas dalam bentuk Entity Relationship Diagram (ERD) adalah sebagai berikut:

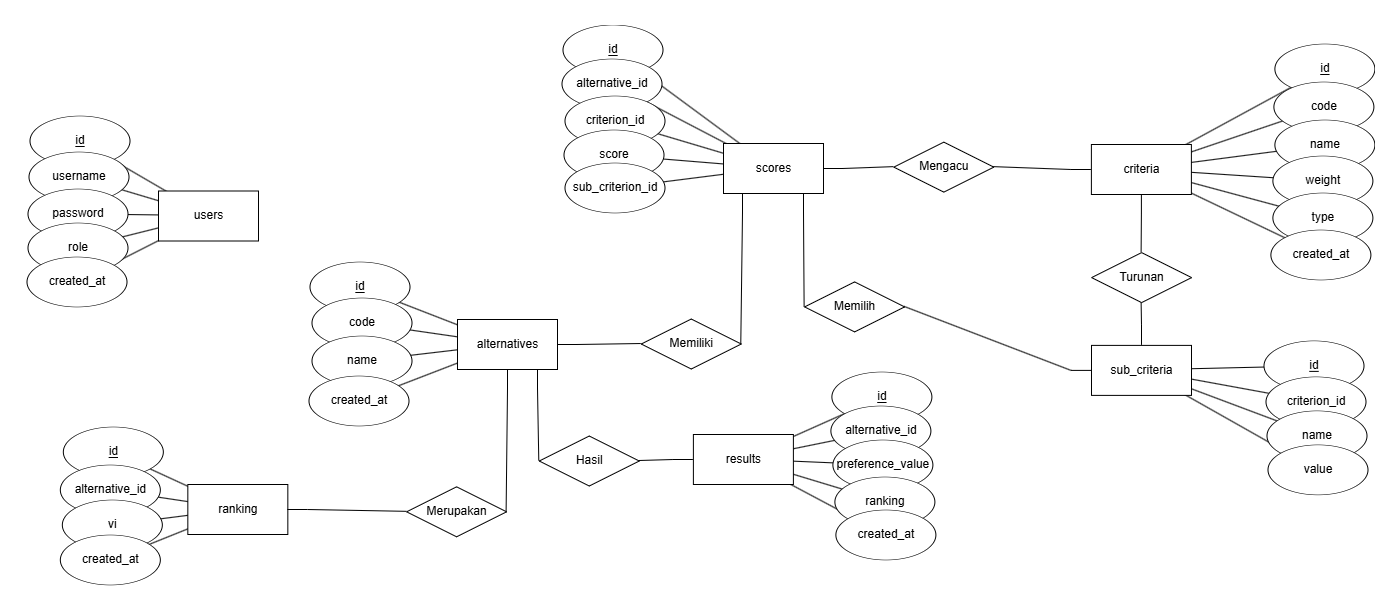


Pada ERD tersebut, peneliti merancang dan memiliki 7 entitas utama yang saling berelasi dalam mendukung proses pemilihan guru terbaik menggunakan metode ARAS. Entitas users berfungsi sebagai data pengguna yang memiliki hak akses ke dalam sistem, seperti admin atau kepala sekolah, yang nantinya melakukan proses input dan manajemen data. Entitas criteria digunakan untuk menyimpan daftar kriteria penilaian yang menjadi dasar dalam evaluasi guru, misalnya kehadiran, kedisiplinan, dan kompetensi pedagogik.

Kemudian terdapat entitas alternatives yang merepresentasikan daftar guru yang menjadi objek penilaian dalam sistem. Masing-masing guru akan dievaluasi berdasarkan kriteria yang telah ditentukan. Dalam satu kriteria, dapat terdiri dari beberapa tingkatan nilai, yang dikelola oleh entitas sub\_criteria. Setiap subkriteria memiliki nilai tertentu yang merepresentasikan bobot atau level terhadap suatu kriteria.

Selanjutnya, entitas scores digunakan untuk menyimpan nilai hasil penilaian masing-masing guru terhadap tiap kriteria dan subkriteria yang dipilih. Nilai-nilai ini kemudian diolah melalui proses perhitungan metode ARAS dan hasil perhitungannya disimpan di entitas ranking, yang berisi nilai Vi dari masing-masing alternatif. Terakhir, entitas results digunakan untuk menyimpan hasil akhir berupa nilai preferensi dan peringkat guru terbaik berdasarkan perhitungan tersebut. Dengan perancangan seperti ini, proses penilaian menjadi lebih terstruktur, transparan, dan dapat dipertanggungjawabkan secara sistematis.

### 3.2.2 Transformasi ERD ke Logical Record Structure (LRS)

Transformasi Entity Relationship Diagram (ERD) ke dalam bentuk Logical Record Structure (LRS) bertujuan untuk menggambarkan struktur logis dari tabel-tabel yang ada dalam sistem beserta atribut-atributnya secara lebih rinci. Proses ini menjadi langkah awal dalam menentukan desain fisik basis data yang akan digunakan dalam implementasi sistem.

1

1

M

M

0..1

M

M

1

M

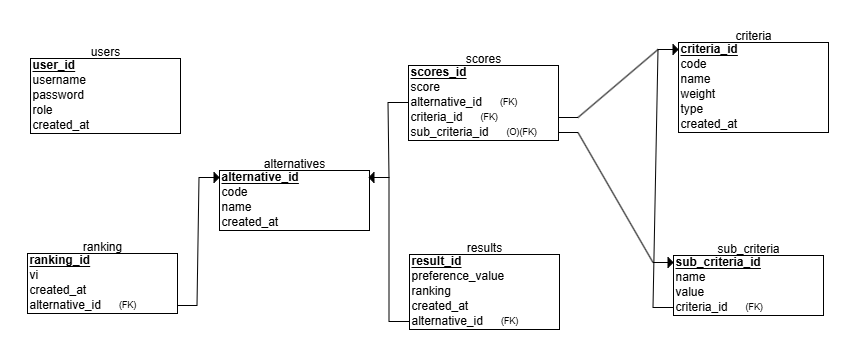
1

M

1

### 3.2.3 Logical Record Structure (LRS)

Logical Record Structure (LRS) merupakan representasi struktur tabel-tabel dalam basis data secara logis yang dihasilkan dari transformasi Entity Relationship Diagram (ERD). LRS berfungsi untuk menjelaskan secara rinci isi dari setiap record atau tabel, atribut-atribut yang dimiliki, tipe data, serta keterhubungan antar tabel melalui primary key dan foreign key. Struktur ini memudahkan dalam proses implementasi sistem karena telah menggambarkan susunan data yang akan digunakan dan diolah di dalam sistem pendukung keputusan berbasis web. Adapun struktur logis dari masing-masing tabel yang digunakan dalam sistem ini adalah sebagai berikut:



### 3.2.4 Normalisasi(SKIP)

### 3.2.5 Spesifikasi Basis Data

Spesifikasi basis data berfungsi untuk menjelaskan secara teknis mengenai struktur tabel yang digunakan dalam sistem. Setiap tabel dijelaskan mulai dari nama tabel, nama atribut, tipe data, panjang karakter (jika ada), serta kunci utama dan kunci relasi (foreign key) jika terdapat hubungan antar tabel. Spesifikasi ini digunakan sebagai acuan dalam implementasi database pada sistem berbasis web yang dibangun. Berikut adalah spesifikasi dari masing-masing tabel yang digunakan dalam sistem:

1. User

Nama Tabel : users

Isi : Data pengguna sistem

Primary Key : id

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Keterangan** |
| id | Int | 11 | Menyimpan Auto Increment id |
| username | Varchar | 100 | Username admin |
| password | Varchar | 255 | Password admin |
| role | Enum | - | Peran admin (default: admin) |
| created\_at | Timestamp | - | Tanggal dibuat akun |

1. Alternative

Nama Tabel : alternatives

Isi : Data guru sebagai alternatif

Primary Key : id

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Keterangan** |
| id | Int | 11 | Menyimpan Auto Increment id |
| code | Varchar | 20 | Kode alternatif (misal: A1) |
| name | Varchar | 100 | Nama guru |
| created\_at | Timestamp | - | Tanggal input data guru |

1. Kriteria

Nama Tabel : criteria

Isi : Data kriteria penilaian

Primary Key : id

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Keterangan** |
| id | Int | 11 | Menyimpan Auto Increment id |
| code | Varchar | 10 | Kode kriteria (misal: C1) |
| name | Varchar | 100 | Nama kriteria (misal: Disiplin) |
| weight | Decimal | 5,2 | Bobot kriteria |
| type | Enum | - | Jenis kriteria: benefit atau cost |
| created\_at | Timestamp | - | Tanggal input data |

1. Sub Kriteria

Nama Tabel : sub\_criteria

Isi : Nilai level dari setiap kriteria

Primary Key : id

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Keterangan** |
| id | Int | 11 | Menyimpan Auto Increment id |
| criterion\_id | Int | 11 | Relasi ke criteria.id |
| name | Varchar | 100 | Nama subkriteria |
| value | Decimal | 5,2 | Nilai subkriteria |

1. Score

Nama Tabel : scores

Isi : Data penilaian setiap guru

Primary Key : id

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Keterangan** |
| id | Int | 11 | Menyimpan Auto Increment id |
| alternative\_id | Int | 11 | Relasi ke alternatives.id |
| criterion\_id | Int | 11 | Relasi ke criteria.id |
| score | Decimal | 10,2 | Skor hasil penilaian |
| sub\_criterion\_id | Int | 11 | Relasi ke sub\_criteria.id (boleh null) |

1. Ranking

Nama Tabel : users

Isi : Data pengguna sistem

Primary Key : id

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Keterangan** |
| id | Int | 11 | Menyimpan Auto Increment id |
| alternative\_id | Int | 11 | Relasi ke alternatives.id |
| vi | Decimal | 10,4 | Nilai hasil akhir (Vi) metode ARAS |
| created\_at | Timestamp | - | Waktu perhitungan dilakukan |

1. Result

Nama Tabel : users

Isi : Data pengguna sistem

Primary Key : id

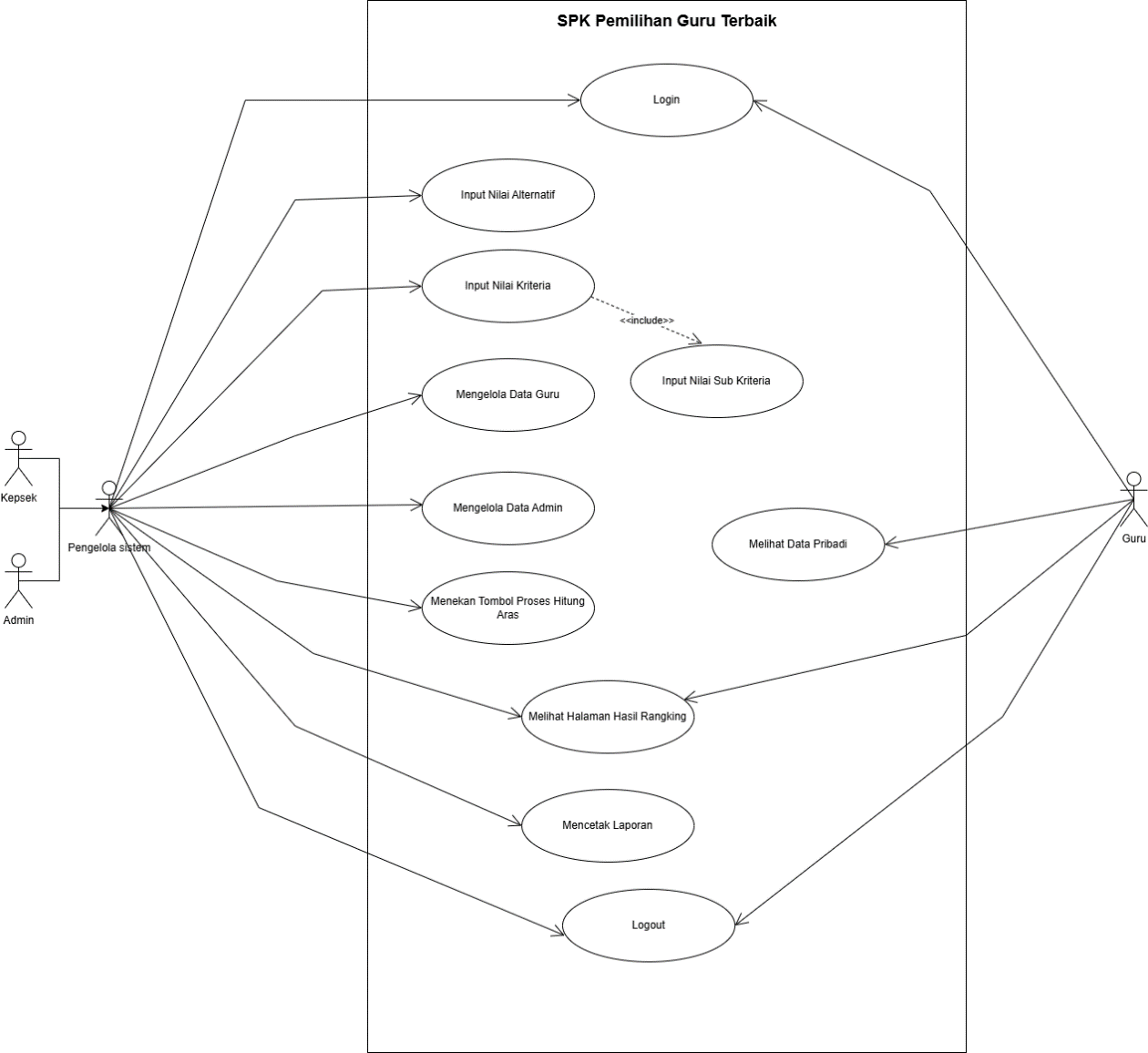
|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Keterangan** |
| id | Int | 11 | Menyimpan Auto Increment id |
| alternative\_id | Int | 11 | Relasi ke alternatives.id |
| preference\_value | Decimal | 10,4 | Nilai preferensi akhir |
| ranking | Int | 11 | Peringkat hasil pemilihan guru |
| created\_at | Timestamp | - | Waktu penyimpanan hasil |

## 3.3 Perancangan Unified Modeling Language (UML)

Perancangan sistem menggunakan Unified Modeling Language (UML) bertujuan untuk menggambarkan spesifikasi, visualisasi, serta dokumentasi sistem yang akan dibangun secara terstruktur dan sistematis. UML memudahkan pengembang maupun stakeholder dalam memahami alur kerja sistem, interaksi antar aktor, serta hubungan antar komponen di dalam sistem. Dalam proyek ini, perancangan UML mencakup beberapa diagram, yaitu Use Case Diagram, Activity Diagram, Sequence Diagram, dan Class Diagram yang masing-masing merepresentasikan sudut pandang berbeda dari sistem pendukung keputusan pemilihan guru terbaik menggunakan metode ARAS berbasis web.

### 3.3.1 Use Case Diagram

Use Case Diagram digunakan untuk menggambarkan interaksi antara pengguna (aktor) dengan sistem yang akan dibangun. Diagram ini merepresentasikan fungsi-fungsi utama yang dapat dijalankan oleh aktor terhadap sistem, serta batasan-batasan dari sistem itu sendiri. Dalam sistem pendukung keputusan pemilihan guru terbaik menggunakan metode ARAS berbasis web ini, terdapat tiga aktor utama, yaitu **admin**, **kepala sekolah**, dan **guru**. Admin dan kepala sekolah bertindak sebagai pengelola sistem yang memiliki akses terhadap pengelolaan data pengguna, kriteria, subkriteria, alternatif, serta proses perhitungan dan hasil pemilihan guru terbaik. Sementara itu, aktor guru berperan sebagai alternatif dalam proses evaluasi dan dapat melihat hasil penilaian mereka. Use Case Diagram ini memberikan gambaran umum mengenai fungsionalitas sistem serta bagaimana masing-masing aktor berinteraksi dengan sistem.

Diagram di atas menunjukkan rancangan use case untuk aplikasi sistem pendukung keputusan dalam pemilihan guru terbaik berbasis web menggunakan metode ARAS. Pada rancangan tersebut terdapat tiga aktor utama, yaitu **admin**, **kepala sekolah**, dan **guru**. Admin dan kepala sekolah bertindak sebagai pengelola sistem, yang memiliki akses penuh untuk mengelola data pengguna, kriteria, subkriteria, alternatif (guru), serta melakukan proses perhitungan dan melihat hasil pemilihan guru terbaik berdasarkan metode ARAS. Keduanya juga memiliki wewenang untuk mencetak laporan hasil perhitungan sebagai dokumentasi atau bahan pertimbangan dalam pengambilan keputusan.

Sementara itu, aktor guru juga dapat mengakses sistem secara langsung, namun hanya memiliki hak akses terbatas. Guru hanya dapat melihat hasil penilaian atau peringkat mereka sendiri berdasarkan kriteria yang telah ditentukan oleh pihak pengelola. Mereka tidak memiliki hak untuk mengubah data maupun memproses perhitungan. Pendekatan ini bertujuan untuk memberikan transparansi terhadap hasil evaluasi sekaligus menjaga keamanan dan integritas data di dalam sistem. Dengan demikian, sistem ini dapat berjalan sesuai alur operasional sekolah yang melibatkan partisipasi pengelola dan keterlibatan guru sebagai pihak yang dievaluasi. Berikut adalah penjelasan rincinya:

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Use Case** | **Keterangan** | **Aktor** |
| 1 | Login | Proses autentikasi pengguna sebelum mengakses sistem. | Admin, Kepala Sekolah, Guru |
| 2 | Input Nilai Alternatif | Memasukkan data alternatif (guru) yang akan dinilai dalam sistem. | Admin, Kepala Sekolah |
| 3 | Input Nilai Kriteria | Menentukan bobot dan jenis kriteria penilaian (benefit/cost). | Admin, Kepala Sekolah |
| 4 | Input Nilai Sub Kriteria | Menambahkan sub-kriteria dari masing-masing kriteria untuk penilaian detail. | Admin, Kepala Sekolah |
| 5 | Mengelola Data Guru | Menambah, mengubah, atau menghapus data guru sebagai entitas alternatif. | Admin, Kepala Sekolah |
| 6 | Mengelola Data Admin | Menambah atau mengelola akun admin dalam sistem. | Admin, Kepala Sekolah |
| 7 | Proses Hitung ARAS | Menjalankan perhitungan sistem dengan metode ARAS untuk menentukan peringkat. | Admin, Kepala Sekolah |
| 8 | Melihat Hasil Ranking | Melihat hasil akhir dari perhitungan pemilihan guru terbaik. | Admin, Kepala Sekolah, Guru |
| 9 | Mencetak Laporan | Menghasilkan laporan hasil perhitungan dalam bentuk dokumen atau file. | Admin, Kepala Sekolah |
| 10 | Melihat Data Pribadi | Guru melihat hasil penilaian atau peringkat pribadinya di sistem. | Guru |
| 11 | Logout | Keluar dari sistem untuk mengakhiri sesi. | Admin, Kepala Sekolah, Guru |

### 3.3.2 Activity Diagram

Activity Diagram digunakan untuk menggambarkan alur aktivitas atau proses bisnis dari sistem secara terstruktur. Diagram ini memperlihatkan bagaimana sistem merespons suatu proses, mulai dari awal hingga selesai, termasuk keputusan-keputusan yang mungkin terjadi dalam alur tersebut. Activity diagram sangat membantu dalam memahami urutan logika proses yang terjadi dalam sistem.

Pada sistem ini, activity diagram menggambarkan bagaimana proses pemilihan guru terbaik dilakukan, dimulai dari pengguna (admin/kepala sekolah) melakukan login, mengelola data, memproses perhitungan metode ARAS, hingga melihat dan mencetak hasilnya. Guru sebagai aktor juga memiliki jalur aktivitas tersendiri yaitu login dan melihat hasil penilaian mereka.

1. **Activity Diagram Login**

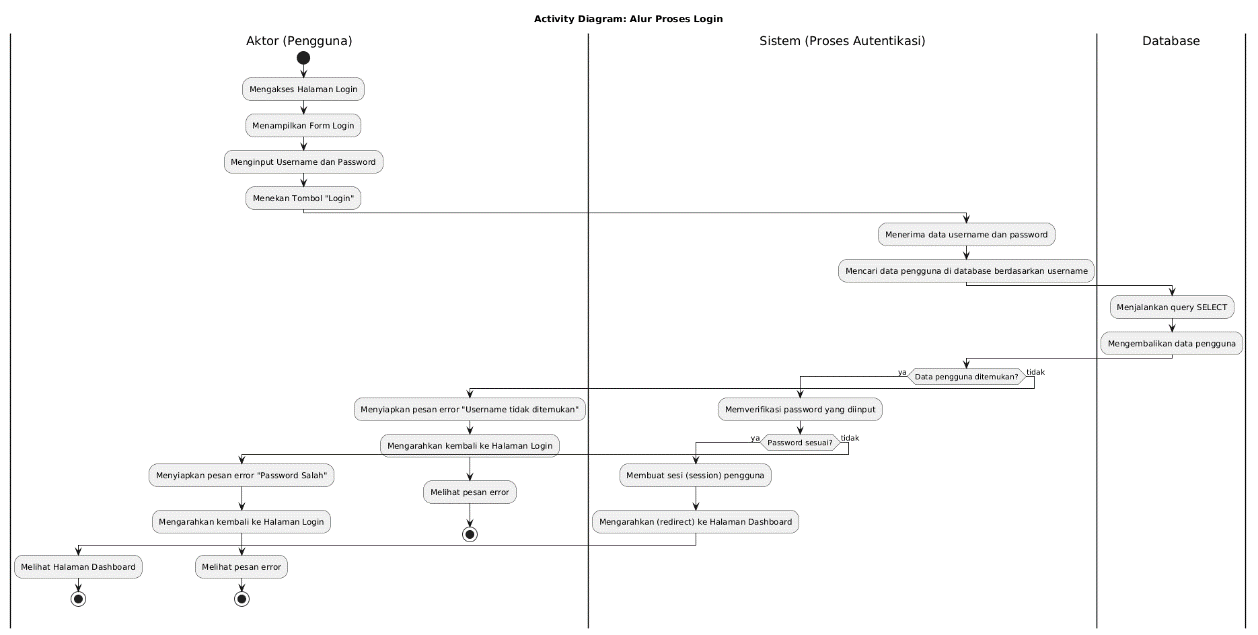


Diagram aktivitas proses login menjelaskan alur kerja pengguna saat memasuki sistem. Proses dimulai ketika pengguna mengakses halaman login, menginput username dan password, lalu menekan tombol login. Sistem kemudian memvalidasi kredensial tersebut dengan data yang tersimpan di database. Jika validasi berhasil, sistem akan membuat sesi (session) untuk pengguna dan mengarahkannya ke halaman dashboard utama. Namun, jika username tidak ditemukan atau password salah, sistem akan menampilkan pesan error dan mengembalikan pengguna ke halaman login untuk mencoba kembali.

1. **Activity Diagram Input Data Alternatif**

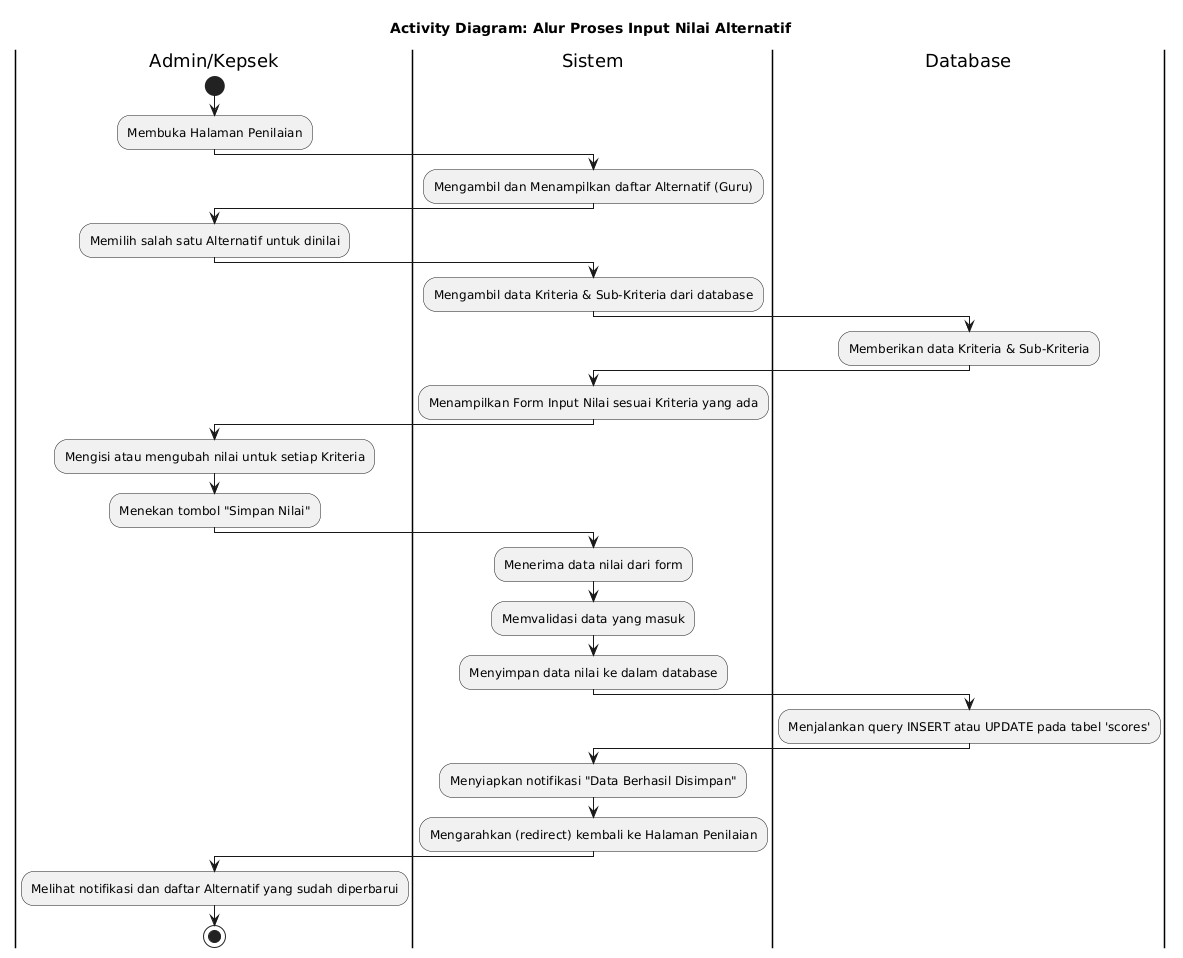
****

Diagram aktivitas proses input nilai alternatif menggambarkan alur kerja dalam memasukkan data penilaian untuk setiap guru. Alur diawali oleh Aktor (Admin/Kepala Sekolah) yang mengakses halaman penilaian dan memilih salah satu alternatif (guru) untuk dinilai. Sistem kemudian menampilkan form yang berisi daftar kriteria, yang diisi oleh Aktor. Setelah data nilai disimpan, sistem akan melakukan validasi dan menyimpan informasi tersebut ke dalam tabel 'scores' di database, lalu memberikan notifikasi keberhasilan dan mengembalikan Aktor ke halaman penilaian.

1. **Activity Diagram Input Data Kriteria**

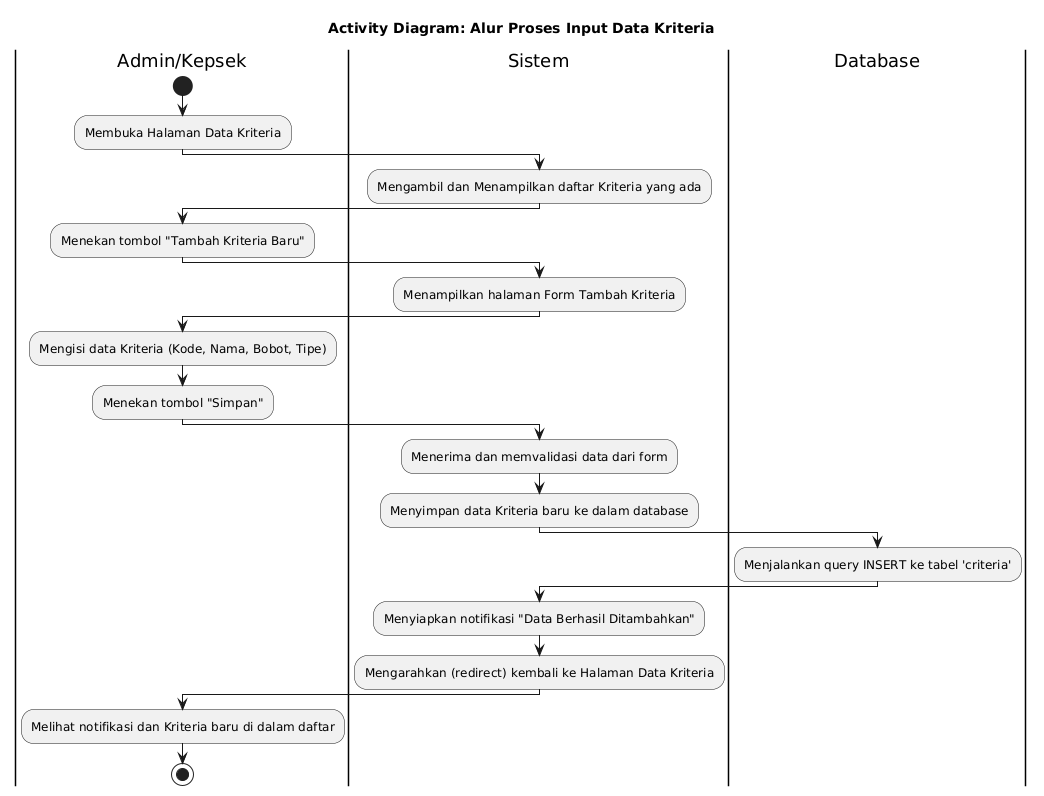


Diagram aktivitas proses input data kriteria merinci langkah-langkah untuk menambahkan kriteria penilaian baru ke dalam sistem. Proses ini diinisiasi oleh Aktor yang memilih menu untuk menambah kriteria baru, yang kemudian akan diarahkan oleh sistem ke halaman form input. Aktor mengisi seluruh atribut yang diperlukan seperti kode, nama kriteria, bobot, dan tipe. Setelah disimpan, sistem memvalidasi dan mengeksekusi perintah INSERT ke tabel 'criteria' di database, kemudian menampilkan notifikasi sukses dan memuat ulang halaman daftar kriteria.

1. **Activity Diagram Input Data Sub Kriteria**

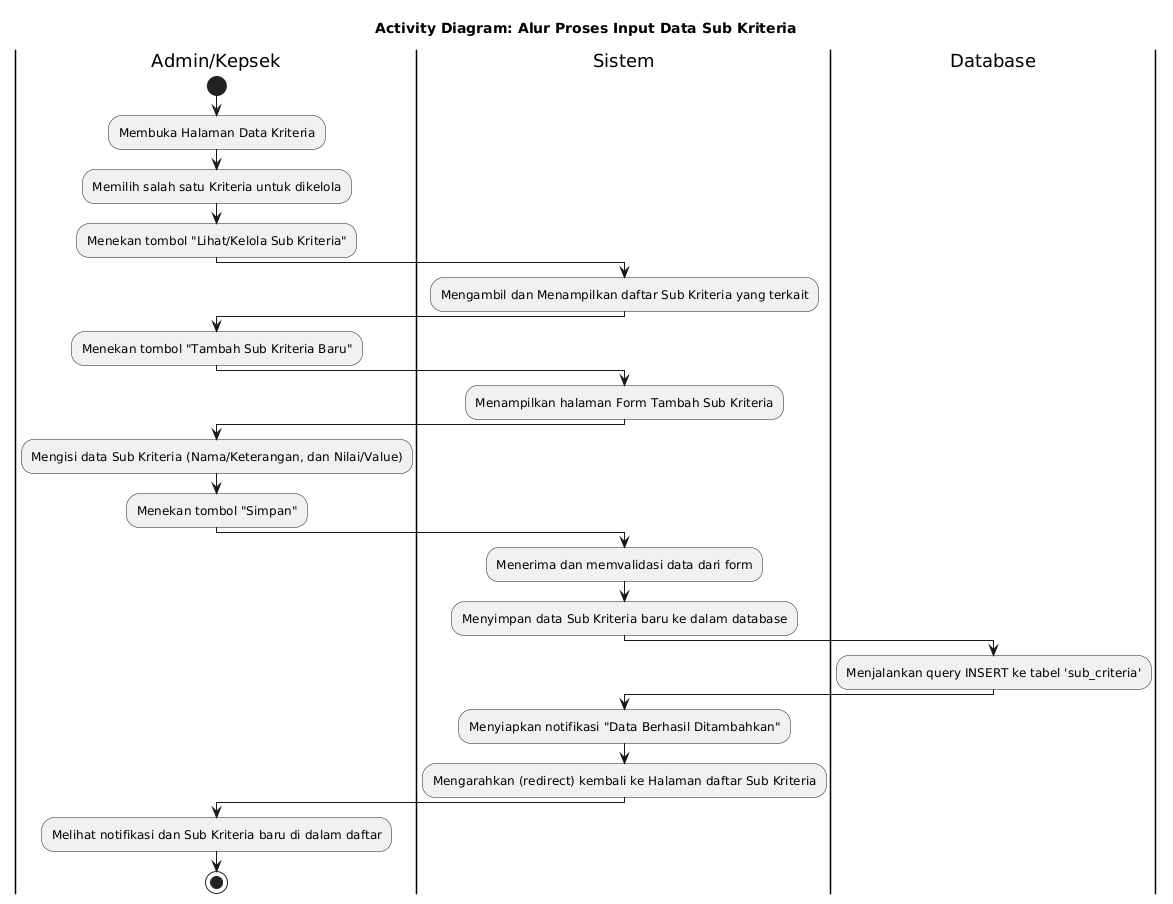
****

Diagram aktivitas proses input data sub kriteria menjelaskan alur untuk menambahkan detail dari sebuah kriteria utama. Alur dimulai ketika Aktor memilih sebuah kriteria induk dan mengakses menu pengelolaan sub kriterianya. Sistem akan menampilkan form untuk menambahkan sub kriteria baru, di mana Aktor menginputkan keterangan dan nilai (value) dari sub kriteria tersebut. Setelah data disimpan, sistem akan menyimpannya ke dalam tabel 'sub\_criteria' di database dan menghubungkannya dengan kriteria induk yang bersangkutan, lalu menampilkan kembali daftar sub kriteria yang telah diperbarui.

1. **Activity Diagram Mengelola Data Guru**

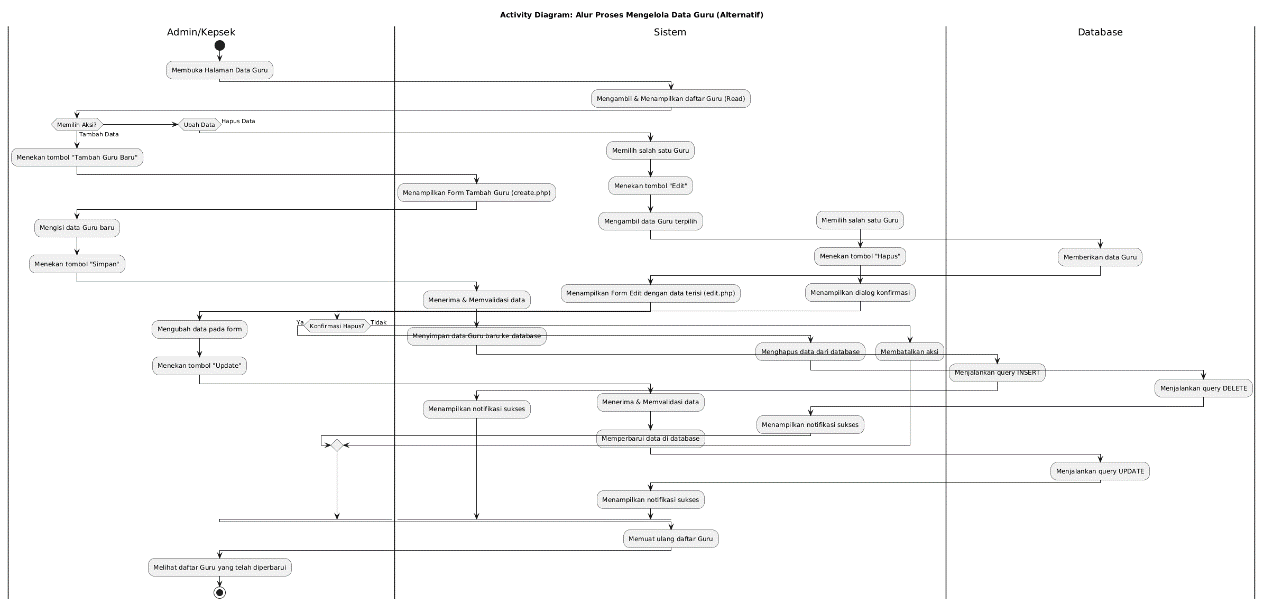
****

Diagram aktivitas proses mengelola data guru menggambarkan alur kerja CRUD (Create, Read, Update, Delete) secara lengkap. Proses diawali dengan sistem menampilkan daftar guru yang ada (Read). Dari halaman ini, Aktor dapat memilih salah satu dari tiga aksi: menambah data guru baru melalui form tambah (Create), mengubah data guru yang sudah ada melalui form edit (Update), atau menghapus data guru dari sistem setelah melalui tahap konfirmasi (Delete). Setiap aksi akan diakhiri dengan sistem menyimpan perubahan ke database dan memuat ulang daftar guru untuk menampilkan data terkini.

1. **Activity Diagram Mengelola Data Admin**

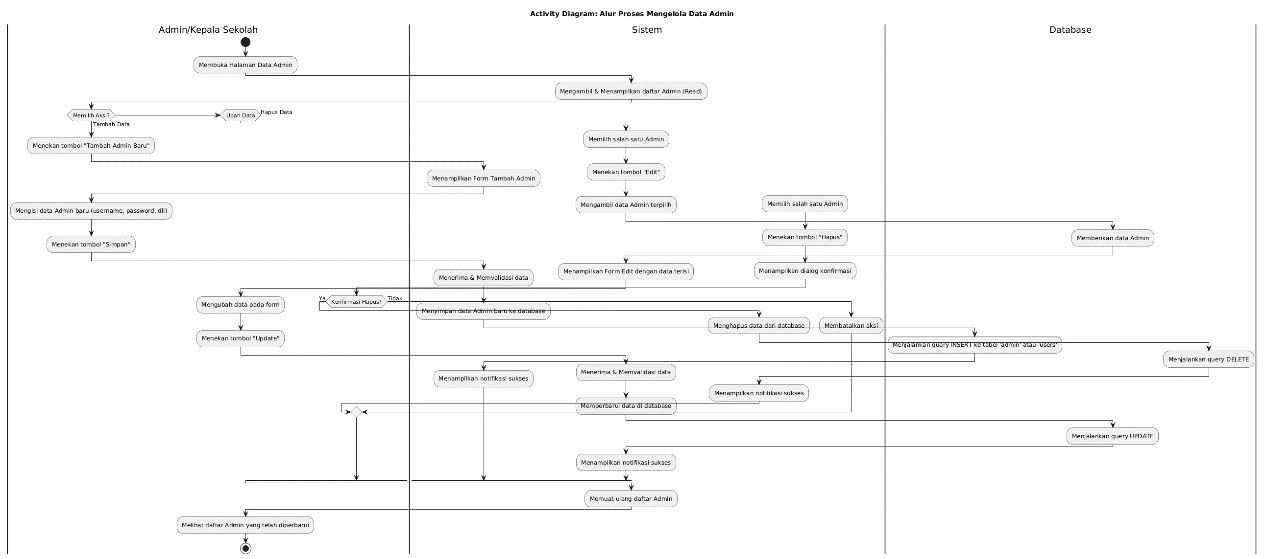
****

Diagram aktivitas proses mengelola data admin memiliki alur kerja yang identik dengan pengelolaan data guru, yang merupakan representasi dari proses CRUD standar. Aktor dengan hak akses tertinggi dapat mengakses halaman pengelolaan admin untuk melihat daftar admin yang terdaftar (Read). Dari sana, Aktor dapat melakukan penambahan admin baru (Create), mengubah data admin yang ada (Update), maupun menghapus akun admin lain dari sistem (Delete). Seluruh perubahan akan disimpan di dalam tabel 'admin' atau 'users' dan sistem akan menampilkan kembali daftar admin yang mutakhir.

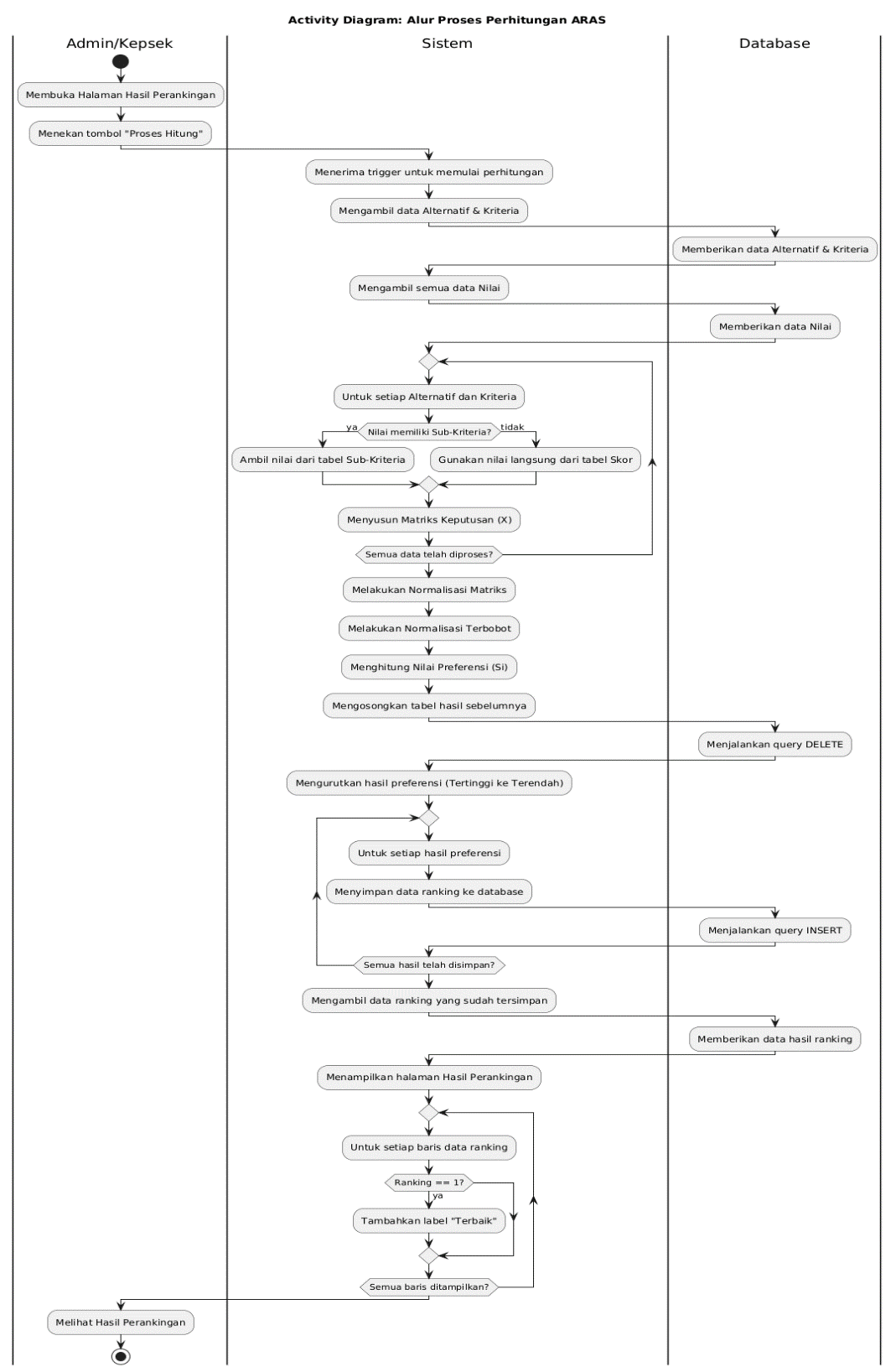
1. **Activity Diagram Proses Perhitungan ARAS**

Diagram aktivitas proses perhitungan ARAS merupakan diagram inti yang menjelaskan alur kerja algoritma sistem. Proses dipicu oleh Aktor yang menekan tombol proses hitung. Sistem kemudian secara sekuensial melakukan serangkaian tugas: mengambil seluruh data alternatif, kriteria, dan nilai dari database; menyusun matriks keputusan awal; melakukan tahap normalisasi dan pembobotan sesuai metodologi ARAS; dan menghitung nilai preferensi akhir untuk setiap alternatif. Hasil perhitungan tersebut kemudian disimpan ke dalam tabel 'results' di database sebelum akhirnya ditampilkan dalam bentuk tabel perankingan kepada pengguna.

1. **Activity Diagram Melihat Hasil Ranking**

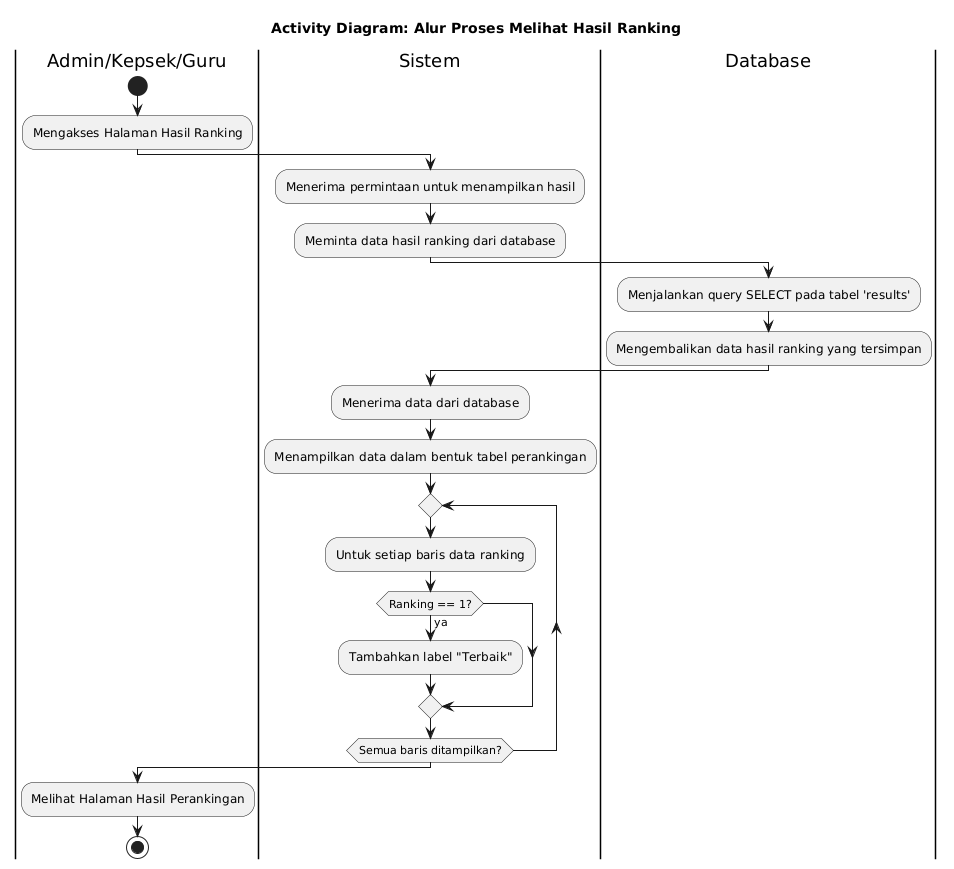


Diagram aktivitas proses melihat hasil ranking menggambarkan alur kerja sederhana saat pengguna ingin melihat data peringkat yang sudah tersimpan tanpa memicu perhitungan ulang. Alur dimulai saat Aktor (Admin, Kepala Sekolah, atau Guru) mengakses halaman hasil ranking. Sistem kemudian akan langsung mengirimkan perintah SELECT ke database untuk mengambil data dari tabel 'results'. Data peringkat yang diterima dari database selanjutnya ditampilkan dalam format tabel yang terurut kepada Aktor.

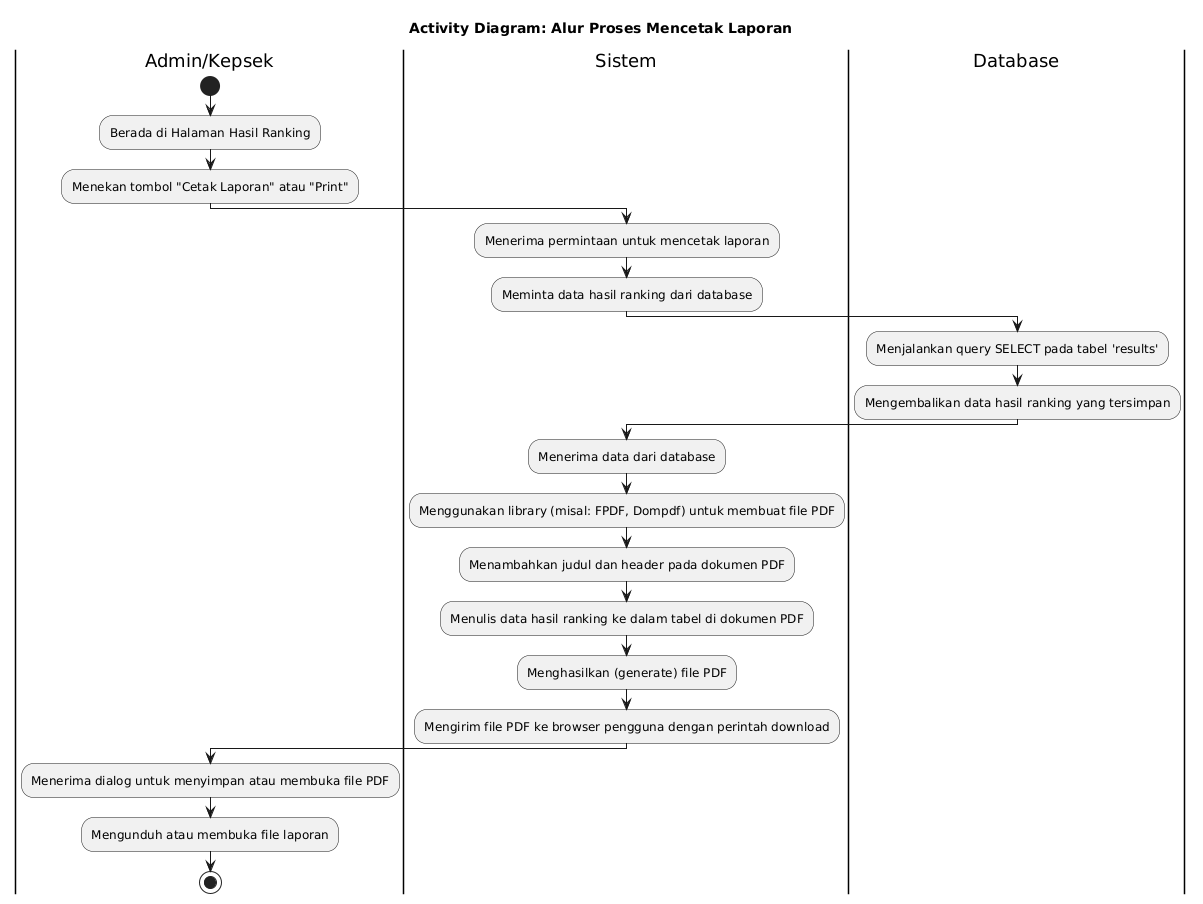
1. **Activity Diagram Mencetak Laporan**

Diagram aktivitas proses mencetak laporan menjelaskan bagaimana sistem mengonversi data hasil perankingan menjadi sebuah dokumen fisik atau digital. Proses diawali oleh Aktor yang menekan tombol cetak pada halaman hasil ranking. Sistem kemudian mengambil data ranking yang relevan dari database, lalu menggunakan sebuah library pembuat dokumen (contoh: FPDF atau Dompdf) untuk menghasilkan file PDF secara dinamis. File PDF yang telah berisi tabel hasil perankingan ini kemudian dikirimkan oleh sistem ke browser pengguna untuk diunduh atau dicetak.

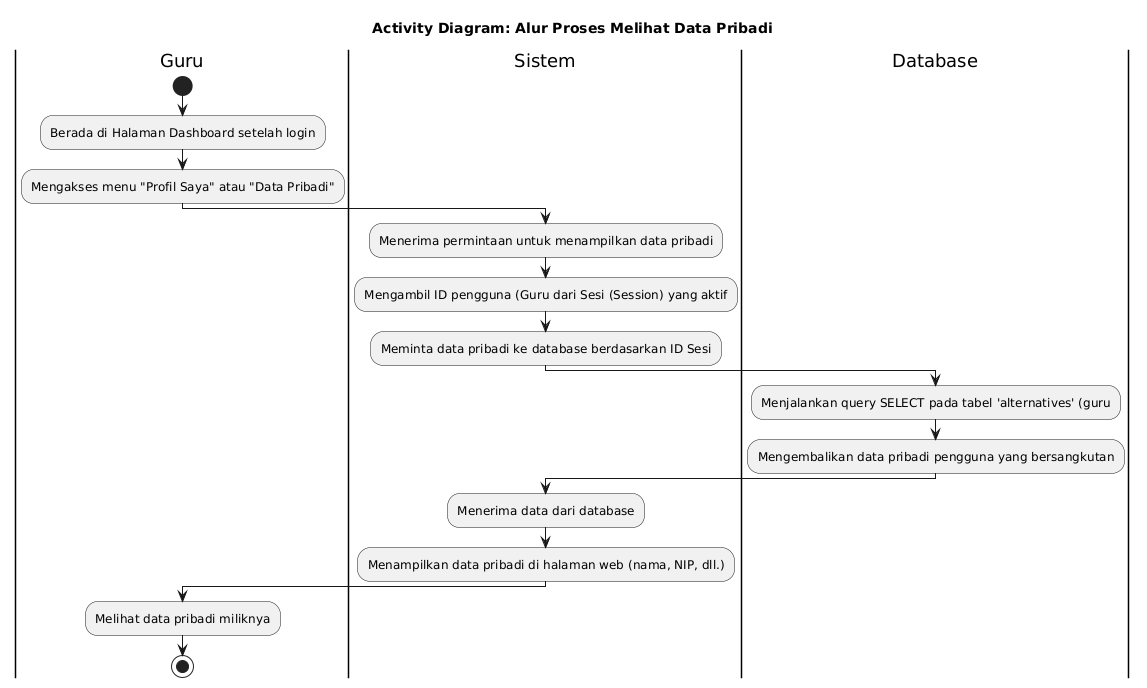
1. **Activity Diagram Melihat Data Pribadi**

Diagram aktivitas proses melihat data pribadi menggambarkan alur kerja yang spesifik untuk Aktor Guru. Setelah login, Guru dapat mengakses halaman profilnya. Sistem akan secara otomatis mengidentifikasi pengguna yang sedang aktif melalui data sesi (session). Menggunakan ID dari sesi tersebut, sistem mengambil data pribadi guru yang bersangkutan dari tabel 'alternatives' di database. Informasi ini kemudian ditampilkan di halaman web, untuk memastikan bahwa seorang guru hanya dapat melihat datanya sendiri.

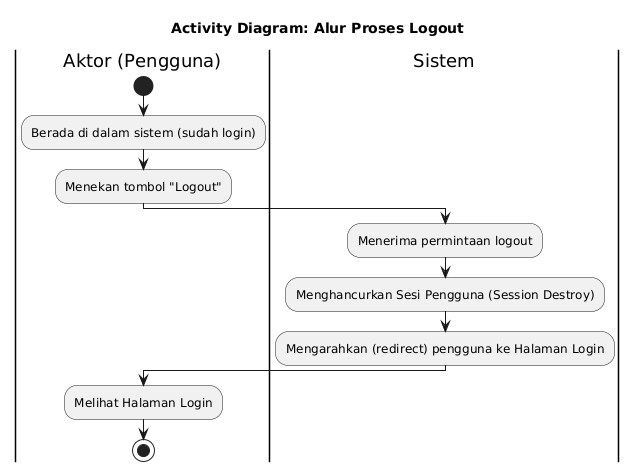
1. **Activity Diagram Logout**

Diagram aktivitas proses logout merinci alur kerja yang krusial untuk keamanan sistem. Proses ini diinisiasi oleh pengguna mana pun yang sedang login dengan menekan tombol logout. Sistem akan segera merespons dengan menghancurkan sesi (session destroy) pengguna di sisi server, yang secara efektif menghapus status login mereka. Setelah sesi berhasil dihancurkan, sistem secara otomatis mengarahkan (redirect) browser pengguna kembali ke halaman login, memastikan akses ke dalam sistem telah terputus sepenuhnya.

### 3.3.3 Sequence Diagram

Sequence diagram adalah salah satu jenis diagram UML yang menggambarkan interaksi antar objek dalam sistem berdasarkan urutan waktu. Diagram ini menunjukkan bagaimana objek-objek saling berkomunikasi melalui pesan (message) dalam suatu skenario atau proses tertentu. Sequence diagram digunakan untuk memodelkan alur proses secara dinamis dan runtut sehingga memudahkan pengembang dalam memahami interaksi antara pengguna dan sistem. Pada sistem ini, sequence diagram dibuat untuk menggambarkan proses-proses utama seperti login admin, pengelolaan data, hingga perhitungan dan penampilan hasil pemilihan guru terbaik menggunakan metode ARAS.

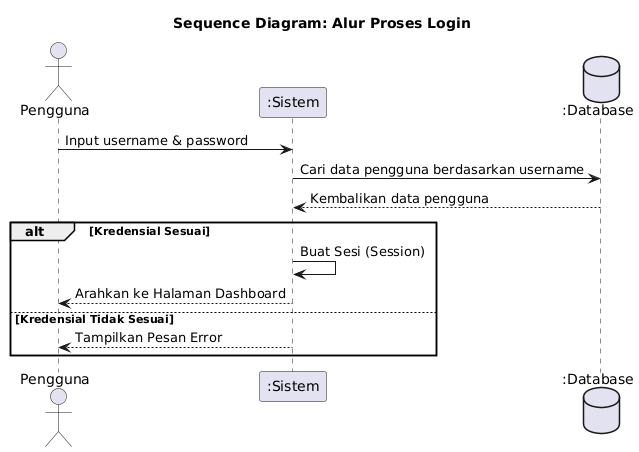
1. **Sequence** **Diagram Login**

Diagram ini menggambarkan urutan interaksi saat pengguna masuk ke sistem. Pengguna menginputkan kredensial ke dalam Sistem, yang kemudian memvalidasinya dengan data di Database. Jika kredensial sesuai, Sistem akan membuat sebuah sesi (session) internal untuk menandai status login dan mengarahkan pengguna ke halaman dashboard. Jika tidak, Sistem akan menampilkan pesan error kepada pengguna.

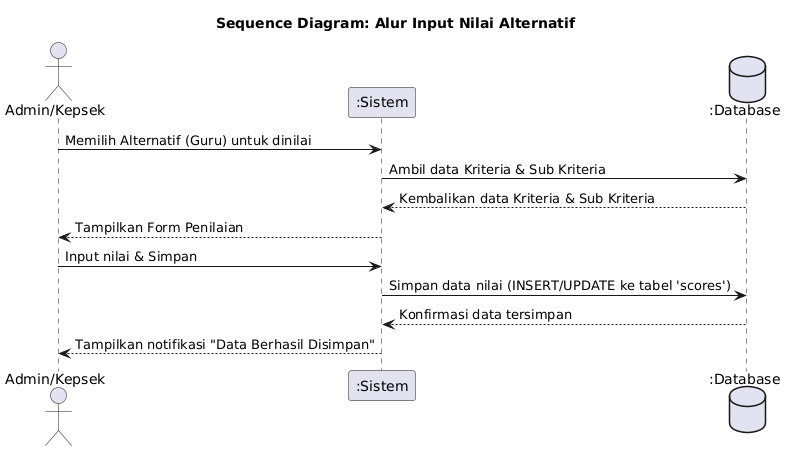
1. **Sequence Diagram Input Nilai Alternatif**

Diagram ini menjelaskan interaksi saat Aktor memasukkan data penilaian. Alur dimulai ketika Aktor memilih seorang guru di dalam Sistem, yang kemudian direspons oleh Sistem dengan menampilkan form penilaian setelah mengambil data kriteria dari Database. Setelah Aktor mengirimkan nilai, Sistem akan menyimpannya ke dalam Database dan memberikan notifikasi keberhasilan sebagai konfirmasi akhir.

1. **Sequence Diagram Input Data Kriteria**

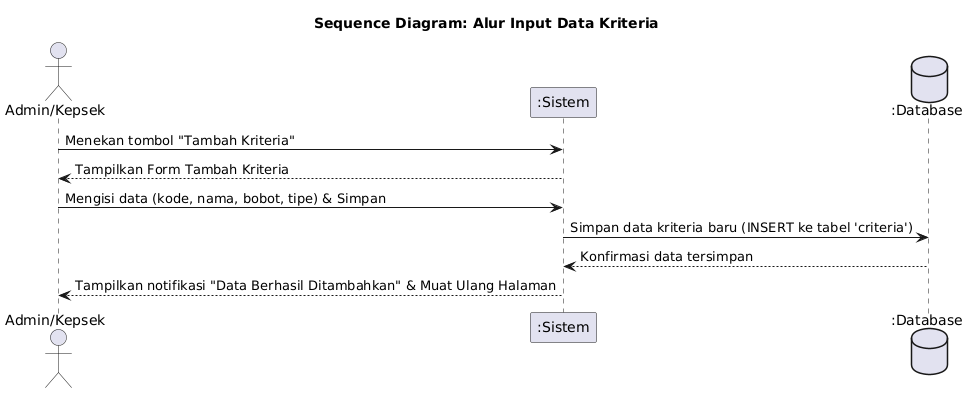


Diagram ini menunjukkan alur pembuatan data kriteria baru. Aktor memulai interaksi dengan meminta Sistem untuk menampilkan form tambah kriteria. Setelah Aktor mengisi dan mengirimkan data, Sistem akan meneruskannya ke Database untuk disimpan menggunakan perintah INSERT. Sebagai respons akhir, Sistem memberikan notifikasi sukses kepada Aktor.

1. **Sequence Diagram Input Data Sub Kriteria**

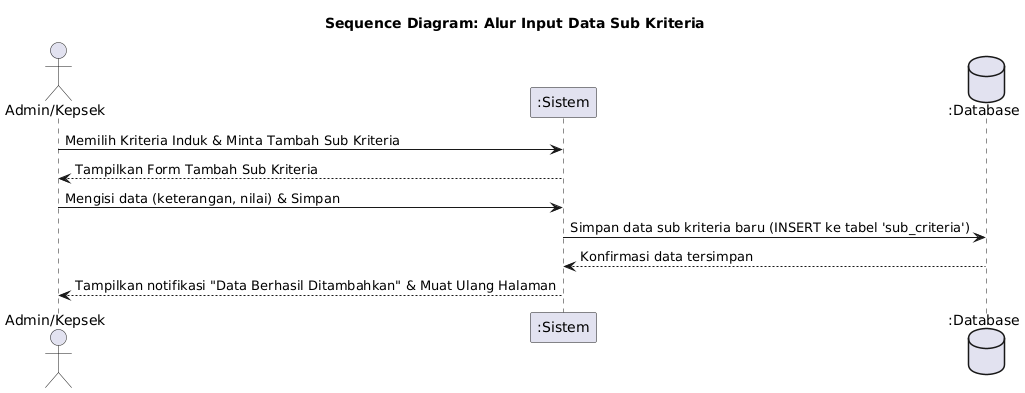


Diagram ini merinci alur pembuatan data sub kriteria yang terikat pada sebuah kriteria induk. Aktor memilih kriteria induk dan meminta untuk menambah sub kriteria, yang direspons oleh Sistem dengan menampilkan form. Setelah data diisi dan disimpan, Sistem akan menyimpannya ke Database dengan relasi ke kriteria induk, lalu memberikan notifikasi keberhasilan kepada Aktor.

1. **Sequence Diagram Mengelola Data Guru**

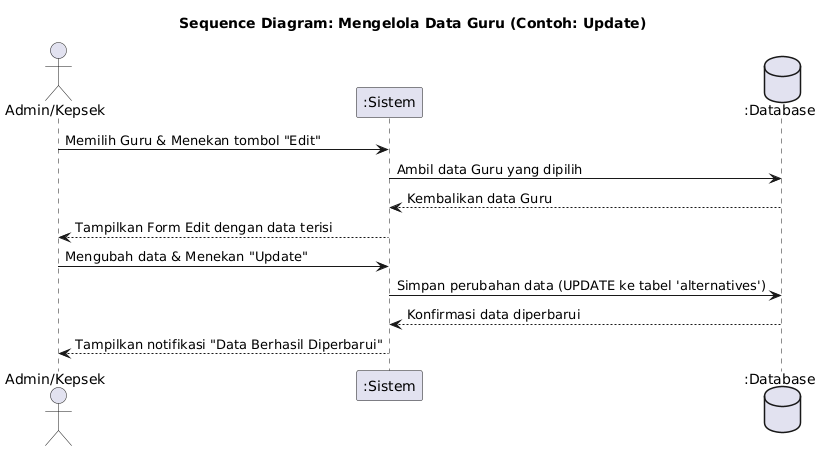


Diagram ini menggambarkan interaksi untuk salah satu proses pengelolaan data, yaitu mengubah data guru. Aktor memilih data guru untuk diedit, lalu Sistem mengambil data tersebut dari Database untuk ditampilkan di form. Setelah Aktor melakukan perubahan dan menyimpan, Sistem akan mengirimkan data baru ke Database untuk diperbarui (di-UPDATE), kemudian menampilkan notifikasi sukses.

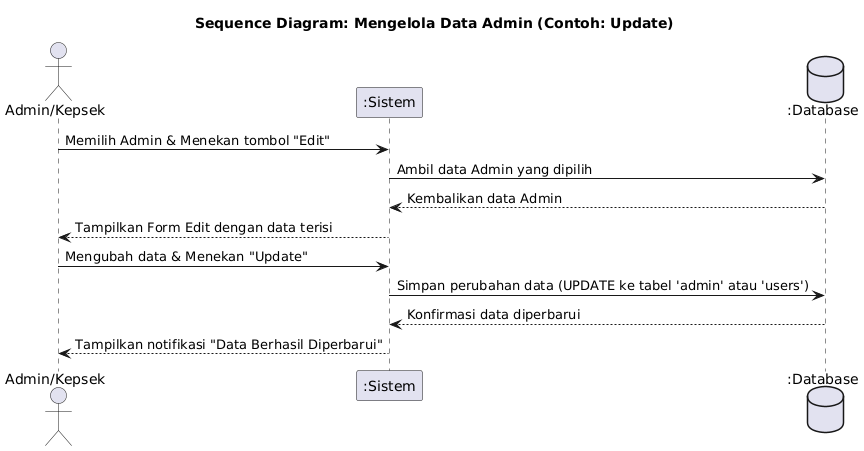
1. **Mengelola Data Admin**

Diagram ini memiliki pola interaksi yang identik dengan pengelolaan data guru, yang merepresentasikan alur UPDATE standar. Aktor memilih akun admin untuk diubah, lalu Sistem mengambil data yang ada dari Database untuk ditampilkan. Setelah Aktor mengirimkan pembaruan, Sistem menyimpannya ke Database dan memberikan pesan konfirmasi kepada Aktor.

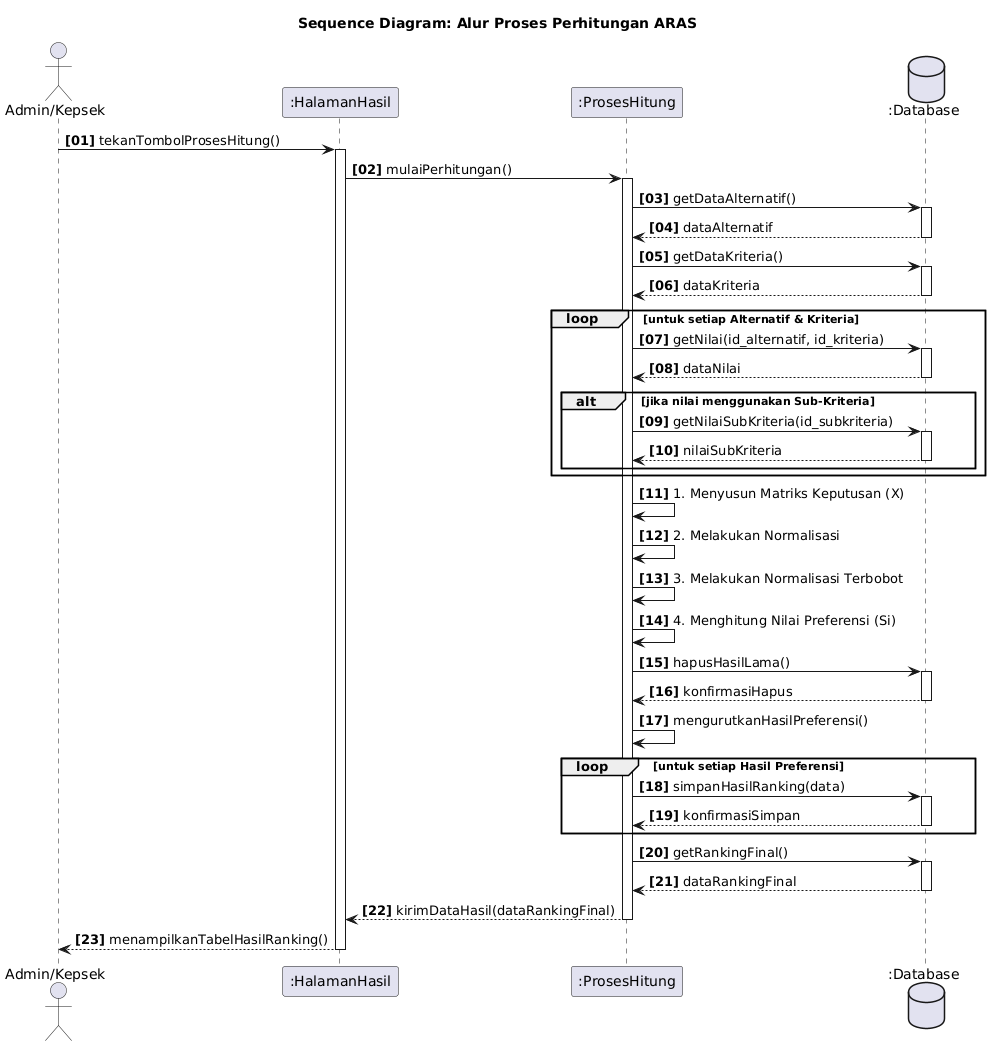
1. **Sequence Diagram Hitung Aras**

Diagram sekuensial ini merinci interaksi antar komponen selama proses perhitungan ARAS. Aktor memicu proses pada Halaman Web (View), yang kemudian mendelegasikan tugas ke Proses Backend (Controller). Controller memulai serangkaian komunikasi dengan Database untuk mengambil semua data (alternatif, kriteria, nilai), melakukan kalkulasi internal sesuai metode ARAS, menghapus data hasil lama, menyimpan hasil ranking yang baru, dan terakhir mengirimkan data final kembali ke View untuk ditampilkan kepada Aktor.

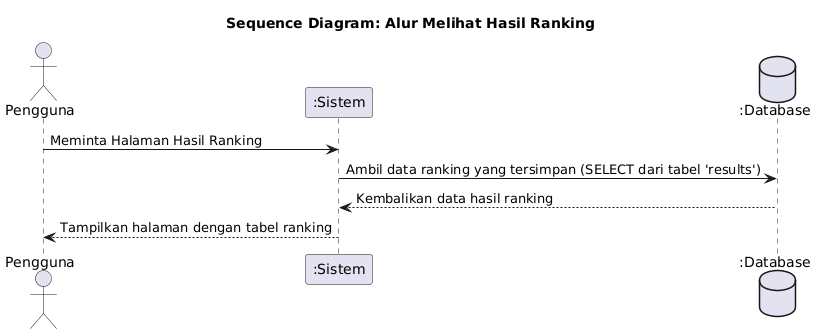
1. **Sequence Diagram Melihat Hasil Ranking**

Diagram ini menjelaskan interaksi sederhana untuk menampilkan data peringkat yang sudah ada. Aktor meminta halaman hasil ranking kepada Sistem. Sistem kemudian langsung meminta data peringkat yang tersimpan di Database, lalu menampilkannya dalam bentuk tabel kepada Aktor tanpa ada proses perhitungan ulang.

1. **Sequence Diagram Mencetak Laporan**

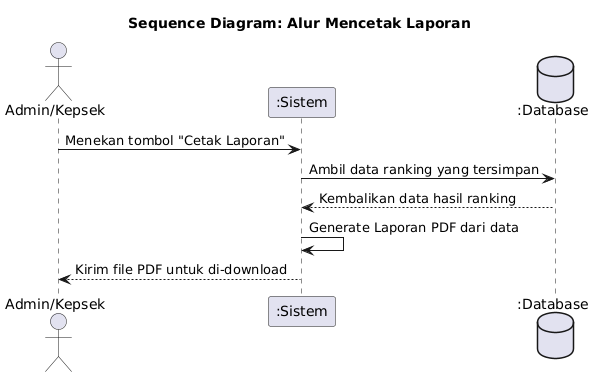


Diagram ini menunjukkan alur saat Aktor meminta sistem untuk membuat file laporan. Aktor memicu aksi cetak pada Sistem, yang kemudian mengambil data ranking terbaru dari Database. Sistem melakukan proses internal untuk menghasilkan (generate) dokumen PDF dari data tersebut, lalu mengirimkan file PDF yang sudah jadi kembali kepada Aktor untuk diunduh.

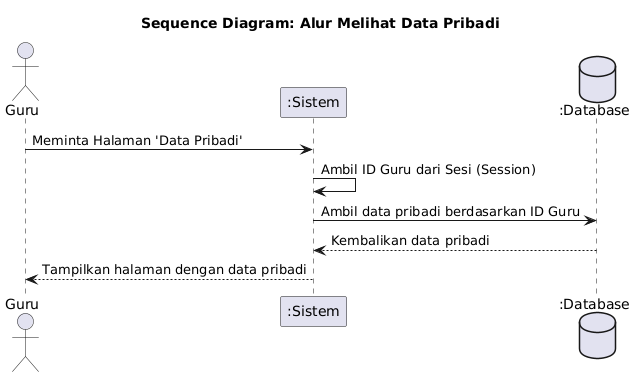
1. **Sequence Diagram Melihat Data Pribadi**

Diagram ini menjelaskan alur spesifik saat Guru melihat profilnya sendiri. Guru meminta halaman data pribadi kepada Sistem. Sistem kemudian melakukan pemeriksaan internal untuk mendapatkan ID Guru dari sesi login yang aktif, lalu menggunakan ID tersebut untuk meminta data yang spesifik dari Database, sehingga memastikan kerahasiaan dan integritas data.

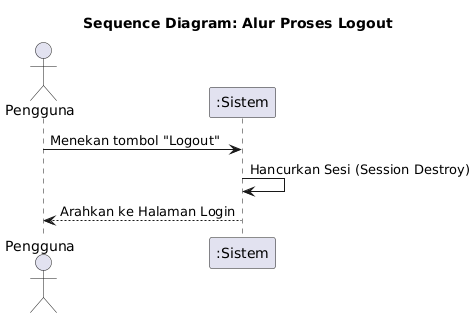
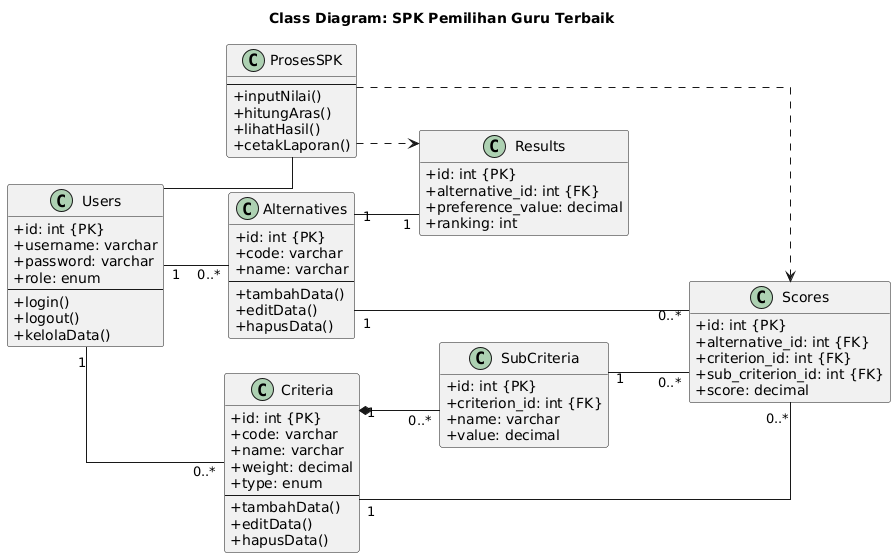
1. **Sequence Diagram Logout**

Diagram ini menggambarkan proses akhir dari sesi pengguna yang sangat ringkas. Aktor memicu aksi logout pada Sistem. Sistem merespons dengan melakukan proses internal untuk menghancurkan sesi yang aktif, kemudian langsung mengarahkan browser Aktor kembali ke halaman login.

### 3.3.4 Class Diagram

Class diagram merupakan salah satu jenis diagram UML yang digunakan untuk menggambarkan struktur statis dari sistem yang mencakup kelas-kelas yang terlibat, atribut, metode (fungsi), serta relasi antar kelas. Class diagram sangat penting dalam proses perancangan sistem berbasis objek karena menyajikan gambaran menyeluruh mengenai entitas dan hubungan antar entitas dalam sistem. Dalam pengembangan sistem pendukung keputusan pemilihan guru terbaik berbasis web ini, class diagram dibuat untuk memodelkan komponen-komponen utama sistem seperti pengguna (admin), kriteria, subkriteria, alternatif (guru), penilaian, serta proses perhitungan dan hasil akhir menggunakan metode ARAS.

Class Diagram ini memodelkan struktur kelas statis dari sistem penunjang keputusan yang dibangun, berfungsi sebagai cetak biru arsitektur perangkat lunak. Diagram ini terdiri dari kelas-kelas utama yang merepresentasikan entitas data penting seperti Users, Alternatives (Guru), Criteria, SubCriteria, Scores, dan Results, di mana setiap kelas memiliki atribut yang sesuai dengan struktur tabel di database dan metode yang menggambarkan operasi yang dapat dilakukan. Relasi antar kelas, seperti komposisi antara Criteria dan SubCriteria serta asosiasi antara Alternatives dan Scores, menunjukkan hubungan logis dan kepemilikan data. Terdapat pula kelas ProsesSPK yang bertindak sebagai controller, bertanggung jawab atas logika bisnis utama seperti hitungAras() dan memiliki dependensi terhadap kelas-kelas entitas lain untuk memanipulasi data, sehingga memberikan gambaran lengkap mengenai komponen sistem dan keterhubungannya.

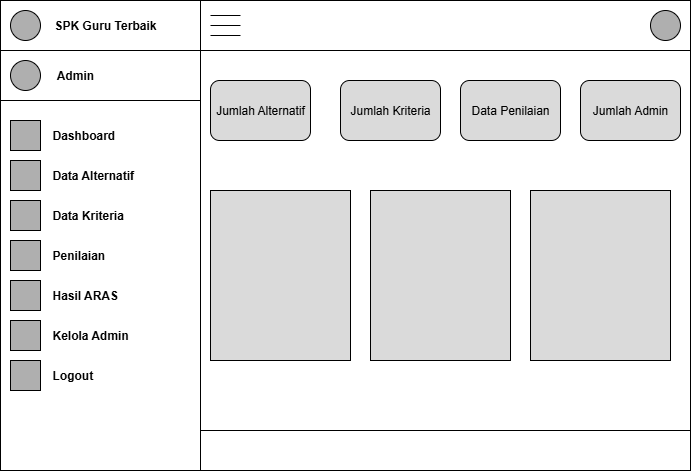
## 3.4 User Interface

User Interface (UI) merupakan tampilan antarmuka yang digunakan oleh pengguna untuk berinteraksi langsung dengan sistem. Desain UI harus dibuat seefektif dan seefisien mungkin agar memudahkan pengguna dalam menggunakan fitur-fitur yang tersedia. Dalam sistem pendukung keputusan pemilihan guru terbaik berbasis web ini, rancangan antarmuka disusun dengan memperhatikan aspek kemudahan penggunaan (user friendly), konsistensi tampilan, dan keterpaduan fungsi. Rancangan UI mencakup halaman login, dashboard admin, pengelolaan data kriteria dan alternatif, pengisian nilai, hingga tampilan hasil perhitungan menggunakan metode ARAS. Berikut merupakan rancangan tampilan aplikasi berbasis web yang dikembangkan:

1. **Login**

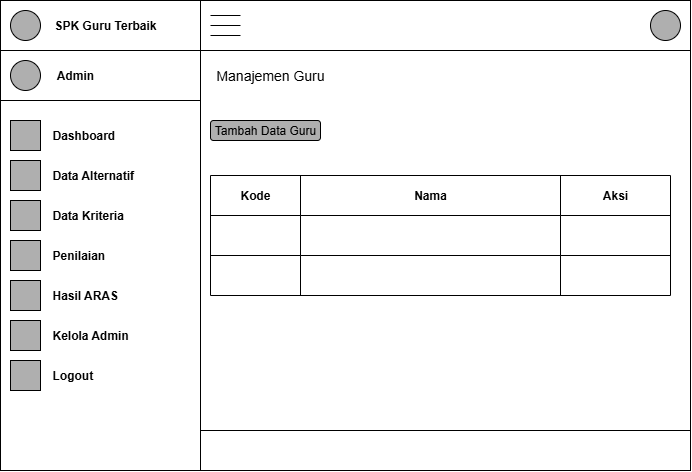
Tampilan ini adalah **gerbang utama** sistem, berfungsi sebagai halaman otentikasi. Pengguna diwajibkan untuk memasukkan **username** dan **password** yang valid untuk dapat mengakses fitur-fitur di dalam sistem. Desainnya yang minimalis memastikan pengguna dapat fokus untuk masuk ke dalam aplikasi dengan cepat dan aman.

1. **Dashboard**

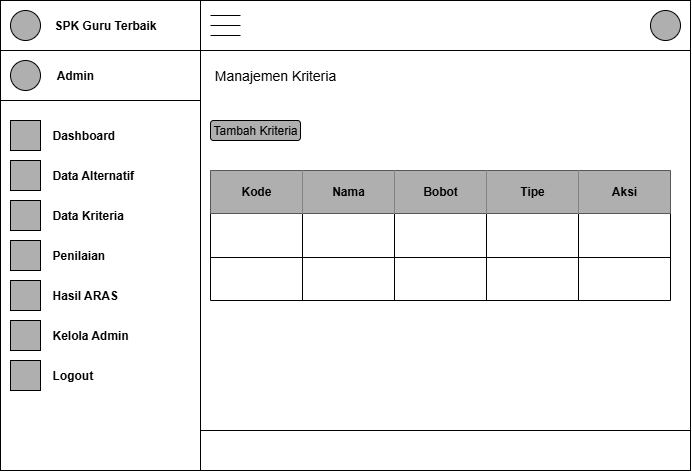


Ini adalah **halaman pertama** yang dilihat pengguna setelah berhasil login. Dashboard menyajikan **ringkasan informasi penting** dalam bentuk kartu (*cards*), seperti jumlah total alternatif (guru), kriteria, dan admin. Halaman ini juga berfungsi sebagai **pusat navigasi**, di mana pengguna dapat mengakses semua modul utama sistem melalui menu *sidebar* di sebelah kiri.

1. **Data Alternatif**

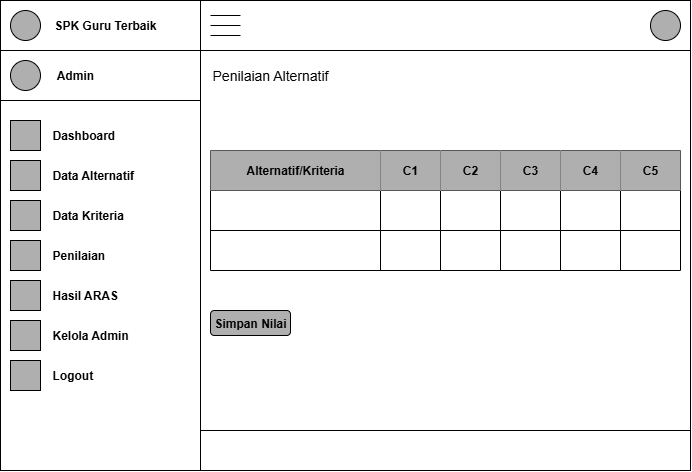


Halaman ini merupakan antarmuka untuk **mengelola data master guru** atau alternatif. Admin dapat melihat daftar semua guru yang terdaftar dalam sebuah tabel. Terdapat juga fungsionalitas untuk **menambah data guru baru** melalui tombol yang tersedia, serta melakukan aksi **edit** atau **hapus** pada setiap data guru yang sudah ada.

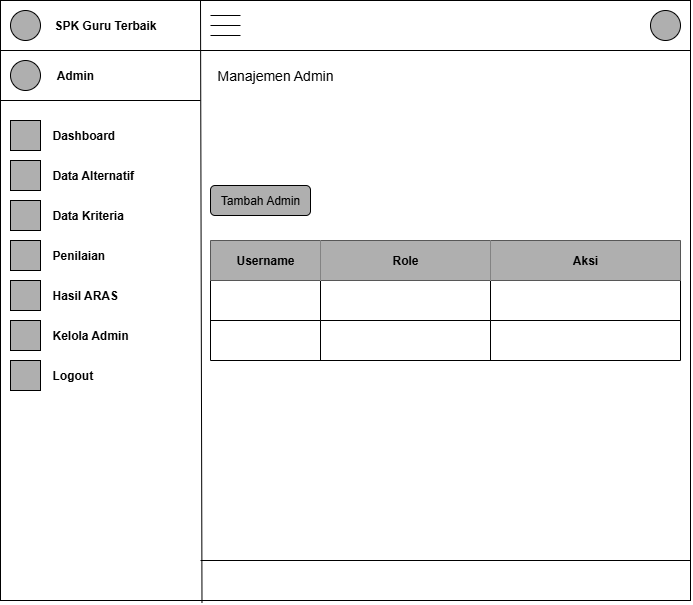
1. **Data Kriteria**

Antarmuka ini berfungsi sebagai pusat **pengelolaan data kriteria** yang akan digunakan dalam perhitungan ARAS. Admin dapat mendefinisikan, melihat, dan memodifikasi semua kriteria, termasuk **kode, nama, bobot, dan tipe** (benefit/cost). Fungsionalitas CRUD (Create, Read, Update, Delete) yang lengkap memastikan admin dapat dengan mudah menyesuaikan parameter penilaian sesuai kebutuhan.

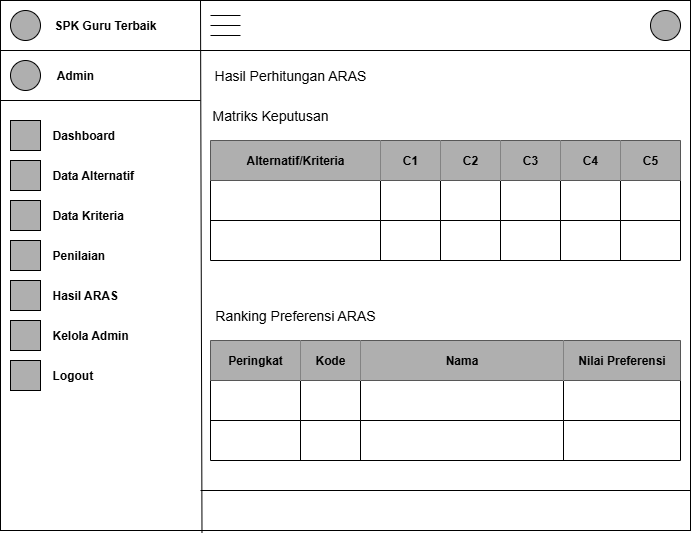
1. **Penilaian Alternatif**



Ini adalah halaman inti untuk **proses input data** dalam sistem pendukung keputusan. Antarmuka ini dirancang dalam bentuk **matriks**, memungkinkan admin untuk secara efisien memasukkan nilai (skor) untuk setiap alternatif (guru) berdasarkan setiap kriteria yang ada. Setelah semua nilai terisi, data dapat disimpan untuk kemudian diproses dalam tahap perhitungan.

1. **Kelola Admin**

Halaman ini dikhususkan untuk **pengelolaan akun pengguna** dengan hak akses admin. Melalui antarmuka ini, admin utama dapat **menambah admin baru**, melihat daftar admin yang aktif, serta mengatur peran (*role*) atau menghapus akun admin dari sistem. Ini adalah fitur penting untuk menjaga keamanan dan administrasi sistem.

1. **Hasil Aras**

Halaman ini berfungsi untuk **menampilkan hasil akhir** dari proses perhitungan metode ARAS. Tampilan ini menyajikan dua bagian utama: **matriks keputusan** sebagai langkah awal perhitungan dan **tabel ranking preferensi** sebagai hasil final. Tabel ranking mengurutkan semua guru dari peringkat tertinggi hingga terendah, sehingga pembuat keputusan dapat dengan mudah mengidentifikasi alternatif terbaik.

# 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

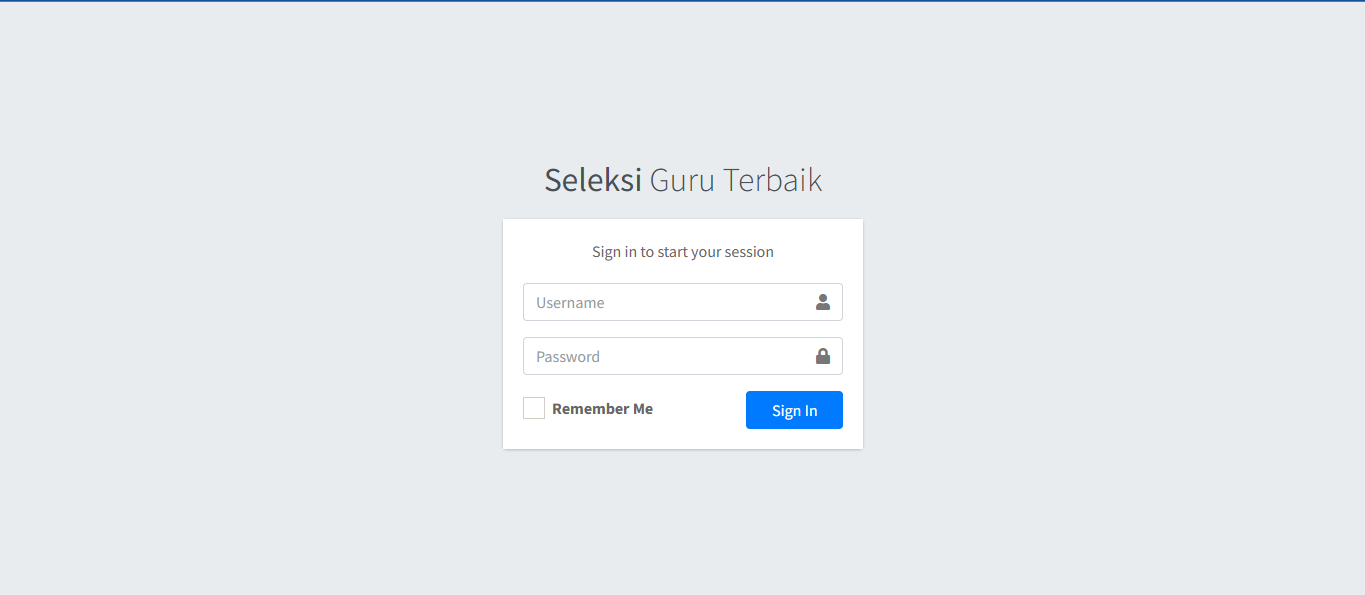
Selain perangkat keras, perangkat lunak juga sangat berperan penting dalam pengembangan sistem. Perangkat lunak digunakan sebagai alat bantu untuk melakukan perancangan antarmuka, pembuatan program, pengelolaan basis data, hingga proses perhitungan metode ARAS. Adapun spesifikasi perangkat lunak yang 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

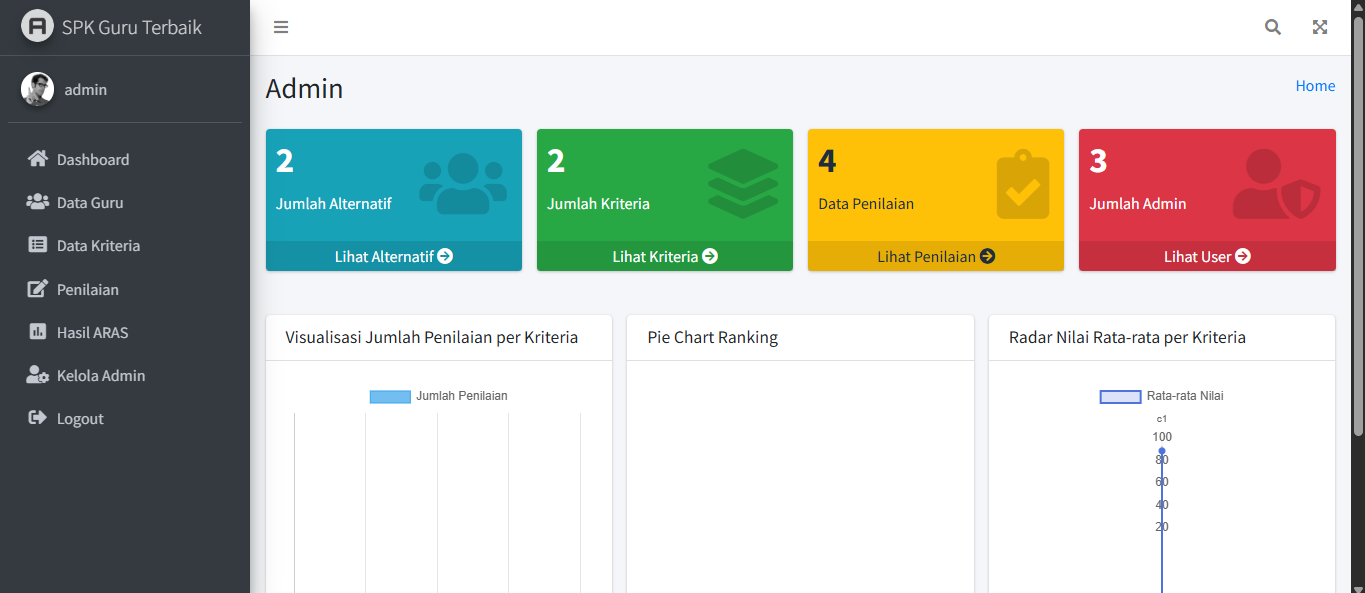
Implementasi antarmuka pada sistem ini meliputi tampilan halaman login, halaman dashboard, halaman data kriteria, halaman data alternatif, halaman penilaian alternatif, halaman hasil perhitungan menggunakan metode ARAS, serta halaman kelola admin. Penerapan antarmuka ini bertujuan untuk merealisasikan 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



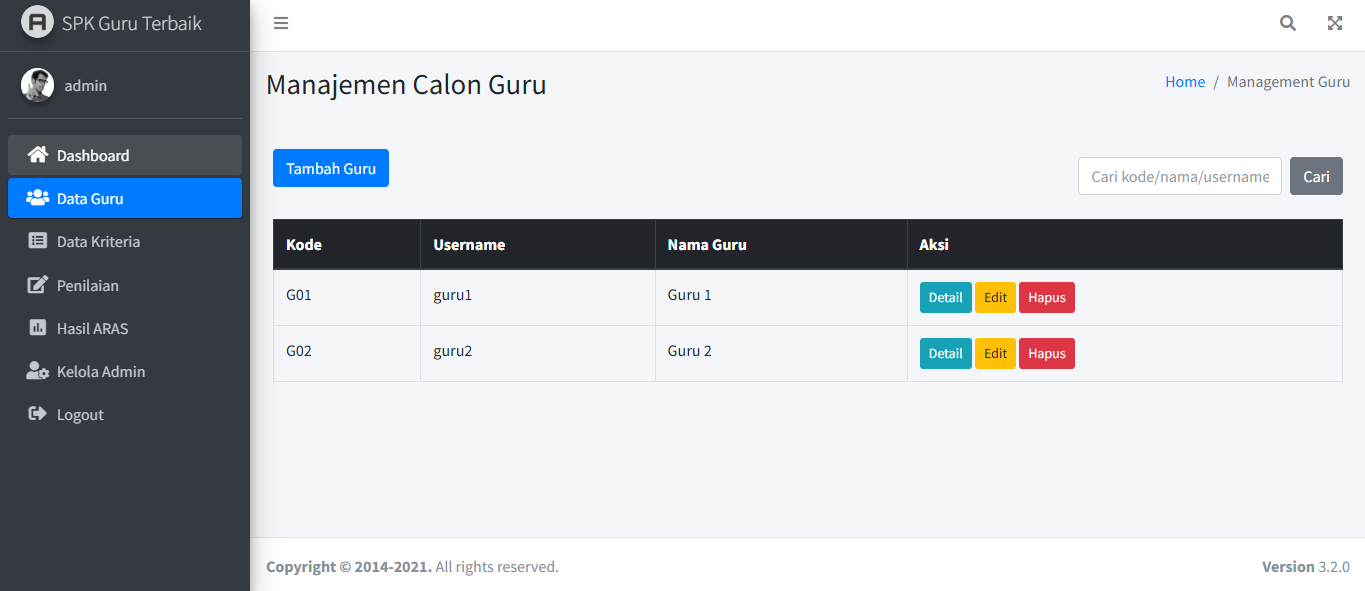
Halaman login merupakan pintu masuk utama bagi admin untuk mengakses sistem pendukung keputusan pemilihan guru terbaik. Pada halaman ini, admin diminta untuk memasukkan username dan password yang telah terdaftar sebelumnya. Tujuan dari halaman login ini adalah untuk menjaga keamanan dan membatasi akses 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



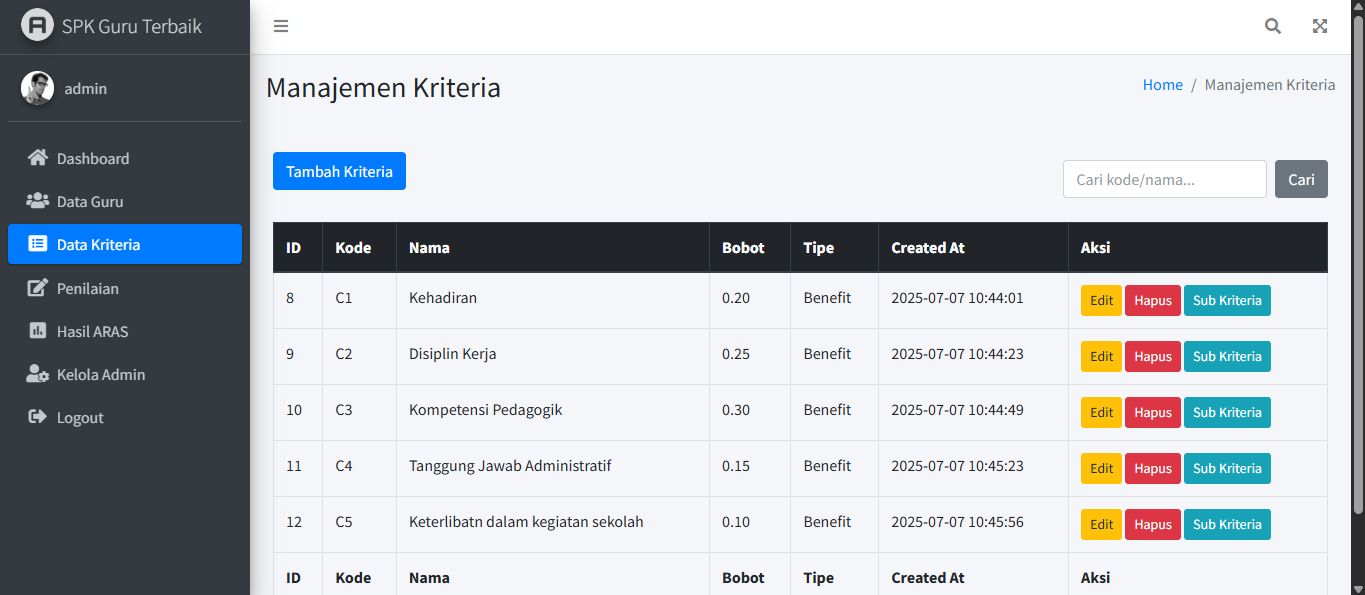
Halaman dashboard merupakan tampilan utama yang ditampilkan setelah admin berhasil login ke dalam sistem. Halaman ini berfungsi sebagai pusat navigasi untuk mengakses seluruh fitur yang tersedia dalam sistem. Pada halaman ini ditampilkan ringkasan informasi penting, seperti jumlah data guru (alternatif), 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



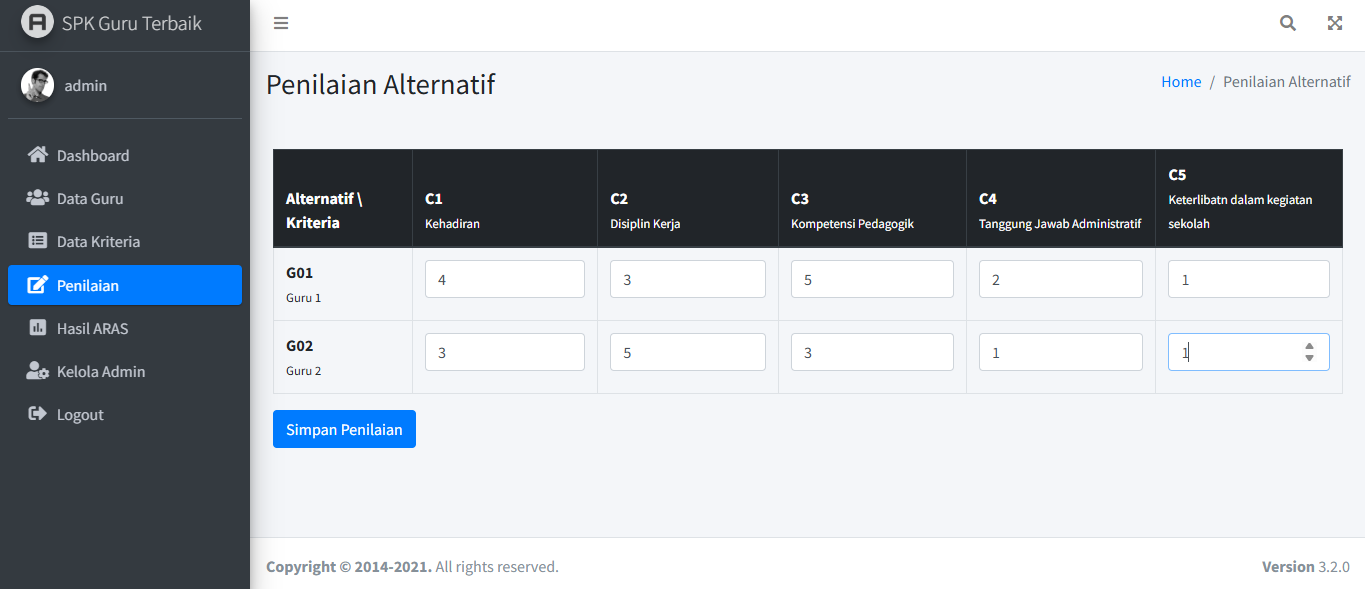
Halaman data alternatif digunakan untuk mengelola daftar guru yang akan menjadi kandidat dalam pemilihan guru terbaik. Dalam halaman ini, admin dapat melakukan input data guru seperti kode dan nama guru, serta mengedit atau menghapus data yang sudah ada. Data alternatif ini akan digunakan dalam proses 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



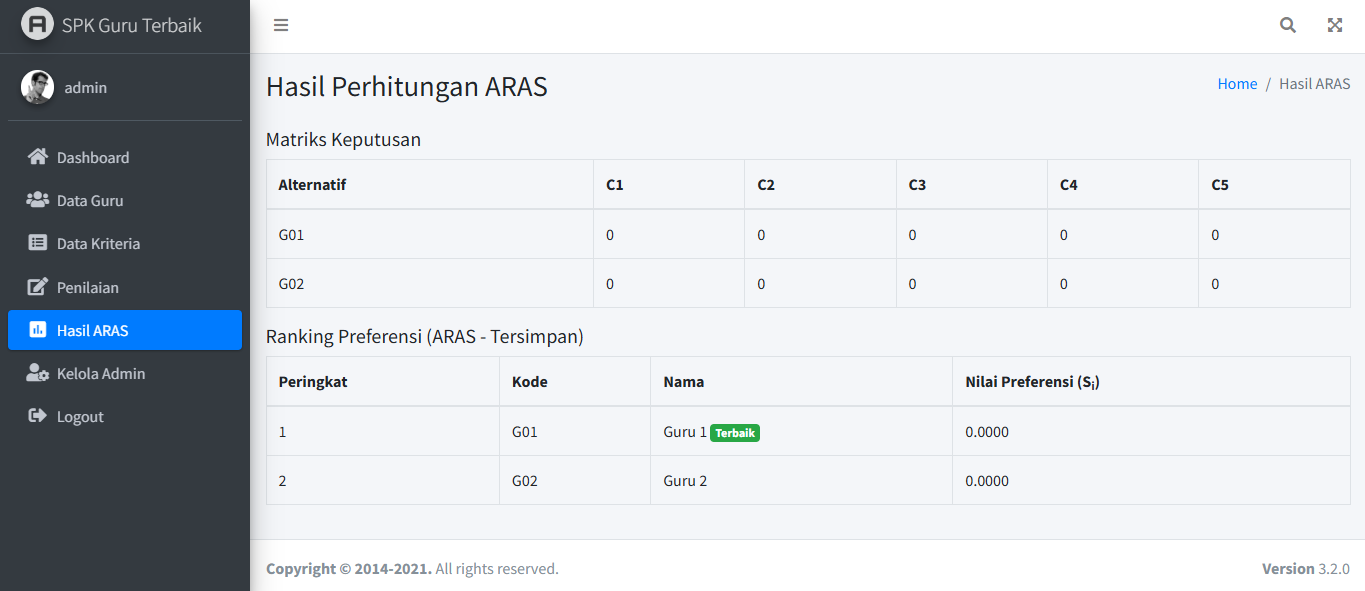
Halaman data kriteria digunakan untuk mengelola kriteria penilaian yang menjadi dasar dalam proses perhitungan metode ARAS. Admin dapat menambahkan kriteria baru dengan memasukkan kode, nama kriteria, bobot, serta tipe kriteria (benefit atau cost). Selain itu, admin juga dapat mengedit dan menghapus 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



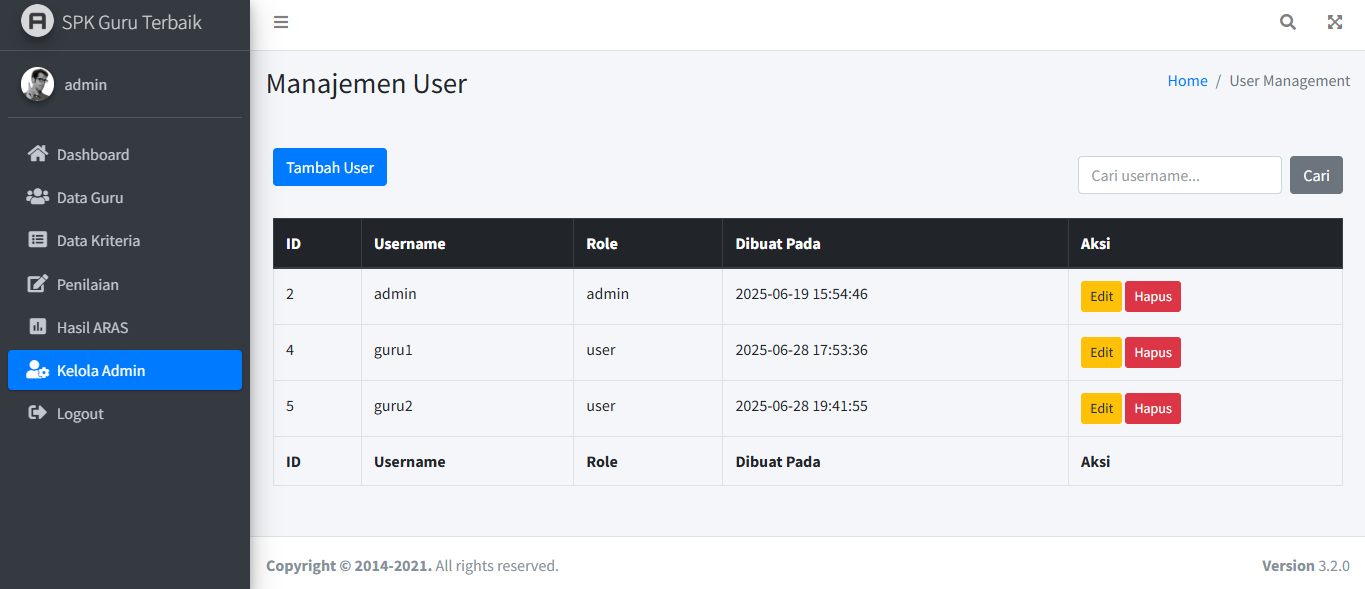
Halaman penilaian alternatif digunakan oleh admin untuk memberikan nilai terhadap setiap guru (alternatif) berdasarkan masing-masing kriteria yang telah ditentukan sebelumnya. Pada halaman ini, admin memilih guru, kemudian memasukkan nilai kriteria yang sesuai, baik berupa skor langsung maupun pemilihan 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**

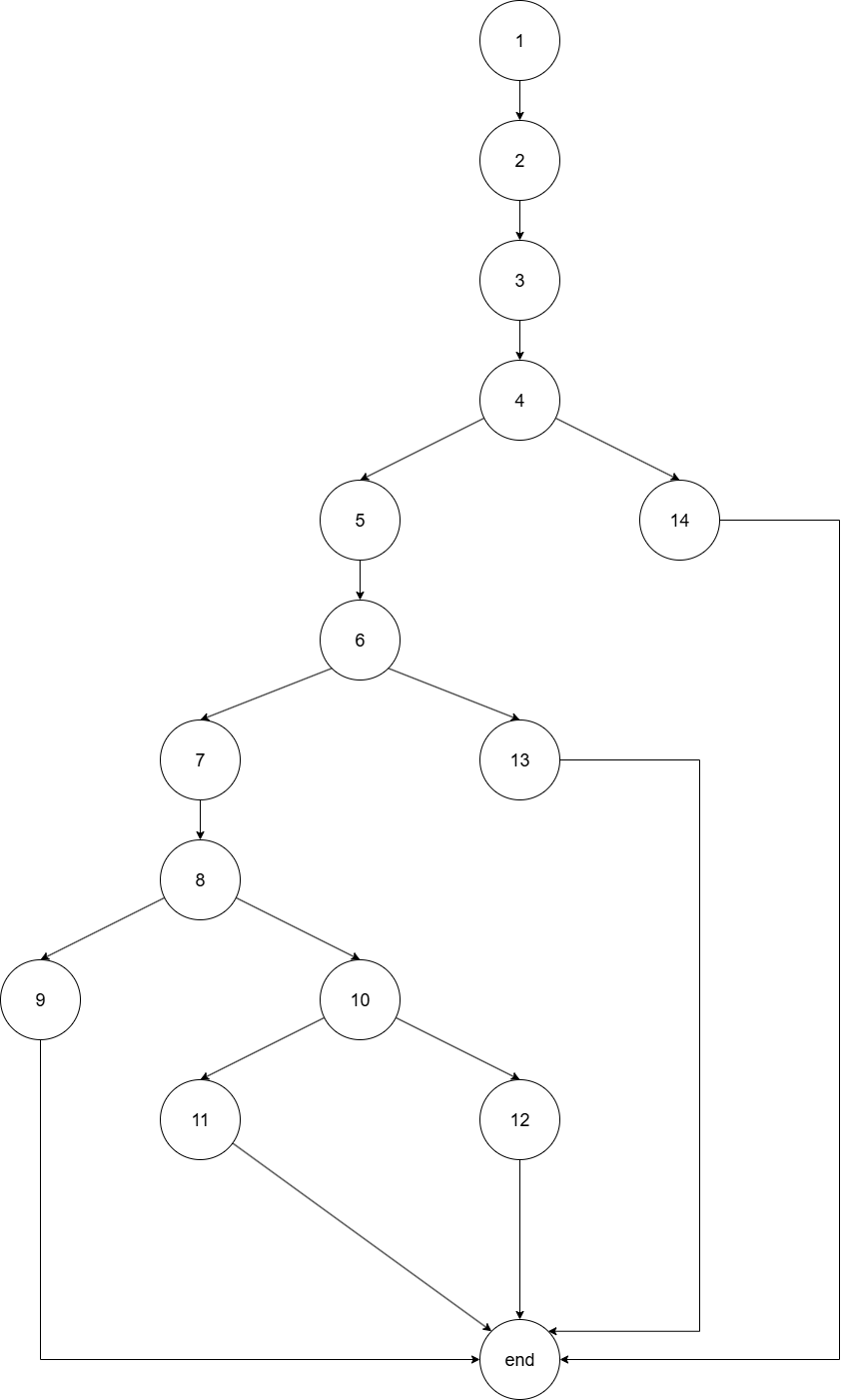
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Skenario Pengujian** | **Hasil yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| 1 | Logout Normal: Menekan tombol "Logout" pada sidebar saat sedang login. | Sistem mengakhiri sesi pengguna dan langsung mengarahkan (redirect) ke 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**

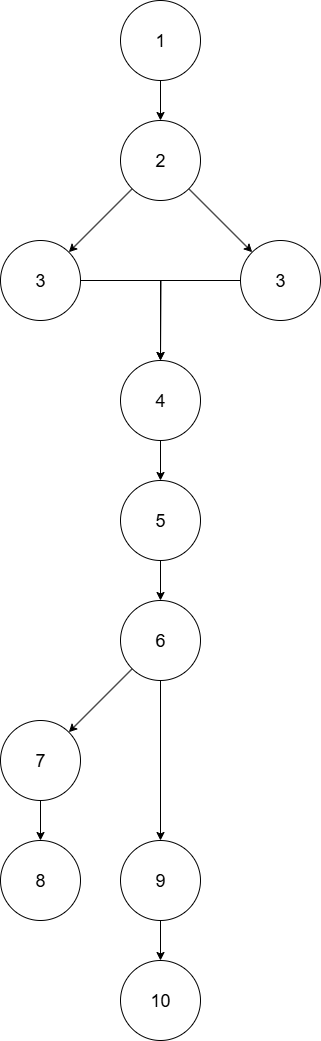
|  |  |
| --- | --- |
| **Node** | **Source Code** |
| 1 | session\_start();  include '../config/config.php';  include '../config/koneksi.php'; |
| 2 | if ($\_SERVER['REQUEST\_METHOD'] == 'POST') { |
| 3 | $username = $\_POST['username'];  $password = $\_POST['password'];  $sql = "SELECT \* FROM users 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**

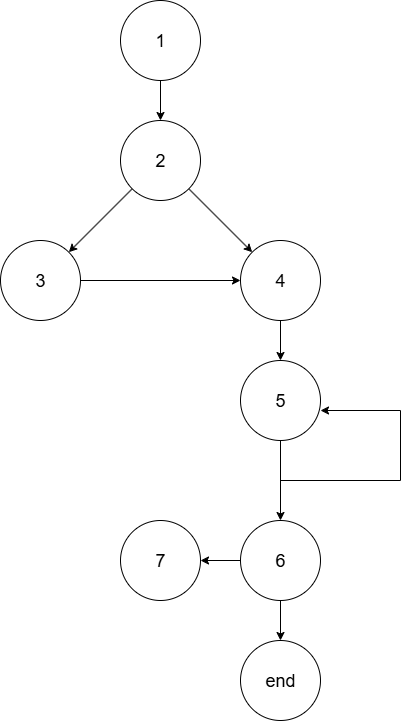
|  |  |
| --- | --- |
| **Node** | **Source Code** |
| 1 | <?php  $search = isset($\_GET['search']) ? mysqli\_real\_escape\_string($koneksidb, $\_GET['search']) : '';  $query = "SELECT a.\*, u.username  FROM alternatives a  JOIN users u ON a.user\_id = u.id"; |
| 2 | if (!empty($search)) {  $query .= " WHERE a.name LIKE '%$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**

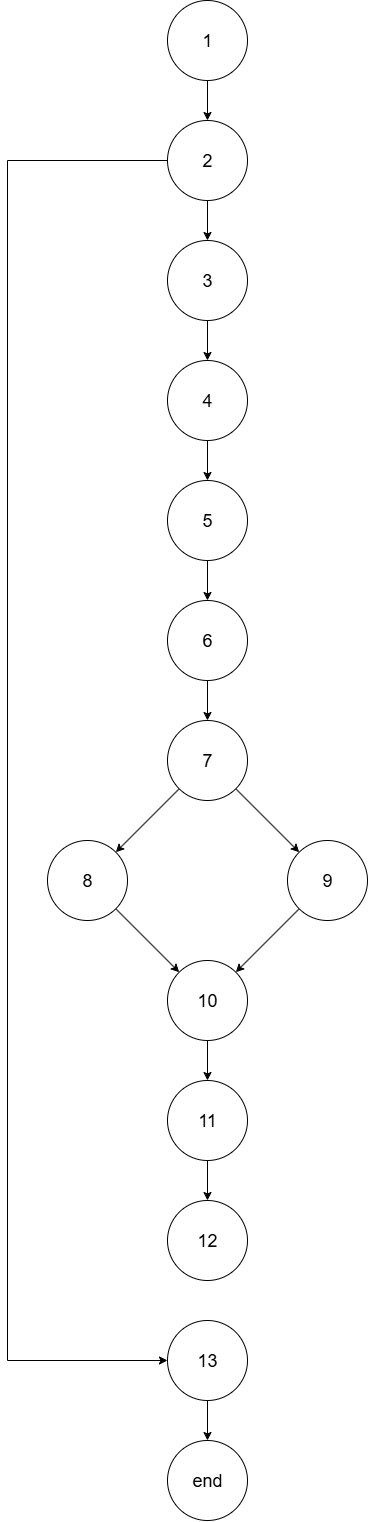
|  |  |
| --- | --- |
| **Node** | **Source Code** |
| 1 | <?php  $search = isset($\_GET['search']) ? mysqli\_real\_escape\_string($koneksidb, $\_GET['search']) : '';  $query = "SELECT \* FROM criteria"; |
| 2 | if (!empty($search)) { |
| 3 | $query .= " WHERE name LIKE '%$search%' OR code LIKE '%$search%'";  } |
| 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**

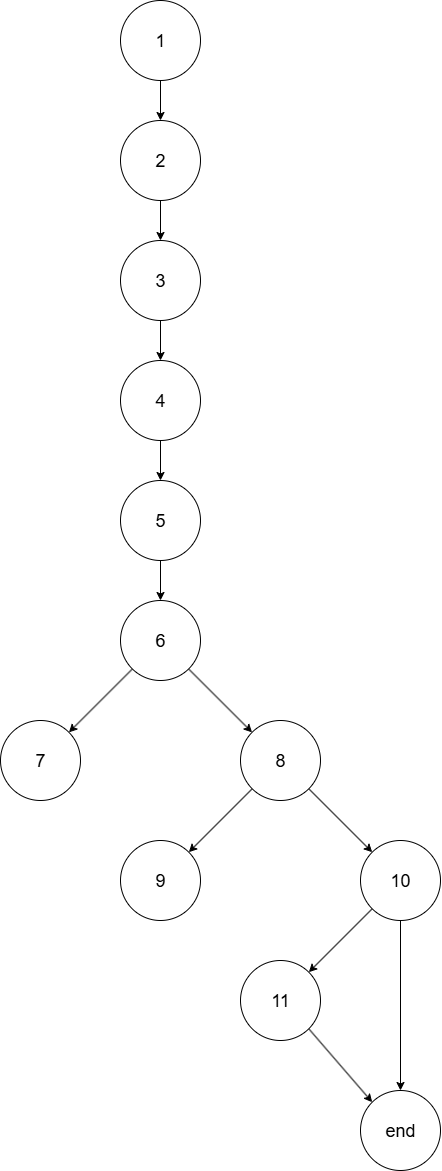
|  |  |
| --- | --- |
| **Node** | **Source Code** |
| 1 | <?php  $alternatif = mysqli\_query($koneksidb, "SELECT \* FROM alternatives ORDER BY code ASC");  $kriteria = mysqli\_query($koneksidb, "SELECT \* FROM criteria ORDER BY code ASC"); |
| 2 | if ($\_SERVER['REQUEST\_METHOD'] === 'POST') { |
| 3 | foreach ($\_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**

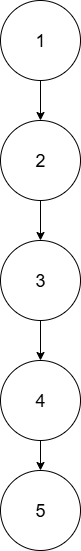
|  |  |
| --- | --- |
| **Node** | **Source Code** |
| 1 | <?php  include '../../../config/config.php';  include '../../../config/koneksi.php';  session\_start(); |
| 2 | cek\_admin(); |
| 3 | $page\_title = 'Manajemen User';  $breadcrumbs = [  ['title' => 'Home', 'link' => base\_url('dashboard/admin')],  ['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 |

# BAB V PENUTUP

## 5.1 Kesimpulan

## 5.2 Saran

# DAFTAR PUSTAKA

’Afiifah, K., Azzahra, Z. F., & Anggoro, A. D. (2022). Analisis Teknik Entity-Relationship Diagram dalam Perancangan Database Sebuah Literature Review. *Intech*, *3*(2), 18–22. https://doi.org/10.54895/intech.v3i2.1682

Aditiyawarman, D. (2016). IMPLEMENTASI PROBLEM BASED LEARNING UNTUK MENINGKATKAN KEMAMPUAN MAHASISWA DALAM. *Jurnal Informatika*, *3*(September), 277–289.

Alpandi, A. (2020). Sistem Pendukung Keputusan Penilaian Kinerja Dosen Dengan Menggunakan Metode Additive Ratio Assessment (Aras). *Simtek : Jurnal Sistem Informasi Dan Teknik Komputer*, *5*(1), 476–483. https://doi.org/10.51876/simtek.v5i1.69

Alti, D. D., Hasibuan, U. M., & Tusadiah, H. (2024). Analysis of the Benefits of Tutoring in Improving Student Learning Skills. *Cendekiawan : Jurnal Pendidikan Dan Studi Keislaman*, *3*(2), 429–433. https://doi.org/10.61253/cendekiawan.v3i2.236

Andoyo, A., Angraeni, E. Y., & Khumaidi, A. (2021). SISTEM PENDUKUNG KEPUTUSAN Konsep, Implementasi & Pengembangan. In *Penerbit Adab (CV. Adanu Abimata)*. https://books.google.co.id/books/about/SISTEM\_PENDUKUNG\_KEPUTUSAN\_Konsep\_Implem.html?id=YTgmEAAAQBAJ&redir\_esc=y

Damayanti. (2020). Sistem Pendukung Keputusan Penerimaan Member Merah Untuk Pedagang Yang Layak Pada Indogrosir Dengan Menggunakan Metode ARAS. *Seminar Nasional Teknologi Komputer & Sains (SAINTEKS)*, 453–458.

Destriana, R., Husain, S. M., Handayani, N., & Siswanto, A. T. P. (2022). Diagram UML Dalam Membuat Aplikasi Android Firebase “Studi Kasus Aplikasi Bank Sampah.” In *Deepublish*. https://books.google.co.id/books?id=6bM-EQAAQBAJ&printsec=frontcover#v=onepage&q&f=false

Geograf. (2023). *Pengertian Sistem: Definisi dan Penjelasan Lengkap Menurut Ahli - Geograf*. Geograf.Id. https://geograf.id/jelaskan/pengertian-sistem/

Habsy, B. A., Nurjanah, I., Putri, S. A., & Naisyla, A. Z. (2024). KONSEP DASAR PENDIDIKAN: MENUMBUHKAN PEMAHAMAN UNTUK MENCIPTAKAN PEMBELAJARAN YANG BERKUALITAS. *Jurnal Penelitian Guru Indonesia*, *4*(6), 4204–4227.

Hairani, L. (2021). APLIKASI SISTEM PENDUKUNG KEPUTUSAN REKOMENDASI PENGANGKATAN KARYAWAN TETAP MENGGUNAKAN METODE TOPSIS BERBASIS WEB. *Jurnal Informatika Dan Rekayasa Perangkat Lunak (JATIKA)*, *2*(2), 262–267.

Hanifah, U., Alit, R., & Sugiarto, S. (2016). Penggunaan Metode Black Box Pada Pengujian Sistem Informasi Surat Keluar Masuk. *Scan : Jurnal Teknologi Informasi Dan Komunikasi*, *11*(2). https://doi.org/10.33005/scan.v11i2.643

Hawari Nasution, M. A. Al, Siswanto, S., & Suryana, E. (2023). Rancangan Media Pembelajaran Berupa Aplikasi Augmented Reality Berbasis Android. *Jurnal Media Infotama*, *19*(2), 528–537. https://doi.org/10.37676/jmi.v19i2.4771

Hendra, H., & Riti, Y. F. (2023). Perancangan Dan Implementasi Website Dengan Konsep Ui/Ux Untuk Mengoptimalkan Marketing Perusahaan. *Jurnal Informatika Dan Teknik Elektro Terapan*, *11*(3 S1). https://doi.org/10.23960/jitet.v11i3s1.3430

Hendrawan, E., Meisel, M., & Sari, D. N. (2023). ANALYSIS AND IMPLEMENTATION OF COMPUTER NETWORK SYSTEMS USING SOFTWARE DRAW . IO. *Asia Information System Journal*, *2*(1), 9–15.

Hidayat, A., Yani, A., Rusidi, & Saadulloh. (2019). Membangun Website Sma Pgri Gunung Raya Ranau Menggunakan Php Dan Mysql. *JTIM: Jurnal Teknik Informatika Mahakarya*, *2*(2), 41–52.

Ihksan, M., Susilo, H., & Abdillah, N. (2023). BASIS DATA 2023: Konsep Dasar Membangun Database. In *CV. Suluah Kato Khatulistiwa*. https://books.google.co.id/books?id=UszMEAAAQBAJ&printsec=frontcover&hl=id#v=onepage&q&f=false

Joshi, K., Kumar, R., Kumar, A., Reshi, J., Sharma, A., & Dumka, A. (2022). A Framework Optimization in Social Media using Xampp: A Systematic Approach. *2022 International Conference on 4th Industrial Revolution Based Technology and Practices, ICFIRTP 2022*, *March 2023*, 1–4. https://doi.org/10.1109/ICFIRTP56122.2022.10059447

KBBI. (n.d.). *Hasil Pencarian - KBBI VI Daring*. Retrieved May 26, 2025, from https://kbbi.kemdikbud.go.id/entri/sekolah

Khomsi Pane, S. Y., Ramadhan, N. G., & Adhinata, F. D. (2022). Perancangan Basis Data Menggunakan Normalisasi Tabel Pada Perusahaan Dagang Barokah Abadi. *Journal of Dinda : Data Science, Information Technology, and Data Analytics*, *2*(2), 90–96. https://doi.org/10.20895/dinda.v2i2.563

Komarudin, M. (2016). Pengujian Perangkat Lunak Metode Black-Box Berbasis Equivalence Partitions pada Aplikasi Sistem Informasi di Sekolah. *Jurnal Mikrotik*, *06*(3), 02–16.

Manurian, W., Mubarok, I., Agustin, A. S., Haryanto, & Sania, N. (2020). Perancangan Sistem Informasi Pencatatan Poin Pelanggaran Tata Tertib Siswa Berbasis Website Pada SMK YP Karya 1 Tangerang. *Journal Informatics, Science & Technology (Online)*, *10*(1), 1–9.

Narulita, S., Nugroho, A., & Abdillah, M. Z. (2024). *Diagram Unified Modelling Language ( UML ) untuk Perancangan Sistem Informasi Manajemen Penelitian dan Pengabdian Masyarakat ( SIMLITABMAS ) Universitas Nasional Karangturi Semarang , Indonesia ( deskripsi ) dan perancangan sistem , khususnya pada pemrograman berorientasi objek ( Nistrina*. *3*, 244–256.

Pamungkas, D. W. L., & Rochimah, S. (2019). Pengujian Aplikasi Web - Tinjauan Pustaka Sistematis. *Jurnal IPTEK*, *23*(1), 17–24. https://doi.org/10.31284/j.iptek.2019.v23i1.459

Pangestu, B., Kosim, & Kosasih, A. (2022). Application of Additive Ratio Assessment (ARAS) Method for the Selection of Youth Red Cross Chairperson at SMA Negeri 1 Lebakwangi Kuningan. *Journal of General Education and Humanities*, *1*(2), 83–94. https://doi.org/10.58421/gehu.v1i2.18

Pradana, Y. I., Yetri, M., & Calam, A. (2024). Penerapan Metode Additive Ratio Assessment (ARAS) Dalam Pendukung Keputusan Recruitment Kepala Toko. *JURNAL SISTEM INFORMASI TGD*, *3*(4), 483–494.

Pradipta, R. A., Wintoro, P. B., Budiyanto, D., Studi, P., Informatika, T., & Lampung, U. (2022). PERANCANGAN PEMODELAN BASIS DATA SISTEM. *Jurnal Informatika Dan Teknik Elektro Terapan*, *10*(2).

Purnama, I., Ritonga, A. A., Pane, R., Bangun, B., & Pratama, Ri. S. (2020). Perancangan Sistem Informasi Data Bahan-Bahan Material Sinar, U D Sigambal, Baru. *Journal Computer Science and Information Technology(JCoInT)*, *1*(1), 1–7.

Ramdani, N. G., Fauziyyah, N., Fuadah, R., Rudiyono, S., Septiyaningrum, Y. A., Salamatussa‟adah, N., & Hayani, A. (2023). Definisi Dan Teori Pendekatan, Strategi, Dan Metode Pembelajaran. *Indonesian Journal of Elementary Education and Teaching Innovation*, *2*(1), 20–31.

Ramdany, S. W., Kaidar, S. A., Aguchino, B., Putri, C. A. A., & Anggie, R. (2024). Penerapan UML Class Diagram dalam Perancangan Sistem Informasi Perpustakaan Berbasis Web. *Journal of Industrial and Engineering System*, *5*(1), 30–41. https://doi.org/10.31599/2e9afp31

Sahir, S. H., & Panjaitan, S. A. (2019). Analisis Penerapan Metode Additive Ratio Assessment ( ARAS ) Pada Pemberian Insentif Sales Penjualan Guna Mendukung Keputusan Manajemen. *Seminar Nasional Teknologi Komputer & Sains (SAINTEKS) SAINTEKS*, 456–463.

Saputra, A. R., & Supriatin. (2022). Implementasi Algoritma ARAS Pada SPK Untuk Menentukan Peringkat Dosen Terbaik. *The Indonesian Journal of Computer Science*, *11*(2), 578–591. https://doi.org/10.33022/ijcs.v11i2.3057

Sarwandi, Sianturi, L. T., Hasibuan, N. A., Sudipa, I. G. I., Syahrizal, M., Alwendi, Mesran, Muqimuddin, Meilani, B. D., Ginanta, N. L. W. S. R., & Israwan, L. . F. (2023). Sistem Pendukung Keputusan. In *CV. Graha Mitra Edukasi*. https://books.google.co.id/books/about/Sistem\_Pendukung\_Keputusan.html?id=qmm-EAAAQBAJ&redir\_esc=y

Setiyani, H. (2022). Penerapan Metode Additive Ratio Assesment (ARAS) Pada Sistem Pemilihan Tempat Kursus Bahasa Inggris Online. *Jurnal Sistem Komputer Dan Informatika (JSON)*, *3*(4), 547–555. https://doi.org/10.30865/json.v3i4.4317

Shahu Gaikwad, S., & Adkar, P. (2019). A Review Paper on Bootstrap Framework. *IRE Journals*, *2*(10), 349–351. https://www.think247.com/vertical?s\_pt=sou

Siregar, Z., Erwina, P., & Munandar, M. H. (2021). Sistem Informasi Penyewaan Perumahan Mutiara Simpang Mangga Berbasis Web. *Journal of Student Development Information System (JoSDIS)*, *1*(1), 1–6.

Situmeang, D. S., Saripurna, D., & Siambaton, M. Z. (2022). Analisis Penilaian Kualitas Jenis Pelayanan Terbaik dengan Metode Aras pada Kantor Dinas Kependudukan dan Pencatatan Sipil. *Blend Sains Jurnal Teknik*, *1*(3), 171–185. https://doi.org/10.56211/blendsains.v1i3.138

Sulehu, M., & Handayani, L. (2019). Sistem Pendukung Keputusan Pemilihan Trainer Menggunakan Metode ARAS (Additive Ratio Assessment). *Seminar Nasional Teknologi …*, 431–441. http://prosiding.seminar-id.com/index.php/sainteks/article/view/186%0Ahttps://prosiding.seminar-id.com/index.php/sainteks/article/download/186/181

Suprapti, D., Kamisutara, M., & Artaya, P. (2017). Analisa Pengujian Sistem Informasi Penjualan. *Analisa Pengujian Sistem Informasi Penjualan Menggunakan Metode White Box*, 1–12. https://ojs.widyakartika.ac.id/index.php/sniter/article/view/37

Syarif, M., & Pratama, E. B. (2021). Analisis Metode Pengujian Perangkat Lunak Blackbox Testing Dan Pemodelan Diagram Uml Pada Aplikasi Veterinary Services Yang Dikembangkan Dengan Model Waterfall. *Jurnal Teknik Informatika Kaputama (JTIK)*, *5*(2), 253–258.

Tan, J., Chen, Y., & Jiao, S. (2024). Visual Studio Code in Introductory Computer Science Course: An Experience Report. In *ASEE Annual Conference and Exposition, Conference Proceedings* (Vol. 1, Issue 1). Association for Computing Machinery. https://doi.org/10.18260/1-2--48259

Tandon, J. K., & Sharma, S. (2022). Application Software : Boon For Teaching and Learning. *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)*, *2*(2), 169–175. https://doi.org/10.48175/IJARSCT-3634

Tarigan, C., Ginting, E. F., & Syahputra, R. (2022). Sistem Pendukung Keputusan Dalam Menentukan Kinerja Pengajar Dengan Metode Additive Ratio Assessment (ARAS). *J-SISKO TECH (Jurnal Teknologi Sistem Informasi Dan Sistem Komputer TGD)*, *5*(1), 16–24. https://doi.org/10.53513/jsk.v5i1.4245

Wahidin, M. A., Ariani, S., & Erwanto, M. (2024). *Journal of Computation Science And Artificial Intelligence MATERIAL FURNITURE PADA PT . CIREBON FURNITURE*. *1*(2), 48–56.

Wahyudi, A. D. (2024). Analisis Kepuasan Terhadap Pelayanan Supplier Menggunakan Metode A New Additive Ratio Assessment ( ARAS ). *Journal of Artifisial Intelligence and Tecnology Information (JAITI)*, *2*(1), 1–13.