# BAB III ANALISA DAN PERANCANGAN SISTEM

## 3.1 Analisa Sistem

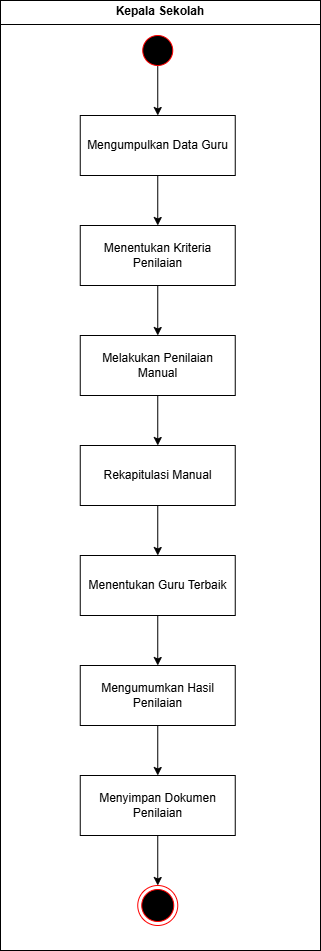
Analisa sistem merupakan tahap awal dalam proses pengembangan sistem yang bertujuan untuk memahami kondisi sistem saat ini dan menentukan kebutuhan sistem yang akan dibangun. Tahapan ini penting dilakukan agar perancangan sistem dapat dilakukan secara tepat sasaran dan sesuai dengan permasalahan yang 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.

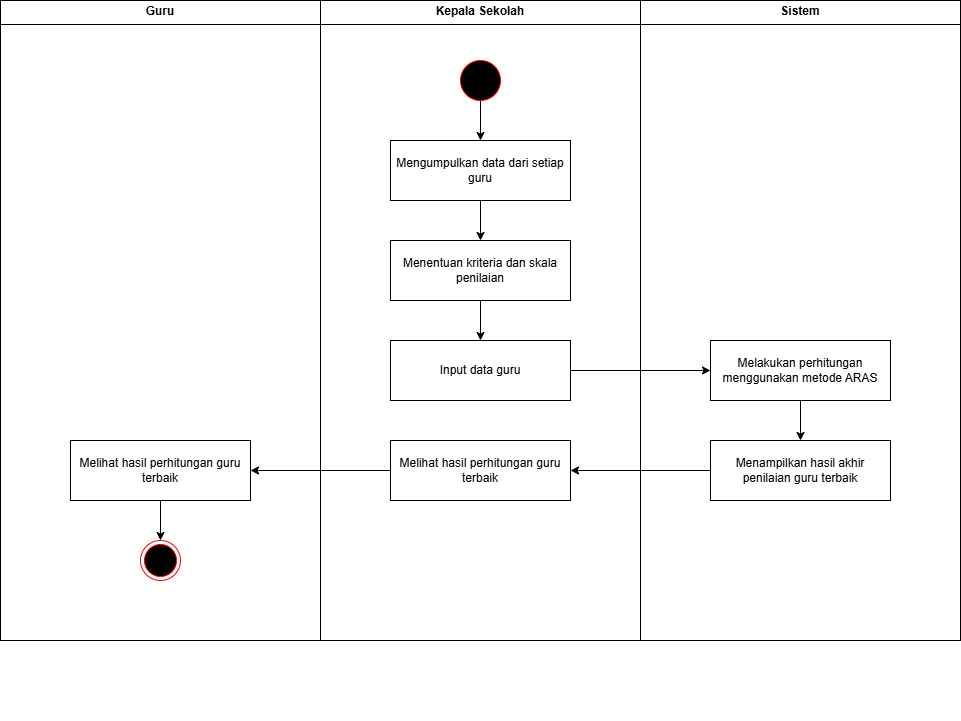


Namun, metode penilaian manual ini memiliki beberapa kekurangan, seperti potensi subjektivitas, rawan kesalahan perhitungan, dan kurang efisien dalam pengumpulan serta pengolahan data. Selain itu, tidak adanya sistem terintegrasi membuat proses evaluasi sulit untuk dilakukan secara konsisten dan terdokumentasi 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.

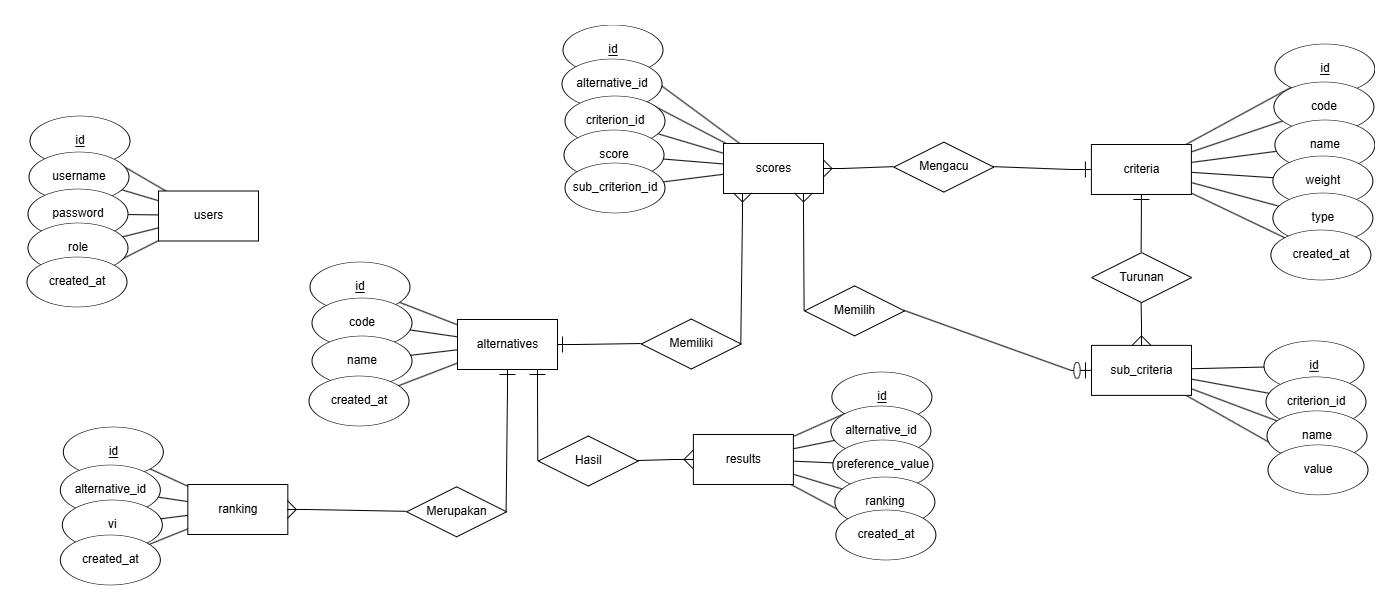
### 3.1.3 Penerapan Perhitungan Motode ARAS(BLM DAPET DATA)

## 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:

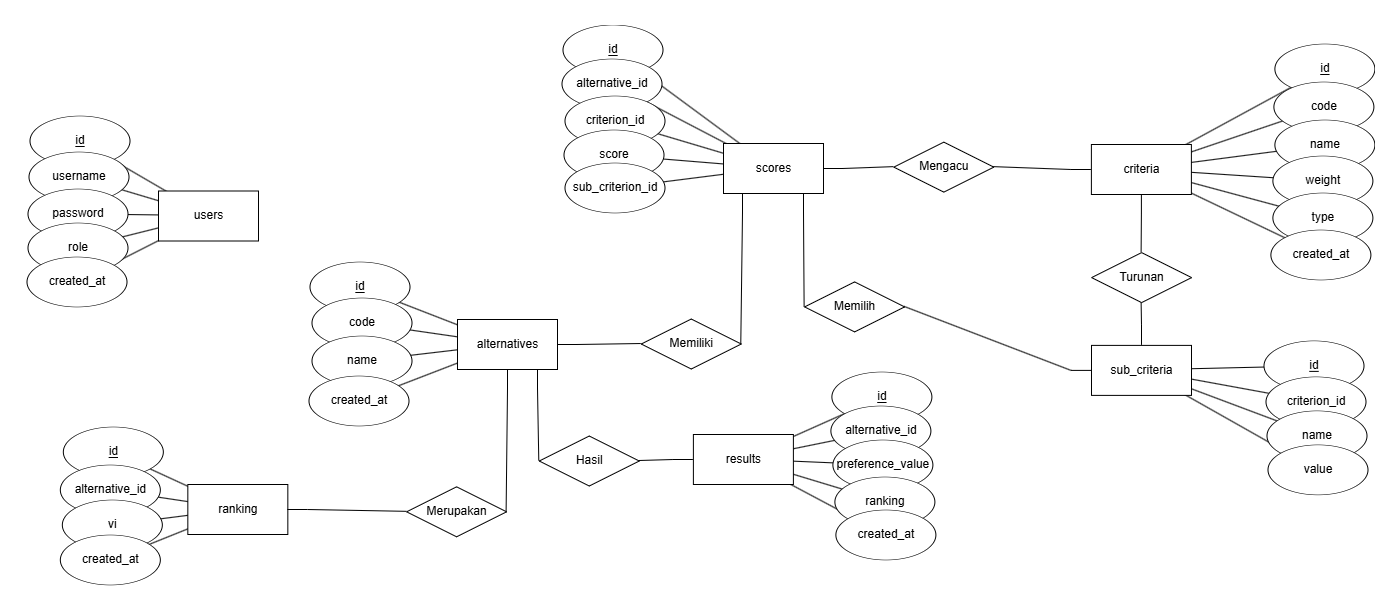


Pada ERD tersebut, peneliti merancang dan memiliki 7 entitas utama yang saling berelasi dalam mendukung proses pemilihan guru terbaik menggunakan metode ARAS. Entitas users berfungsi sebagai data pengguna yang memiliki hak akses ke dalam sistem, seperti admin atau kepala sekolah, yang nantinya melakukan 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

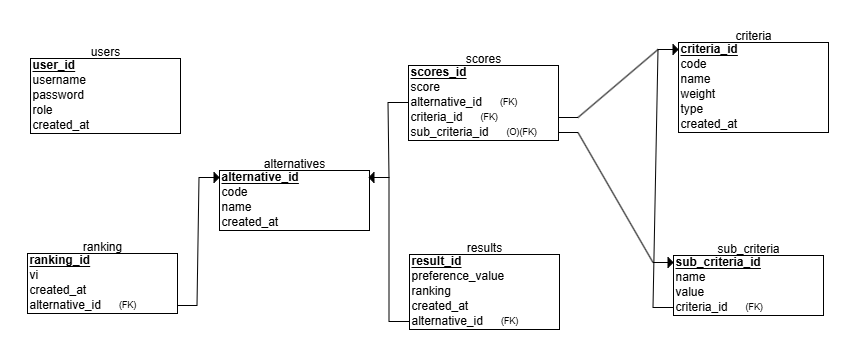
1

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

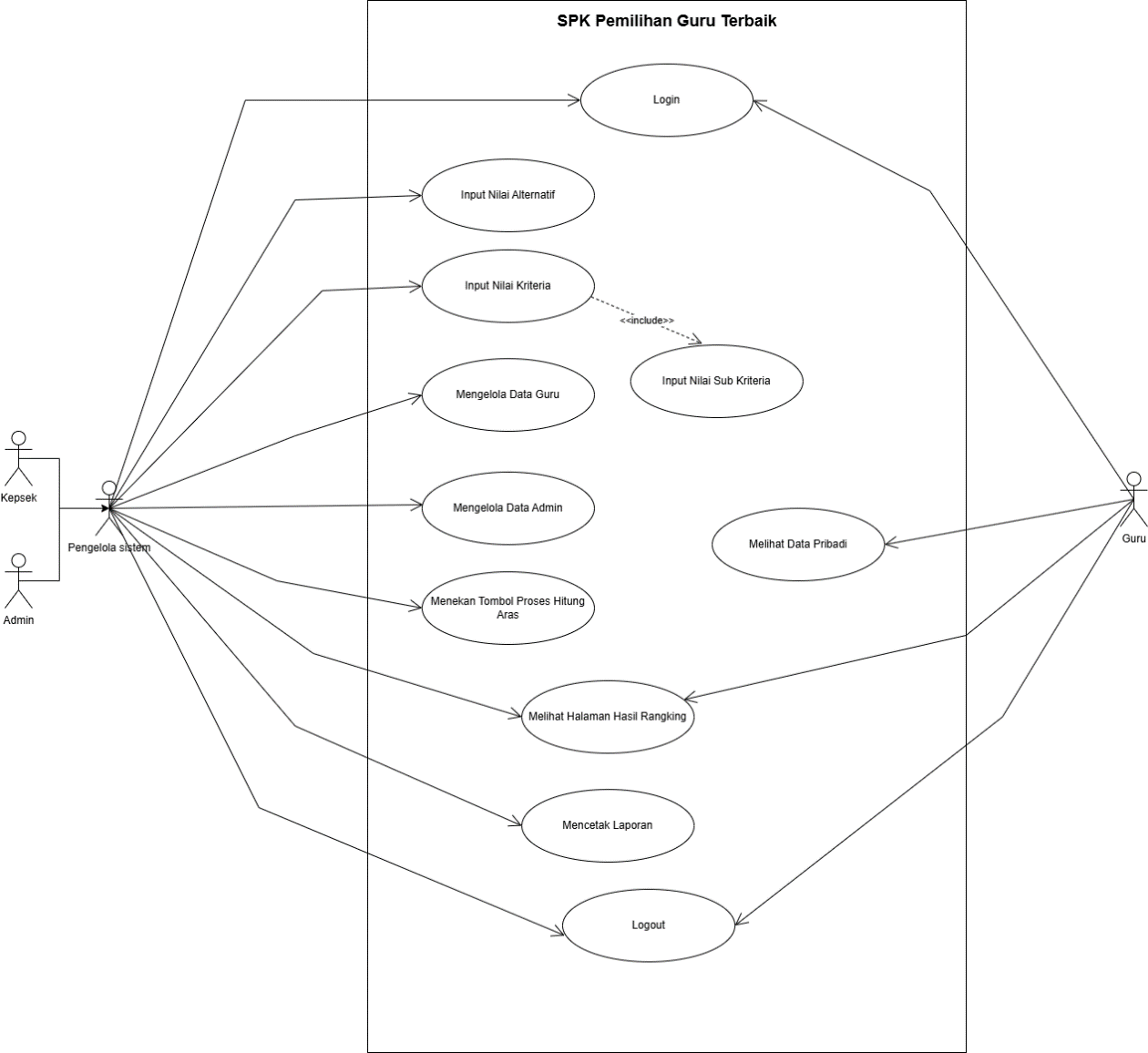
|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Keterangan** |
| id | Int | 11 | Menyimpan Auto Increment id |
| alternative\_id | Int | 11 | Relasi ke alternatives.id |
| preference\_value | Decimal | 10,4 | Nilai preferensi akhir |
| ranking | Int | 11 | Peringkat hasil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

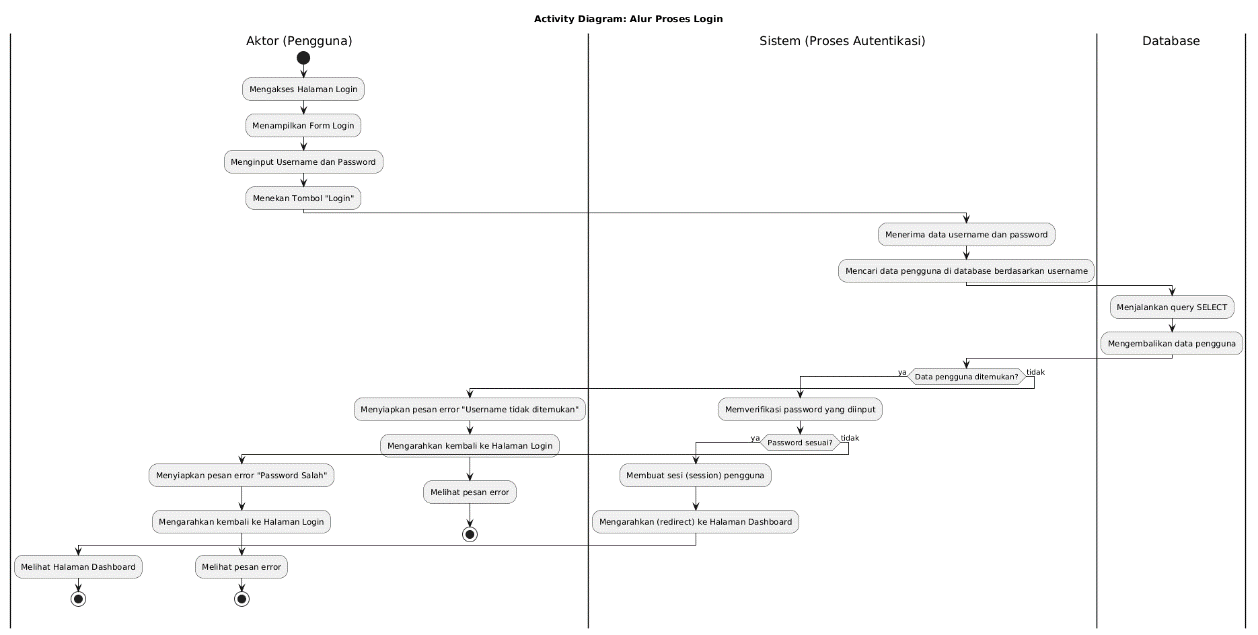


Diagram aktivitas proses login menjelaskan alur kerja pengguna saat memasuki sistem. Proses dimulai ketika pengguna mengakses halaman login, menginput username dan password, lalu menekan tombol login. Sistem kemudian memvalidasi kredensial tersebut dengan data yang tersimpan di database. Jika validasi 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**

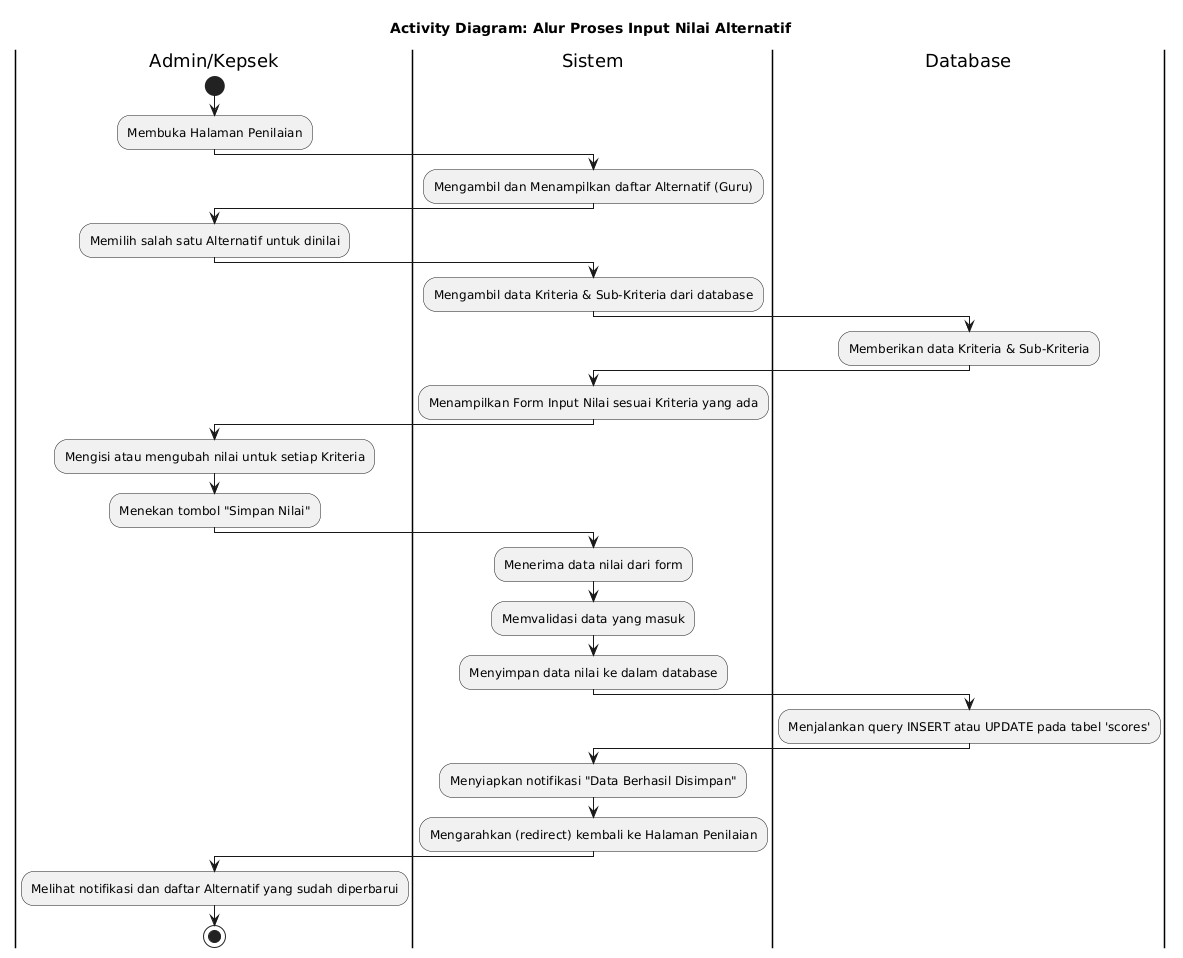
****

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**

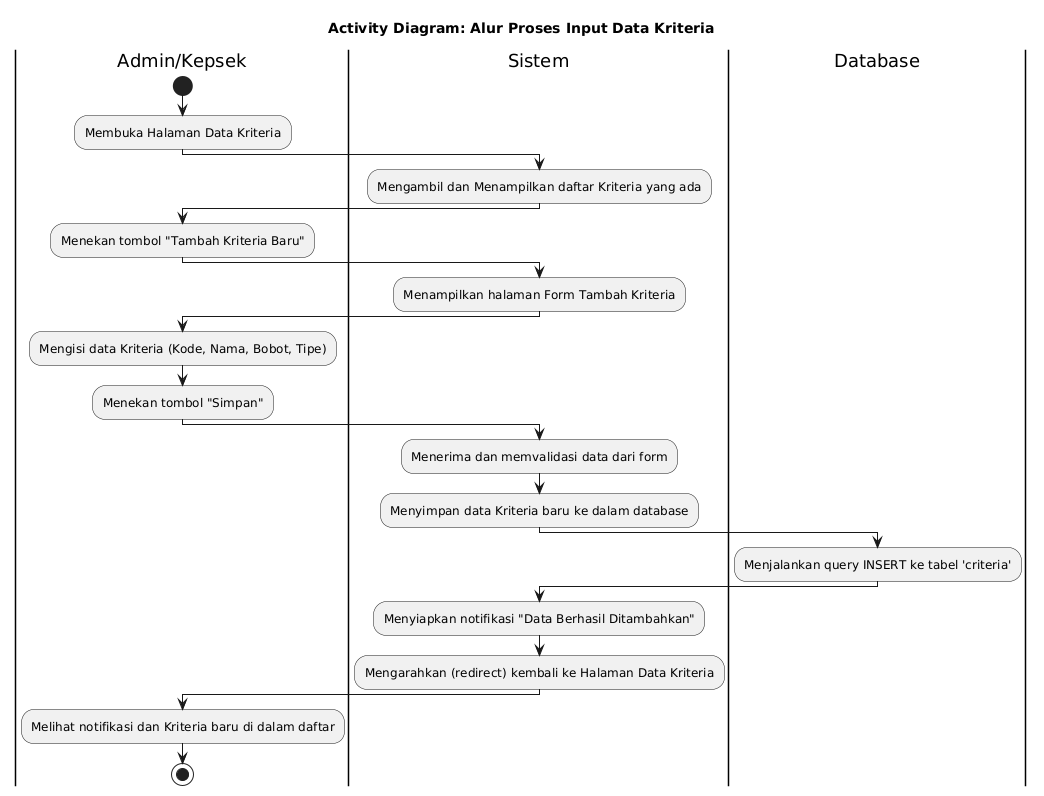


Diagram aktivitas proses input data kriteria merinci langkah-langkah untuk menambahkan kriteria penilaian baru ke dalam sistem. Proses ini diinisiasi oleh Aktor yang memilih menu untuk menambah kriteria baru, yang kemudian akan diarahkan oleh sistem ke halaman form input. Aktor mengisi seluruh atribut 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**

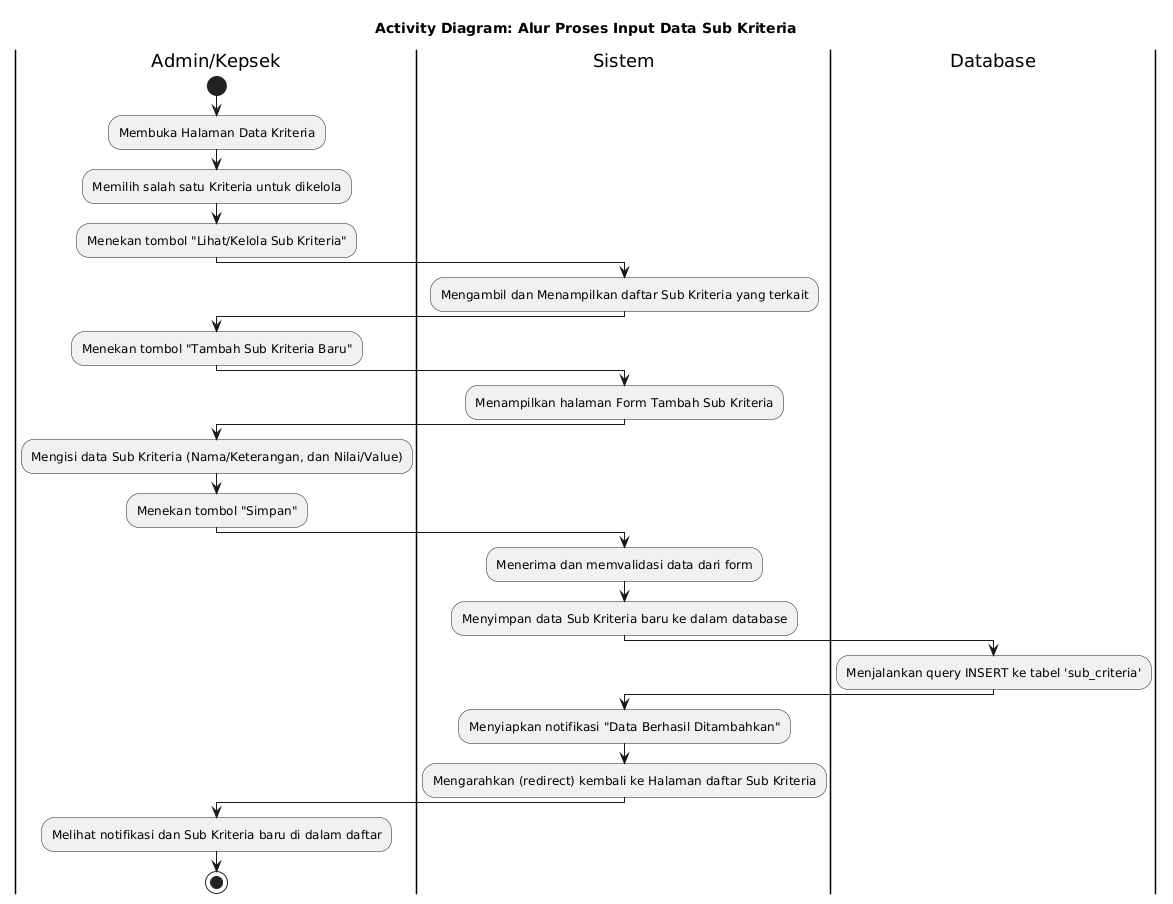
****

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**

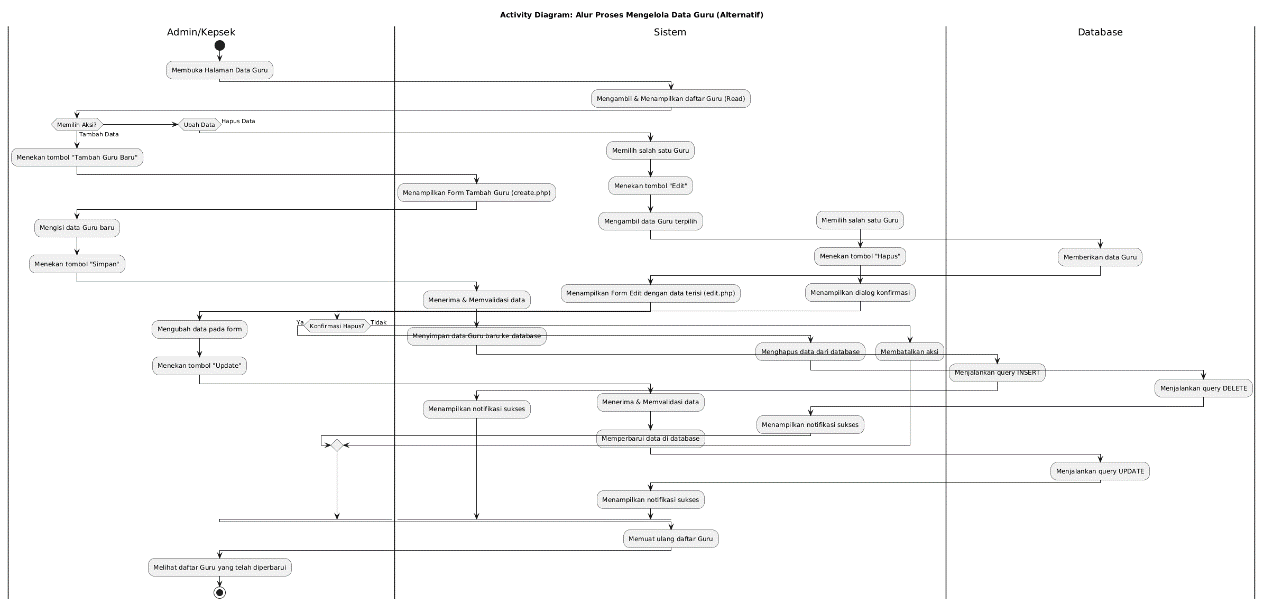
****

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**

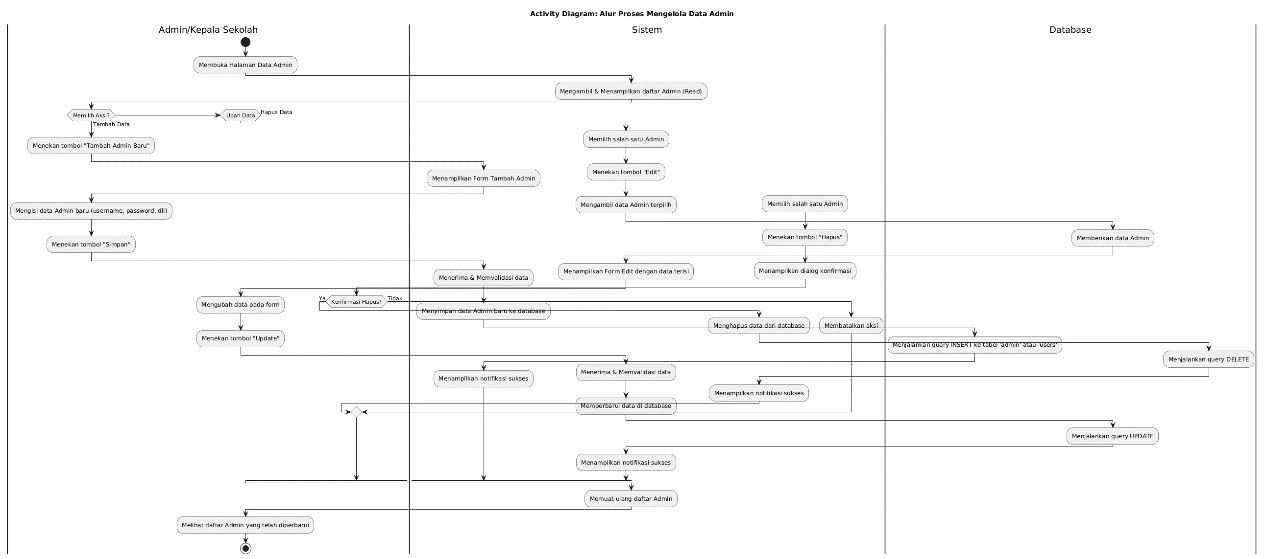
****

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.

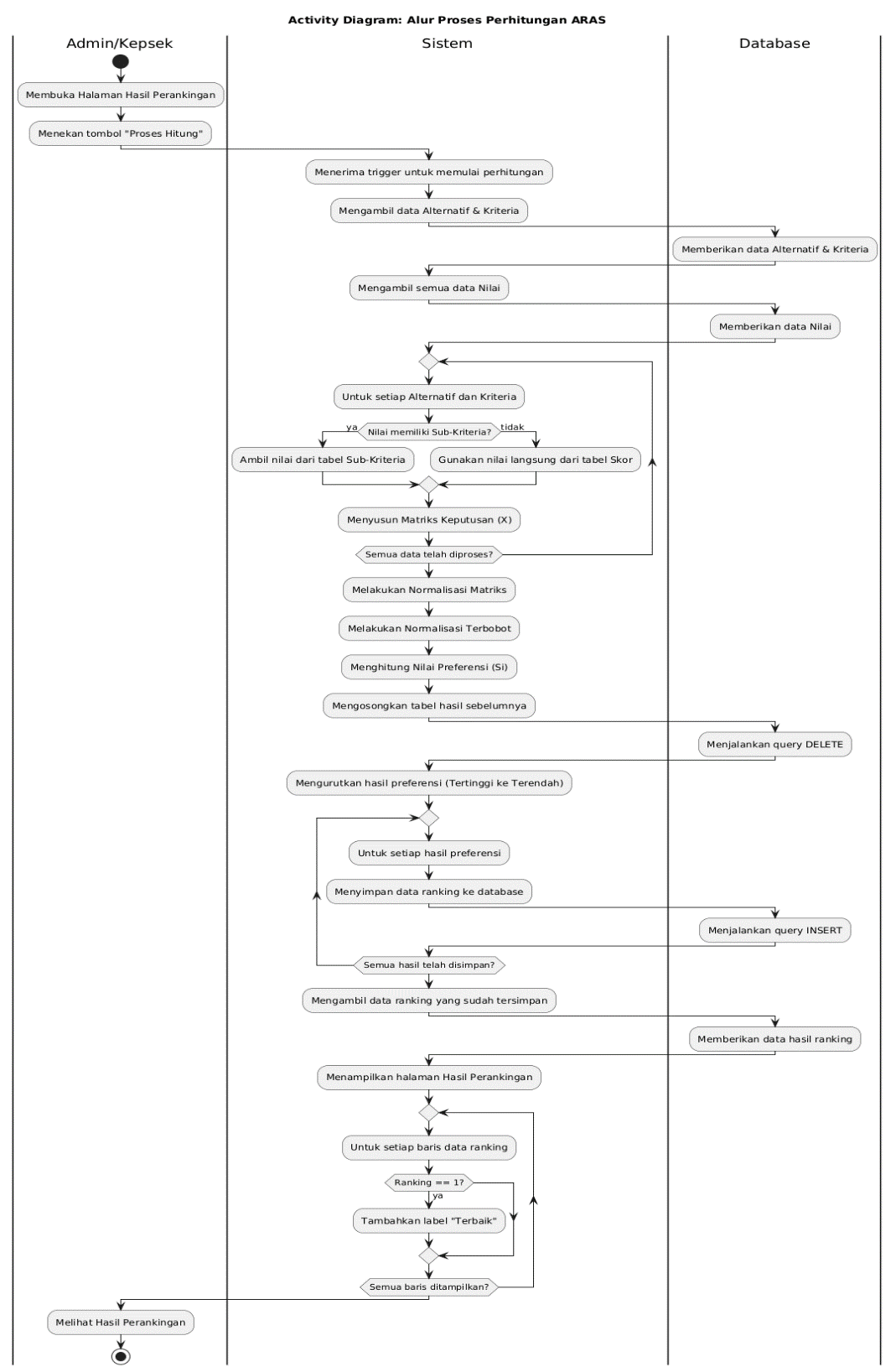
1. **Activity Diagram Proses Perhitungan ARAS**

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**

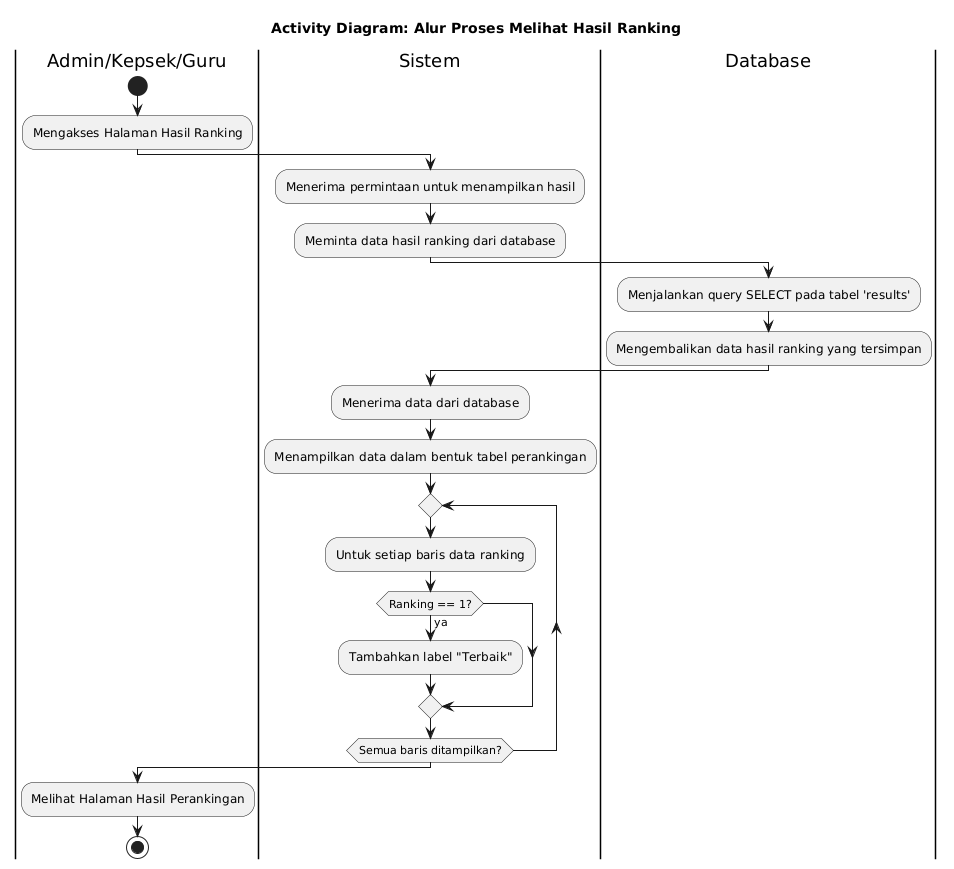


Diagram aktivitas proses melihat hasil ranking menggambarkan alur kerja sederhana saat pengguna ingin melihat data peringkat yang sudah tersimpan tanpa memicu perhitungan ulang. Alur dimulai saat Aktor (Admin, Kepala Sekolah, atau Guru) mengakses halaman hasil ranking. Sistem kemudian akan langsung mengirimkan 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.

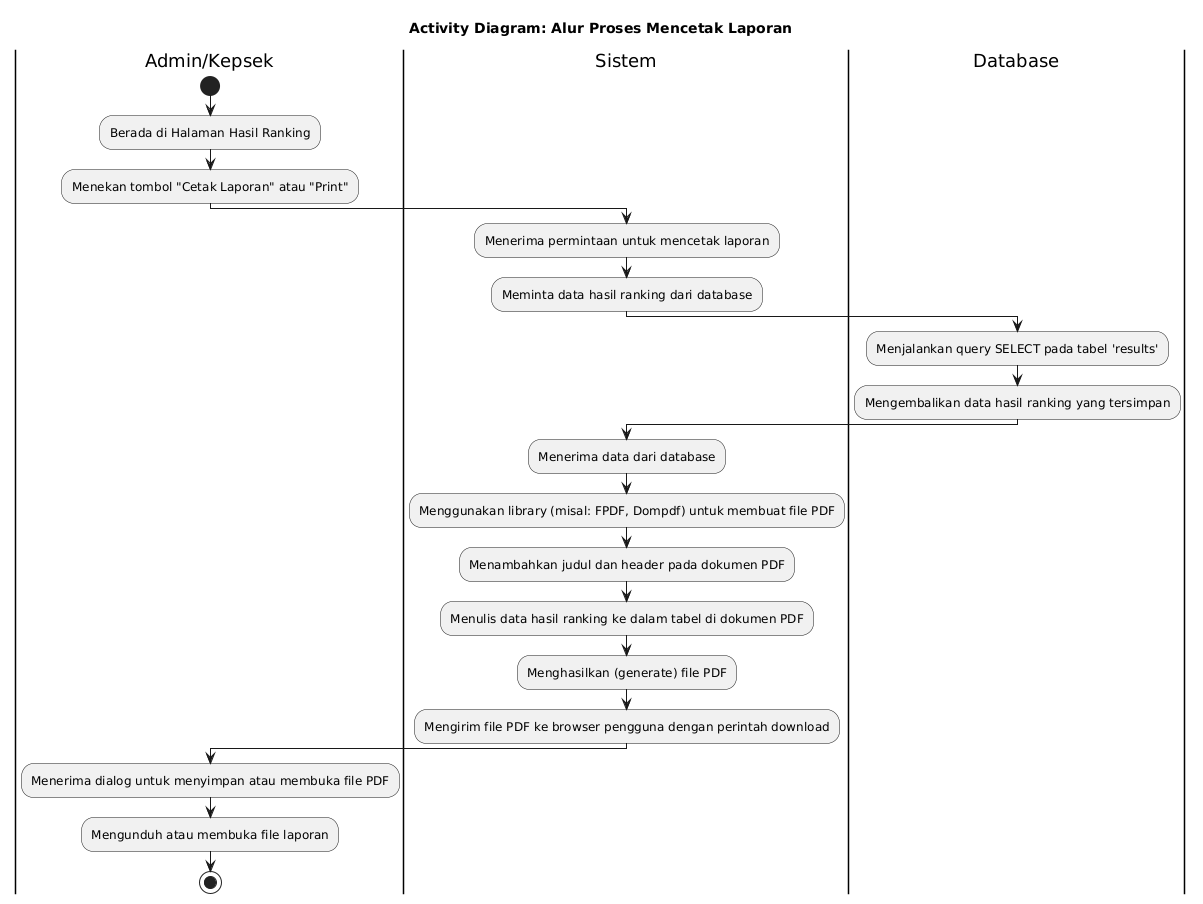
1. **Activity Diagram Mencetak Laporan**

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.

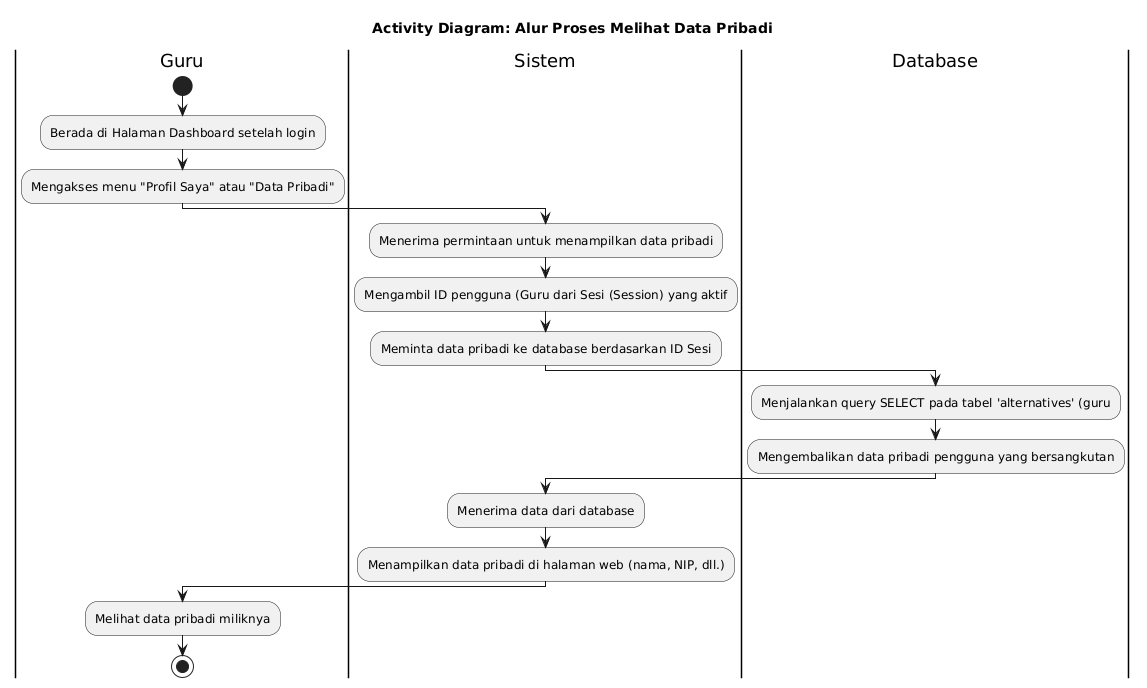
1. **Activity Diagram Melihat Data Pribadi**

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.

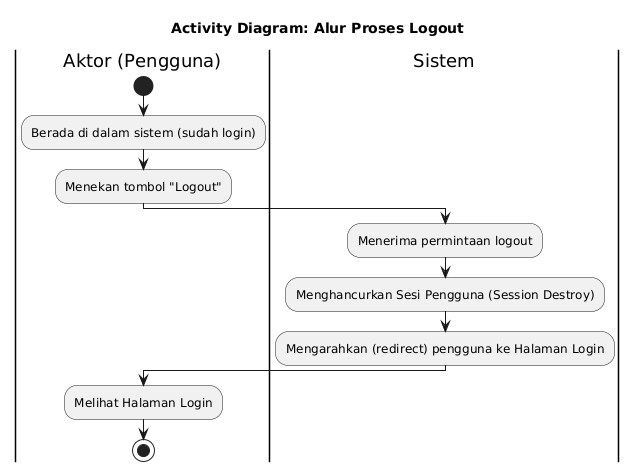
1. **Activity Diagram Logout**

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.

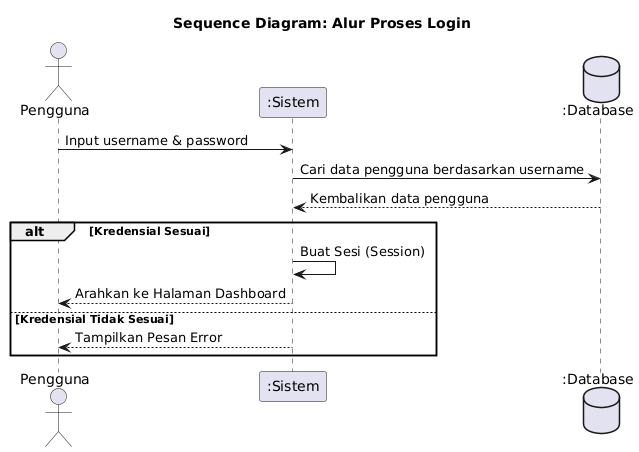
1. **Sequence** **Diagram Login**

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.

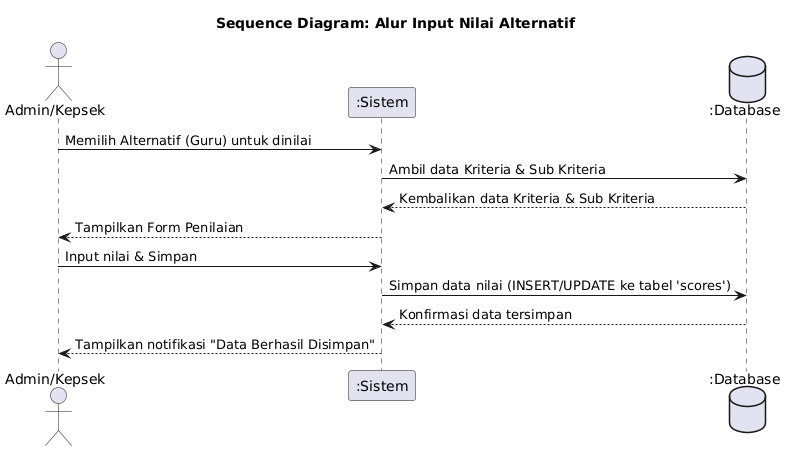
1. **Sequence Diagram Input Nilai Alternatif**

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**

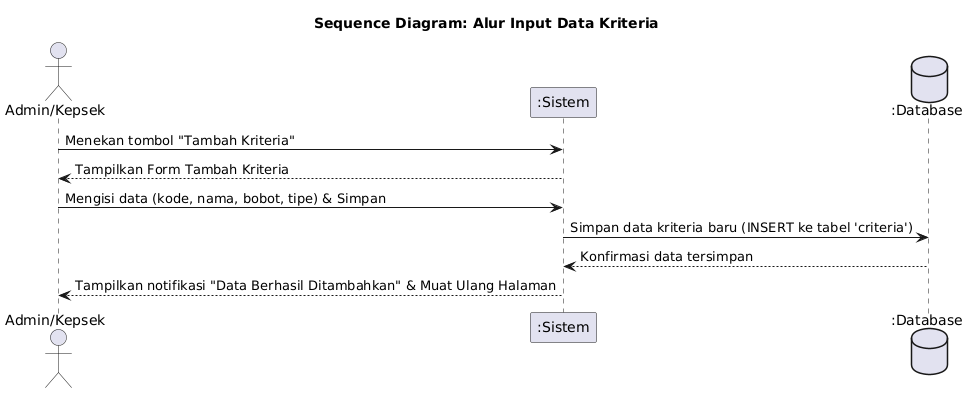


Diagram ini menunjukkan alur pembuatan data kriteria baru. Aktor memulai interaksi dengan meminta Sistem untuk menampilkan form tambah kriteria. Setelah Aktor mengisi dan mengirimkan data, Sistem akan meneruskannya ke Database untuk disimpan menggunakan perintah INSERT. Sebagai respons akhir, Sistem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

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

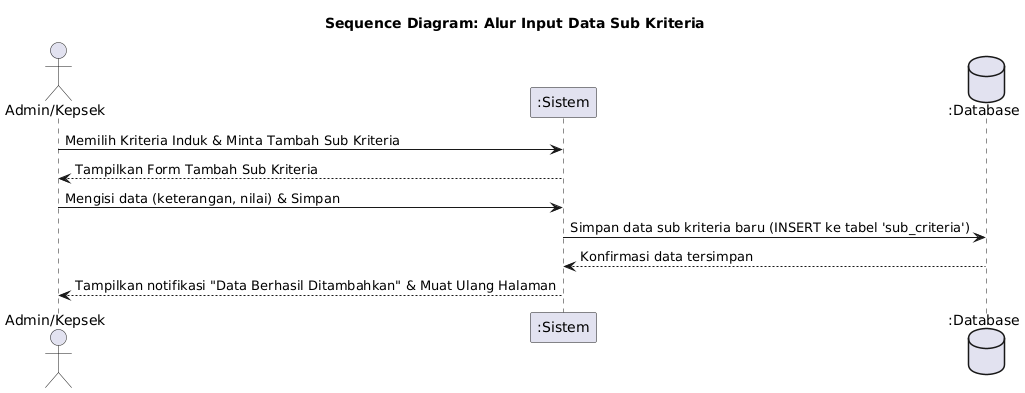


Diagram ini merinci alur pembuatan data sub kriteria yang terikat pada sebuah kriteria induk. Aktor memilih kriteria induk dan meminta untuk menambah sub kriteria, yang direspons oleh Sistem dengan menampilkan form. Setelah data diisi dan disimpan, Sistem akan menyimpannya ke Database dengan relasi ke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

1. **Sequence Diagram Mengelola Data Guru**

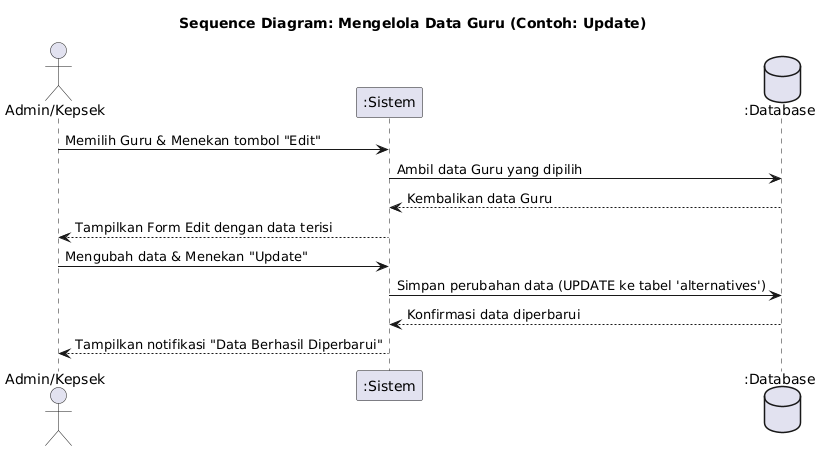


Diagram ini menggambarkan interaksi untuk salah satu proses pengelolaan data, yaitu mengubah data guru. Aktor memilih data guru untuk diedit, lalu Sistem mengambil data tersebut dari Database untuk ditampilkan di form. Setelah Aktor melakukan perubahan dan menyimpan, Sistem akan mengirimkan data baru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

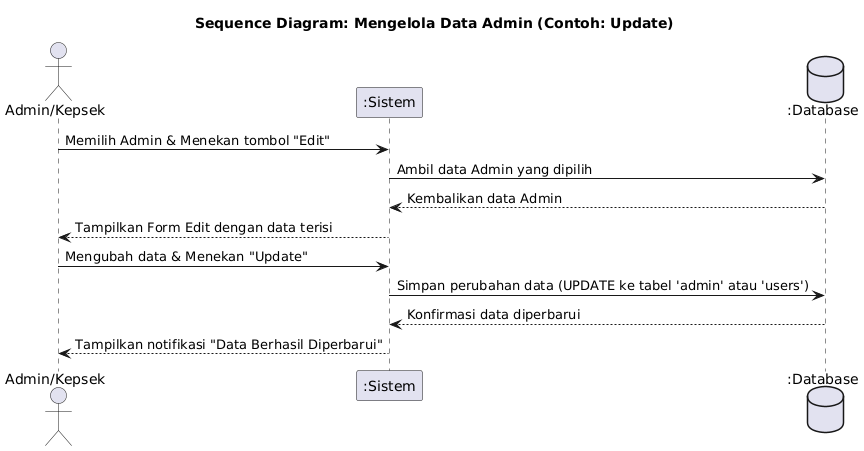
1. **Mengelola Data Admin**

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.

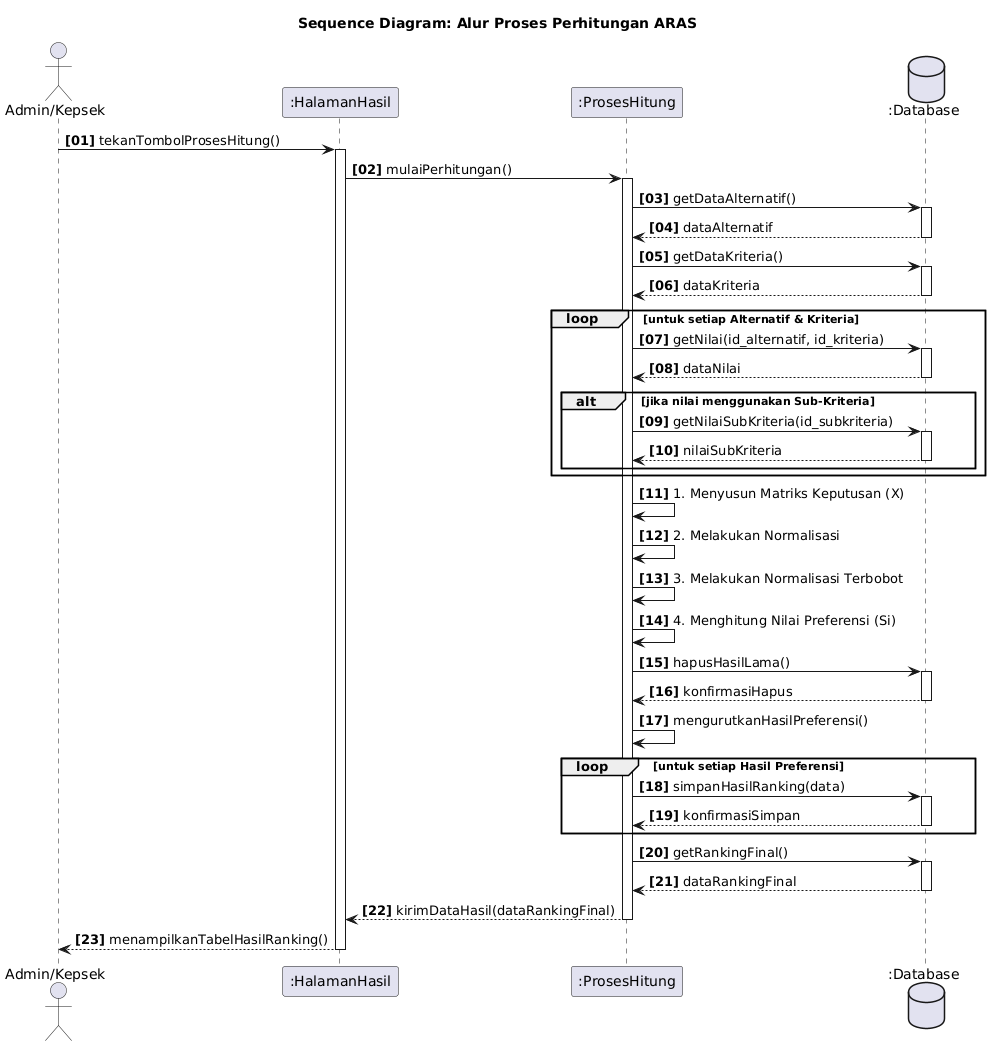
1. **Sequence Diagram Hitung Aras**

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.

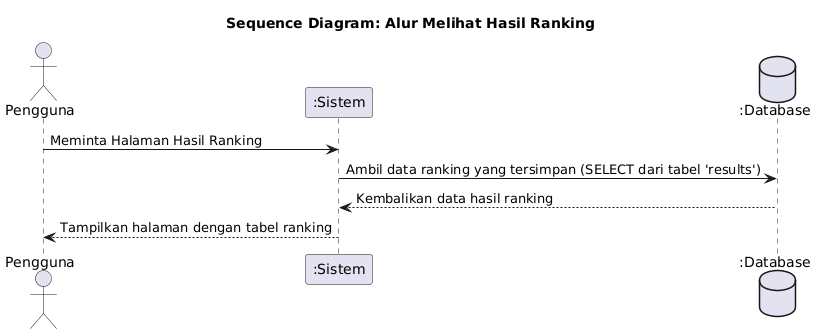
1. **Sequence Diagram Melihat Hasil Ranking**

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**

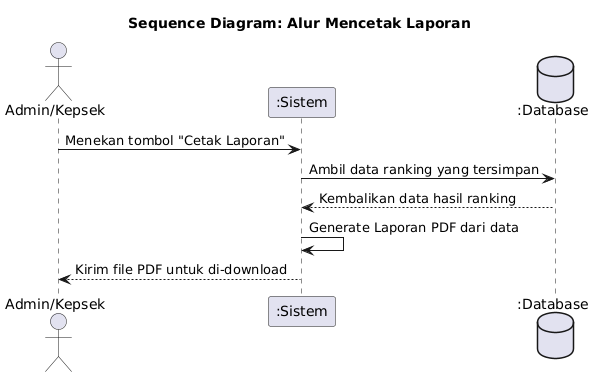


Diagram ini menunjukkan alur saat Aktor meminta sistem untuk membuat file laporan. Aktor memicu aksi cetak pada Sistem, yang kemudian mengambil data ranking terbaru dari Database. Sistem melakukan proses internal untuk menghasilkan (generate) dokumen PDF dari data tersebut, lalu mengirimkan file PDF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

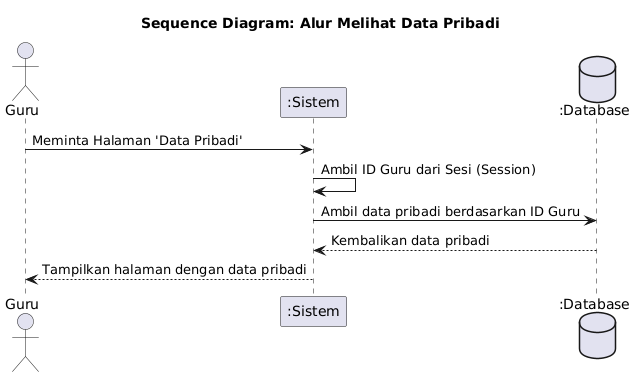
1. **Sequence Diagram Melihat Data Pribadi**

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.

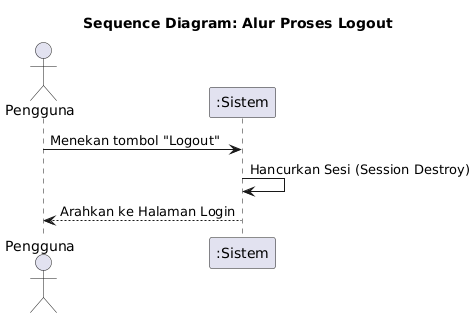
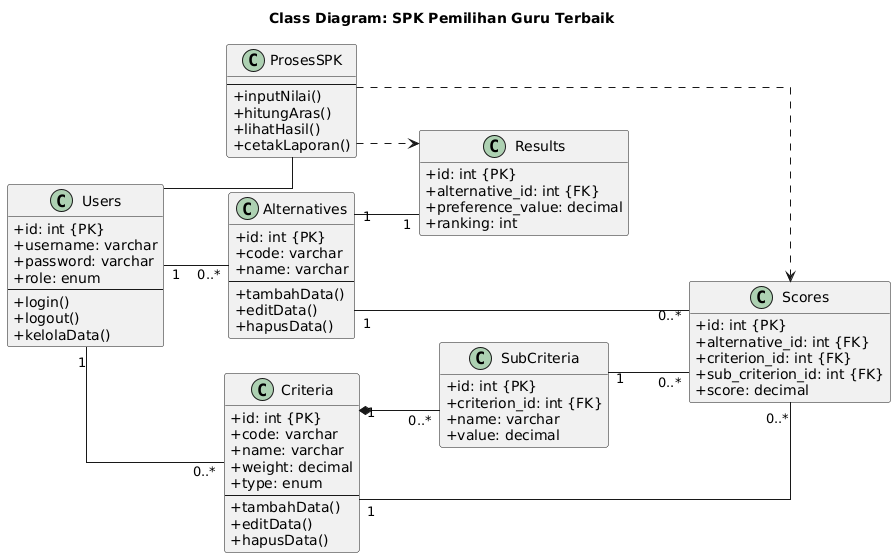
1. **Sequence Diagram Logout**

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.

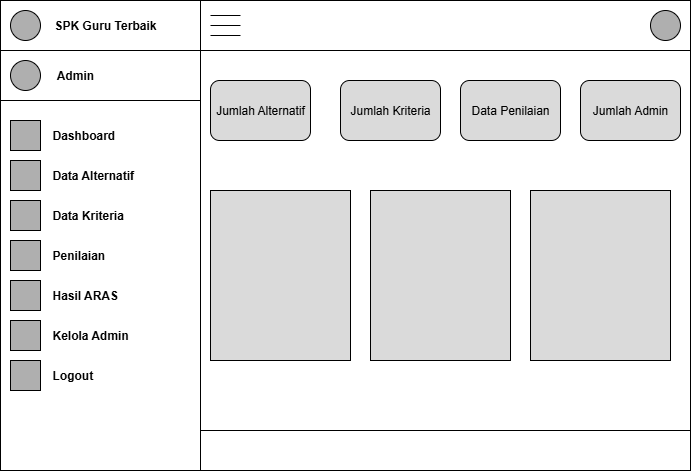
## 3.4 User Interface

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**

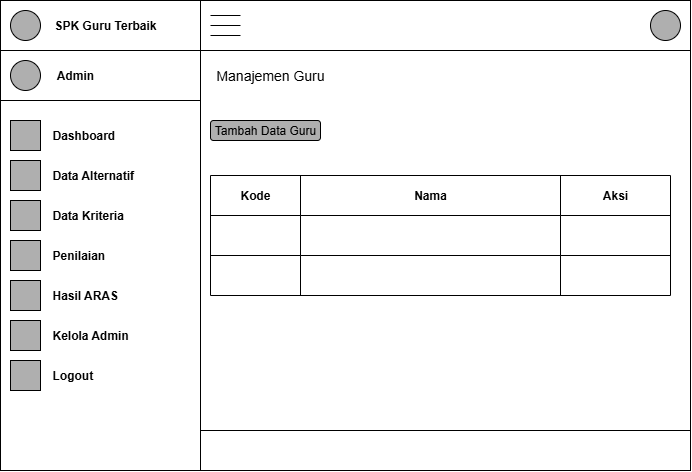
Tampilan ini adalah **gerbang utama** sistem, berfungsi sebagai halaman otentikasi. Pengguna diwajibkan untuk memasukkan **username** dan **password** yang valid untuk dapat mengakses fitur-fitur di dalam sistem. Desainnya yang minimalis memastikan pengguna dapat fokus untuk masuk ke dalam aplikasi dengan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

1. **Dashboard**

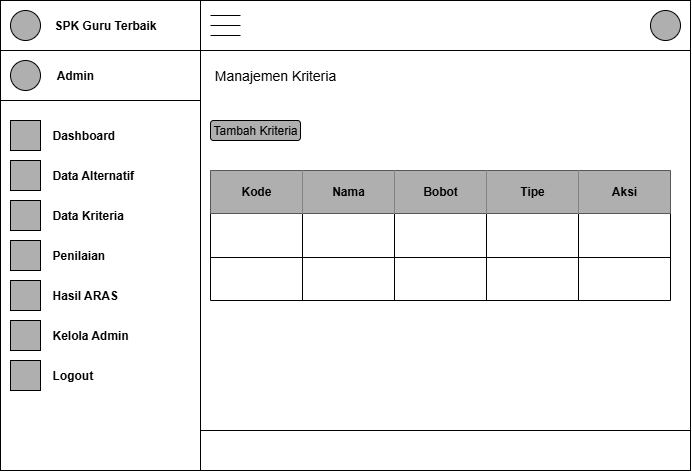


Ini adalah **halaman pertama** yang dilihat pengguna setelah berhasil login. Dashboard menyajikan **ringkasan informasi penting** dalam bentuk kartu (*cards*), seperti jumlah total alternatif (guru), kriteria, dan admin. Halaman ini juga berfungsi sebagai **pusat navigasi**, di mana pengguna dapat mengakses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

1. **Data Alternatif**

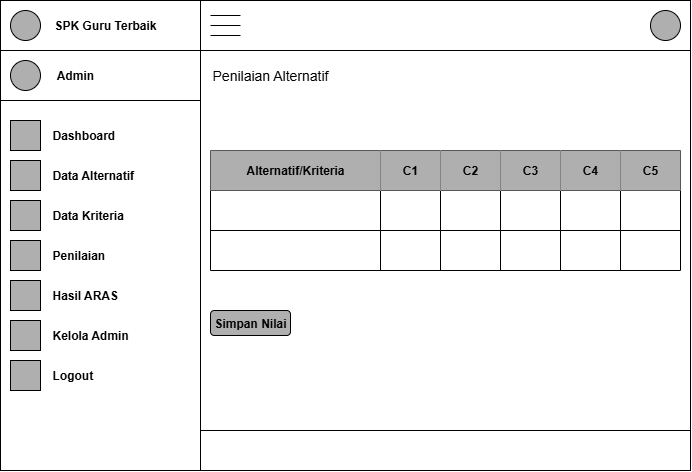


Halaman ini merupakan antarmuka untuk **mengelola data master guru** atau alternatif. Admin dapat melihat daftar semua guru yang terdaftar dalam sebuah tabel. Terdapat juga fungsionalitas untuk **menambah data guru baru** melalui tombol yang tersedia, serta melakukan aksi **edit** atau **hapus** pada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

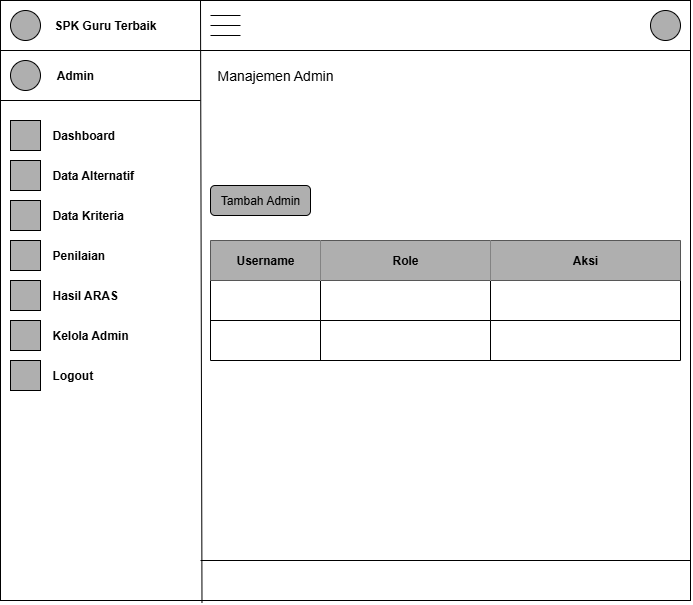
1. **Data Kriteria**

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.

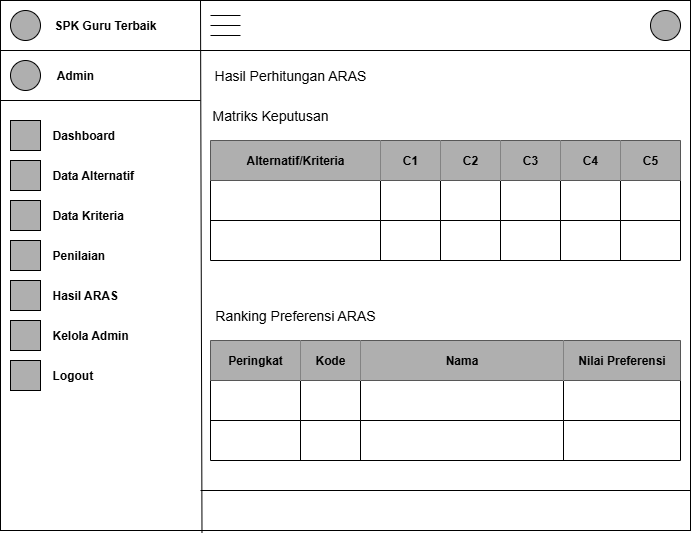
1. **Penilaian Alternatif**



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.