**ANALISA PERBANDINGAN METODE SAW DAN SMART UNTUK SISTEM PENDUKUNG KEPUTUSAN UJIAN KENAIKAN TINGKAT ATLET PANAHAN**

**SKRIPSI**



**Oleh:**

**DESI RATNA NINGSIH**

**1410530187**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER**

**(STMIK) BUMIGORA**

**MATARAM**

**2018**

**ANALISA PERBANDINGAN METODE SAW DAN SMART UNTUK SISTEM PENDUKUNG KEPUTUSAN UJIAN KENAIKAN TINGKAT ATLET PANAHAN**

**SKRIPSI**



Diajukan Sebagai Salah Satu Syarat untuk Memenuhi Kebulatan Studi

Jenjang Strata Satu (S1) Program Studi Teknik Informatika

Pada Sekolah Tinggi Manajemen Informatika dan Komputer

(STMIK) Bumigora Mataram

**Oleh:**

**DESI RATNA NINGSIH**

**1410530187**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER**

**(STMIK) BUMIGORA**

**MATARAM**

**2018**

**ANALISA PERBANDINGAN METODE SAW DAN SMART UNTUK SISTEM PENDUKUNG KEPUTUSAN UJIAN KENAIKAN TINGKAT ATLET PANAHAN**

**SKRIPSI**

Diajukan Sebagai Salah Satu Syarat untuk Memenuhi Kebulatan Studi

Jenjang Strata Satu (S1) Program Studi Teknik Informatika

Pada Sekolah Tinggi Manajemen Informatika dan Komputer

(STMIK) Bumigora Mataram

**Oleh:**

**DESI RATNA NINGSIH**

**1410530187**

**Ketua,**

**Heroe Santoso, M.Kom**

**NIK. 96.6.64**

**ANALISA PERBANDINGAN METODE SAW DAN SMART UNTUK SISTEM PENDUKUNG KEPUTUSAN UJIAN KENAIKAN TINGKAT ATLET PANAHAN**

**LEMBAR PENGESAHAN PENGUJI**

Diajukan Sebagai Salah Satu Syarat untuk Memenuhi Kebulatan Studi

Jenjang Strata Satu (S1) Program Studi Teknik Informatika

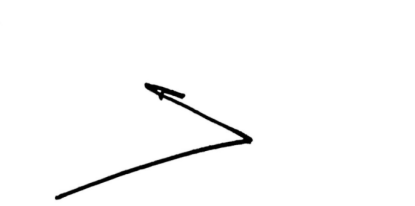
Pada Sekolah Tinggi Manajemen Informatika dan Komputer

(STMIK) Bumigora Mataram

**Oleh:**

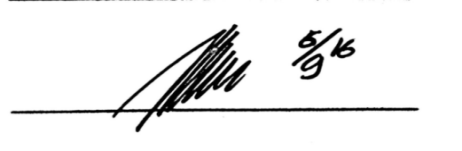
**DESI RATNA NINGSIH**

**1410530187**

****

**Disetujui oleh Penguji :**

1. **Ir. Anthony Anggrawan,M.T.,Ph.D**

**NIP.196112261994031001**

1. **Ir. Bambang Krismono T., M.Kom**

**NIK : 93.6.06**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER**

**(STMIK) BUMIGORA MATARAM**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SKRIPSI**

JUDUL : Analisa Perbandingan Metode SAW dan SMART Untuk Sistem Pendukung Keputusan Ujian Kenaikan Tingkat Atlet Panahan

NAMA : Desi Ratna Ningsih

NIM : 1410530187

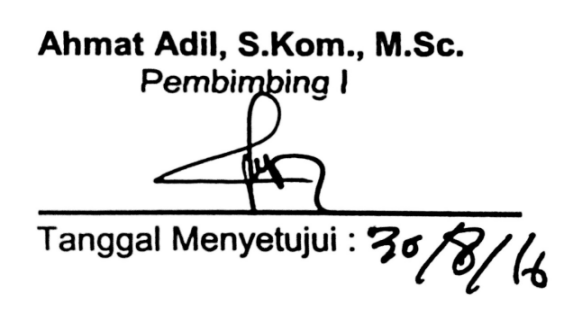
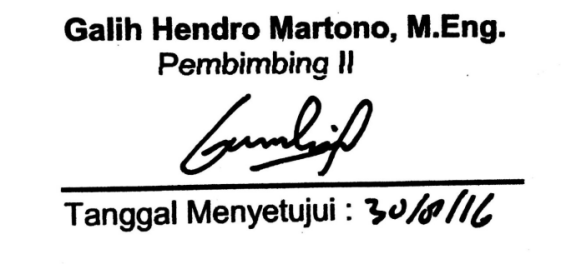
NPM : 14.8.349.74.75.0.5.0187

PROGRAM STUDI : Teknik Informatika

JENJANG : Strata Satu (S1)

DIUJIKAN : 28 Juni 2016

Menyetujui



Telah diterima dan disetujui sebagai salah satu syarat untuk memperoleh Gelar Akademik Sarjana Komputer (S.Kom)

Mengetahui

**Ni Gusti Ayu Dasriani, M.Kom.**

Ketua Program Studi S1 Teknik Informatika

Tanggal Mengetahui :

|  |  |
| --- | --- |
|  | SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER (STMIK) BUMIGORA MATARAM |

# LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama : Desi Ratna Ningsih

NIM : 1410530187

Program Studi : S1 (Strata Satu) Teknik Informatika

Kompetensi : Rekayasa Perangkat Lunak

Menyatakan bahwa skripsi yang berjudul :

*Analisa Perbandingan Metode SAW dan SMART Untuk Sistem Pendukung Keputusan Ujian Kenaikan Tingkat Atlet Panahan***.** Benar-benar merupakan hasil karya pribadi dan seluruh sumber yang dikutip maupun dirujuk telah saya nyatakan dengan benar dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik sesuai dengan yang berlaku.

Mataram, 31 Okt 2018

Desi Ratna Ningsih

**1410530187**

# IZIN PENGGUNAAN

Skripsi ini merupakan syarat kelulusan pada Program Studi S1 Teknik Informatika STMIK Bumigora Mataram, dengan ini penulis setuju jika skripisi ini digandakan (diduplikasi) baik sebagian maupun seluruhnya, ataupun dikembangkan untuk kepentingan akademis yang disetujui oleh pembimbing penulis, Pembantu Ketua 1 atau Ketua STMIK Bumigora Mataram.

Untuk dimaklumi, bahwa menduplikasi, mempublikasikan atau menggunakan skripsi ini, maupun bagian-bagiannya dengan tujuan komersil / keuntungan finansial, tidak diizinkan tanpa adanya izin tertulis dari STMIK Bumigora. Jika hal ini dilanggar maka STMIK Bumigora akan memberikan sanksi sesuai dengna hukum yang berlaku.

Penghargaan akademis terkait isi dari skripsi ini adalah pada penulis dan STMIK Bumigora Mataram. Permintaan izin untuk menduplikasi atau menggunakan materi dari skripsi ini baik sebagian maupun seluruhnya harus ditujukan pada :

Pembantu Ketua I

Ketua Program Studi S1 Teknik Informatika

STMIK Bumigora Mataram

# KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT, Tuhan Yang Maha Esa, atas segala nikmat yang telah diberikan kepada penulis sehingga skripsi ini dapat terselesaikan. Shalawat serta salam selalu tercurah kepada Rasulullah Muhammad SAW. Skripsi yang berjudul **Analisa Perbandingan Metode SAW dan SMART Untuk Sistem Pendukung Keputusan Ujian Kenaikan Tingkat Atlet Panahan**, dibuat sebagai salah satu syarat memperoleh gelar Sarjana Komputer (S.Kom) Program Studi S1 Teknik Informatika, pada Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Bumigora Mataram.

Dengan selesainya skripsi ini, penulis ingin mengucapkan terimakasih kepada pihak-pihak yang telah banyak membantu dalam penyelesaian skripsi ini. Dalam kesempatan ini penulis menyampaikan ucapan terimakasih kepada :

1. Bapak Heroe Santoso, M.Kom, selaku Ketua Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Bumigora Mataram.
2. Ibu Ni Gusti Ayu Dasriani, M.Kom, selaku Ketua Program Studi S1.
3. Bapak Ahmat Adil, S.Kom., Msc, selaku dosen pembimbing yang telah  
   meluangkan waktunya dan memberikan masukan-masukan dalam  
   membimbing penulis.
4. Ibu dan Bapak tercinta yang telah memberikan dukungan moril, dukungan materi dan yang lebih penting telah mendo’akan tiada henti.
5. Teman-teman yang telah banyak memberikan bantuan dan dukungan pada penulis dalam penulisan skripsi.

Semoga skripsi ini dapat memberikan manfaat yang sebesar-besarnya pada kita semua. Sebagai manusia biasa yang mempunyai keterbatasan dan kekurangan, maka penulis meyadari bahwa skripsi ini masih banyak kekurangan-kekurangan baik dalam teknik penulisan, pembahasan dan penyajian untuk itu penulis terbuka untuk menerima kritik dan saran yang membangun dari pembaca untuk kesempurnaan dari skripsi ini, terima kasih

Mataram, Okt 2018

Penulis,

# ABSTRAK

Ujian Kenaikan Tingkat Atlet Panahan merupakan salah satu syarat dari klub panahan muslim Arrihu *Archery* untuk menuju ke jenjang tingkatan kemarihan atlet yang lebih tinggi. Tentu saja dalam kasus ini pihak dari klub panahan muslim Arrihu *Archery* memerlukan sebuah sistem pendukung keputusan yang tepat untuk menentukan lulus dan tidaknya beserta perengkingan atlet di klub panahan muslim Arrihu *Archery.*

Pada penelitian ini, penulis akan memberikan solusi dalah suatu kasus menentukan lulus dan tidaknya beserta perengkingan di klub panahan muslim Arrihu *Archery.* Penelitian ini dilakukan dengan cara mencari nilai bobot terbaik dari setiap nilai atlet yang telah diujikan dengan nilai dari masih-masing kriteria atlet, kemudian data yang di proses diinput dalam aplikasi sistem pendukung keputusan yang nantinya akan menghasilkan informasi lulus dan tidaknya beserta perengkingan atlet di klub panahan muslim Arrihu *Archery*.

**Kata kunci:** Sistem pendukung keputusan, *simple additive weighting* (SAW), klub panahan muslim Arrihu *Archery*, kriteria, informasi, bobot.

# DAFTAR ISI

# DAFTAR GAMBAR

# DAFTAR TABEL

# DAFTAR LAMPIRAN

# BAB I

# PENDAHULUAN

## 1.1 Latar Belakang

Arrihu *Archery* adalah sebuah Klub Panahan Muslim yang didirikan pada tanggal 21 Agustus 2016 dan di resmikan pada tanggal 21 September 2016 di kota Mataram Provinsi Nusa Tenggara Barat. Arrihu *Archery* adalah klub panahan muslim yang bernaung dan mendapat rekomendasi di bawah PERPANI (Persatuan Panahan Indonesia) NTB dengan No Register 09/PERPANI-NTB/IX/2016 (arrihuarchery.id). Dalam perkembangan nya Arrihu *Archery* memiliki cabang klub di pondok pesantren Darul Mujawidin yang berlokasikan di Ampenan. Selain itu, terdapat permintaan untuk pembukaan cabang baru di DPW PKS NTB dan MTSN 1 Mataram. Dalam melaksanakan kegiatan organisasi nya Arrihu *Archery* memilki *web site* dan aplikasi *database* member. Aplikasi tersebut dibangun dengan *python django* dan digunakan untuk melihat profil anggota dan jumlah seluruh anggota di berbagai cabang klub.

Arrihu *Archery* memiliki misi mencetak atlet panahan tradisional dan moderen yang handal dalam segala medan. Demi mewujudkan misi tersebut Arrihu *Archery* berupaya membuat kurikulum dan menambah jumlah pelatih yang tersertifikasi nasional melalui kegiatan *Training for Trainer* (TFT) dari PERPANI NTB. Kurikulum tersebut di ajarkan oleh pelatih kepada atlet atau anggota yang tergabung dalam latihan panahan Arrihu *Archery*. Dalam kurikulum nya Arrihu *Archery* mempelajari dua jenis busur yang berbeda yaitu *horse bow* dan *recurve bow*. Pada kurikulum busur *horse bow*, terdapat tingkatan mempelajari materi antara lain, *Novice, Intermediate, dan Warden*. Selain itu, pada kurikulum busur *recurve bow*, terdapat tingkatan mempelajari materi antara lain, *Novice, Intermediate dan Advanced.* Terdapat tabel bobot di setiap tingkatan yang diujikan, bobot tersebut bersifat dinamis dan tiap tingkatan berbeda pembobotannya.

Ujian Kenaikan Tingkat (UKT) adalah tata cara untuk dapat mengetahui keberhasilan dan kemahiran atlet dalam memanah berdasarkan kurikulum yang ada. Tata cara pelaksanaan UKT dilakukan secara paralel maupun terpusat di berbagai tempat baik secara bersamaan maupun dalam waktu yang berbeda. Secara umum UKT panahan memiliki dua jenis ujian, yaitu ujian tertulis dan ujian praktik. Jenis materi dalam kurikulum terbagi menjadi 6 (enam) aspek dalam pengujiannya, yaitu Pengetahuan (*knowledge*), Fisik (*physic*), Mental (*mental*), Teknik (*technique*), Kecepatan (*speed*) dan Akurasi (*accuracy*). Setiap aspek pengujian tersebut memiliki konten form yang berbeda dalam pengujiannya maupun penilaiannya. Pengujian terhadap atlet menggunakan form tersebut dilakukan oleh pelatih dan hasil total nya di inputkan dan di kalkulasikan untuk dapat mengetahui kelulusan dan rangking atlet.

Hasil UKT yang di dapatkan dari proses kalkulasi adalah kelulusan dan rangking atlet. Proses kalkulasi dengan rumus kelulusan dan rangking ini membutuhkan waktu yang lama jika dikalikan dengan jumlah anggota per satu kali UKT di tiap tingkatan nya. Akibatnya, peserta bisa menunggu lama untuk mendapatkan hasil jika UKT dilakukan secara serentak dengan jumlah peserta yang banyak. Dampak ini membuat latihan panahan bisa berhenti sementara karena materi berdasarkan kurikulum tidak bisa dilanjutkan, sedangkan latihan panahan harus terus di jalankan untuk menjaga kondisi atlet tetap prima. Selain itu, hasil kelulusan dan rangking tidak bisa dilihat secara menyuluruh oleh setiap atlet, pelatih dan pengurus secara *real time*. Sedangkan hasil kelulusan dan rangking dibutuhkan secara *real time* untuk menjadi pertimbangan evaluasi bagi organisasi, pelatih dan atlet yang di bina.

Berdasarkan pada permasalahan tersebut, perlu dibangun Sistem Pendukung Keputusan (SPK) Ujian Kenaikan Tingkat Atlet Panahan untuk mengetahui hasil kelulusan dan rangking secara *real time* dan komperhensif. Selain itu, SPK tersebut dibutuhkan untuk bisa berintegrasi dengan aplikasi *database* member yang telah ada. Sehingga hasil Kelulusan dan Rangking UKT juga dapat di akses oleh anggota umum/atlet, pelatih dan anggota pengurus. Di harapkan dengan tercapainya kebutuhan tersebut atlet dapat mengetahui capaian yang telah dilakukan dalam UKT yang telah dilaksanakan. Di sisi lain, bagi pelatih dan pengurus Arrihu *Archery* hasil capaian kelulusan dan rangking sangat dibutuhkan sebagi indikator keberhasilan secara menyeluruh terhadap pengelolaan kurikulum, atlet dan organisasi.

Pada penelitian ini penulis akan menganalisa perbandingan metode *Simple Multy Attribute Rating Technique* (SMART) dan *Simple Additive Weighting* (SAW), dalam UKT di klub panahan muslim Arrihu *Archery*. Karena metode ini merupakan metode yang umum digunakan untuk komputasi yang relative mudah. Logika dan prinsip dasar yang digunakan metode *Simple Additive Weighting* (SAW) adalah nilai keseluruhan dari sebuah alternative dihitung dari penjumlahan terbobot dari nilai atributnya. Metode *Simple Additive Weighting* (SAW) merupakan metode yang paling dikenal dan banyak digunakan, hal ini disebabkan metode ini sederhana dan mudah dipahami. Konsep dasar metode *Simple Additive Weighting* (SAW) adalah mencari pemjumlahan terbobot dari rating kinerja pada setiap alternatif dari semua atribut. Metode *Simple Additive Weighting* (SAW) membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat dibandingkan dengan semua rating alternatif yang ada. Sedangkan konsep dasar metode *Simple Multy Attribute Rating Technique* (SMART) teknik pengambilan keputusan *multi* kriteria yang didasarkan pada teori bahwa setiap alternatif terdiri dari sejumlah kriteria yang memiliki nilai-nilai dan setiap kriteria memiliki bobot yang menggambarkan seberapa penting ia dibandingkan dengan kriteria lain. Pembobotan ini digunakan untuk menilai setiap alternatif agar diperoleh alternatif terbaik.

Oleh karena itu, skripsi ini dibuat untuk mempermudah seluruh proses ujian kenaikan tingkat atlet panahan dengan pemanfaatan teknologi informasi. Sehingga muncullah sebuah gagasan skripsi dengan judul **“ANALISA PERBANDINGAN METODE SMART DAN SAW UNTUK SISTEM PENDUKUNG KEPUTUSAN UJIAN KENAIKAN TINGKAT ATLET PANAHAN”.**

**1.2 Perumusan Masalah**

Dari latar belakang yang telah dijelaskan sebelumnya, rumusan masalah dari penulisan skripsi ini adalah “Bagaimana Cara Menganalisa Perbandingan Metode SAW dan SMART untuk Ujian Kenaikan Tingkat Atlet Panahan ”

**1.3 Batasan Masalah**

Dari beberapa uraian masalah yang ada, maka diberikan batasan masalah sebagai berikut:

1. Tempat penelitian di Klub Panahan Muslim Arrihu Archery yang bernaung dibawah PERPANI NTB.

2. Sistem pendukung keputusan yang akan di bangun berfokus pada hasil kelulusan dan perengkingan secara menyeluruh oleh setiap atlet, pelatih dan pengurus secara *real time*.

3. Sistem pendukung keputusan di bangun dan di kembangkan berdasarkan Aplikasi Database member yang sudah tersedia sebelumnya.

4. Sistem pendukung keputusan yang diterapkan menggunakan metode Simple Additive Weighting (SAW) *dan Simple Multy Attribute Ratting Technique* (*SMART*) dan menerapkan teknik pembobotan secara langsung (*direct weighting*).

5. Bahasa pemograman yang di gunakan yaitu bahasa pemograman Python dengan Framework Django.

6. Desain database menggunanak MySQL Workbench.

**1.4 Tujuan dan Manfaat Penelitian**

**1.4.1 Tujuan Penulisan**

Menghasilkan Analisa Sistem Pendukung Keputusan Ujian Kenaika Tingkat Atlen Panahan menggunakan perbandingan dua metode yaitu metode Simple Additive Weighting (SAW) dan Simple Multy Attribute Ratting Technique (SMART), yang dapat memberikan solusi dalam hasil kelulusan dan perengkingan secara komperhensif dan *real time* sehinga bisa digunakan oleh setiap atlet, pelatih dan pengurus dalam mengembangkan klub panahan nya.

**1.4.2 Manfaat Penelitian**

1. Bagi Pelatih

Memudahkan pelatih dalam melihat hasil kelulusan dan perengkingan secara komperhensif dan *real time*.

2. Bagi Atlet

Memudahkan dan mempercepat hasil ujian, keamanan data peserta UKT terjaga dan terpusat.

3. Bagi Pengurus

Memberikan/menambah sarana prasarana guna meningkatkan mutu Klub Panahan Muslim Arrihu Archery.

4.Bagi Penulis

Sebagai bahan pembuatan Skripsi guna mencapai sarjana TeknikInformatika di Sekolah Tinggi Manajemen Informatika Bumigora Mataram.

**1.5 Metode Penelitian**

Metodologi yang penulis gunakan dalam pembuatan skripsi ini adalah *waterfall.* Adapun tahapan dalam pengembangan sistem tersebut adalah pengumpulan data, analisa, desain, coding dan ujicoba serta implementasi (Jogiyanto. 2005).

1. Pengumpulan data

Pada tahap pengumpulan data, penulis melakukan pengumpulan data yang berkaitan dengan data - data yang dibutuhkan dalam pembuatan skripsi. Adapun metode wawancara dan observasi terhadap sistem yang akan dibangun.

* 1. Wawancara

Wawancara dilakukan untuk mengumpulkan data dengan memberikan pertanyaan kepada pihak Klub Panahan Muslim Arrihu Archery.

* 1. Observasi

Observasi yang dilakukan dengan metode pengamatan dan pencatatan secara sistematis terhadap objek yang diselidiki.

c. Studi Literatur

Studi Literatur dilakukan dengan cara mempelajari konsep, metode, teknik, maupun informasi dari berbagai sumber seperti internet, buku, jurnal, maupun artikel ilmiah lainnya yang berkaitan dengan sistem informasi materi pendukungnya.

2. Analisa

Pada tahap ini dilakukan untuk mengetahui seluruh kebutuhan aplikasi, termasuk didalamnya kegunaan apliaksi yang diharapka pengguna dan batasan aplikasi.

3. Desain

Tahap selanjutnya setelah analisa adalah desain sistem. Pada tahap desain sistem ini, dilakukan pembuatan *blueprint* (cetak biru) terhadap sistem yang akan dibangun. Pada tahap ini dilakukan normalisasi untuk menghasilkan struktur database, penentuan relasi tabel, pembuatan use case diagram, desain layout (admin, operator data dan user).

4. *Coding*

Tahap ini dilakukan untuk pembuatan aplikasi dari hasil tahap desain. Pada pembuatan apliaksi, penulis menggunakan bahasa pemograman Python dan framework Django untuk membangun pengkodingan aplikasi.

5. Ujicoba

Setelah aplikasi selesai dibuat, dilakukan pengecekan apakah terjadi kesalahan - kesalahan yang kemungkinan terjadi. Tahap ini bertujuan untuk meminimalisir kesalahan (*error*).

6. Implementasi

Tahap terakhir adalah proses penyerahan aplikasi terhadap klub panahan muslim Arrihu Archery sebagai pengguna aplikasi atau sebagai implementasi akhir hasil pembuatan aplikasi.

**1.6 Sistematika Penulisan**

Sistematika dari penulisan skripsi ini dibagi menjadi 5 (lima) bab yang masing - masing bab telah dirancang dengan suatu tujuan tertentu. Berikut penjelasan tentang masing - masing bab:

BAB I :PENDAHULUAN

Bab ini berisikan latar belakang, perumusan masalah, batasan masalah, tujuan dan manfaat penulisan, metodologi penelitian, dan sistematika penulisan.

BAB II :LANDASAN TEORI

Pada bab ini mencakup tentang pengertian berbagai hal yang berhubungan dengan materi skripsi dan bahasa pemograman yang di gunakan.

BAB III :ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini membahas tentang Metodologi, Tinjauan Organisasni (struktur organisasi, visi, misi, tujuan organisasi), Identifikasi Masalah (penjelasan masalah, penyebab masalah, analisis alternative solusi masalah), Analisa Kebutuhan (perangkat lunak, perangkat keras, sumberdaya manusia).

BAB IV :HASIL DAN PEBAHASAN

Pada bab ini memaparkan atau menjelaskan hasil dari penulisan tugas akhir yang berupa implementasi perancangan program dari bab sebelumnya.

BAB V :PENUTUP

Pada bab ini berisikan simpulan dan saran untuk pengembangan/penyempurnaan lebih lanjut. Bab ini juga berisi saran-saran yang dimungkinkan untuk pengembangan perangkat lunak ini.

**BAB II**

**LANDASAN TEORI**

**2.1 Pengertian Aplikasi**

Menurut Dhanta (2009), aplikasi adalah suatu sub kelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakuka suatu tugas yang diinginkan pengguna (p.32). Aplikasi merupakan penggunaan dalam suatu komputer, instruksi (statemen) yang di susun sedemikian rupa sehingga komputer dapat memproses input menjadi output (Jogiyanto, 1999, p.12).

**2.2 Pengertian Aplikasi Web**

Aplikasi web adalah sebuah sistem informasi yang mendukung intraksi pengguna melalui antarmuka berbasis web. Aplikasi web adalah bagian dari *client-side* yang dapat dijalankan oleh *browser* web. Aplikasi web bertujuan untuk digunakan secara luas oleh komunitas pengguna dengan berbagai kebutuhan. Oleh karena itu, antarmuka pengguna dan fitur-fitur usabilitas harus memenuhu kebutuhan yang berdeda (Simarmata, 2010, p.56).

**2.3 Pengertian Sistem**

Terdapat dua kelompok pendekatan didalam mendefinisikansistem, yang menekankan kepada prosedur dan pada kemponen atau elemennya.

1. Pendekatan sistem yang lebih menekankan pada elemen-elemen atau kelompoknya, yang dalam hal ini sistem itu didefinisikan sebagai “Suatu jaringan kerja dari prosedur-proedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu aturan tertentu”.

2. Pendekatan sistem yang lebih menekankan pada prosedurnya: “Pendekatan prosedur adalah pendekatan yang menekankan pada konsep system berdasarkan prosedur-prosedur yang ada dalam sistem”.

Definisi sistem menurut Jogianto (2005), “Sistem adalah kumpulan dari elemen-elemen yang berintegrasi untuk mencapai tujuan tertentu”.

Definisi sistem nemurut Hanif (2007), “Suatu kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung satu sama lain”.

Definisi sistem menurut Sri (2008), “Sistem adalah kumpulan/group dari bagian/komponen aoaoun baik pisik maupun non pisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai suatu tujuan tertentu”.

**2.3.1 Karakteristik Sistem**

Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu (Jogiyanto. 2005), yaitu:

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan. Konponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batas Sistem

Batas sistem adalah daerah yang membatasi antara suatu system dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan.

3. Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang memengaruhi oprasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugukan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

4. Subsistem

Bagian dari sistem yang beraktivitas dan berinteraksi satu sama lain untuk mencapai tujuan dengan sasarannya masing-masing.

5. Penghubung Sistem

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya. Keluaran dari satu subsistem akan menjadi masukan untuk subsistem yang lainnya dengan melalui penghubung.

6. Masukan Sistem

Masukan (*input*) adalah energi yang dimasukan ke dalam sistem. Masukan dapat berupa mesukan perawatan (*maintenance input* dan masukan sinyal (*signal input*). Masukan perawatan adalah energi yang dimasukan supaya sistem tersebut dapat beroprasi. Masukan sinyal adalah energi yang diproses untuk didapatkan keluaran.

7. Keluaran Sistem

Keluaran (*output*) adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Misalkan untuk sistem komputer, panas yang dihasilkan adalah keluaran yang tidak berguna dan merupakan hasil sisa pembuangan, sedangkan informasi adalah keluaran yang dibutuhkan.

8. Pengolahan Sistem

Suatu sistem dapat mempunyai suatu bagian pengolahan yang akan merubah masukan menjadi keluaran.

9. Sasaran Sistem

Suatu sistem pasti mempunyai tujuan atau sasaran kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan berguna. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

**2.3.2 Bentuk Umum Sistem**

Bentuk umum dari sistem terdiri dari atas masukan (input), proses, dan keluaran (output). Dalam bentuk umum sistem ini bisa melakukan satu atau lebih masukan yang akan diproses dan mengkasilkan keluaran sesuai dengan yang direncanakan sebelumnya.

Gambaran umum megenai sistem ditunjukan pada gambar berikut ini:



**Gambar 2.1 Bentuk Umum Sistem**

**[Sumber : Jogianto (1999)]**

**2.4 Pengertian Pengolahan Data**

Definisi menurut Jogianto (2005), “Data adalah kumpulan kejadian yang diperoleh dari satu kenyataan. Data dapat berupa angka-angka, huruf-huruf atau simbol-simbol khusus atau gabungan darinya”. “Sedangkan pengolahan data (*data processing*) adalah manipulasi dari data ke dalam bentuk yang lebih berguna dan lebih berarti, berupa suatu informasi”.

Menurut Teguh Wahyono (2004). Data adalah bahan baku informasi,didefinisikan sebagai kelompok teratur simbol-simbol yang mewakili kuantitas, tindakan, benda, angka, maupun simbol-simbol seperti \*, $ dan /. Data disusun untuk diolah dalam bentuk struktur data, struktur file, dan basis data (p.3).

Menurut Jogianto (2005), “Serangkaian operasi atau informasi-informasi yang telah direncanakan guna mencapai tujuan atau hasil yang diinginkan”. Terdapat 8 (delapan) unsur pokok pengolahan data yaitu:

1. *Writing*, *Typing*, *Card Punching or Paper Tape perforping* (*Frequently, Called Input*) atau menulis, mengetik, membuat lubang atau pons pada kertas atau kartu sering disebut masukan.

2. *Recordingor Printing* (*Frequently Called Output*) atau mencatat atau mencetak.

3. *Sorting* atau mengurutkan.

4. *Reading* atau membaca.

5. *Transmeeting* atau menyampaikan atau memindahkan.

6. *Calculating* atau menghitung.

7. *Comparing* atau membandingkan.

8. *Saving* atau menympan.

**2.5 Python**

Python adalah bahasa pemrograman yang memungkinkan untuk bekerja lebih cepat dan lebih efektif dalam mengintegrasikan sistem (Python, 2015). Python adalah bahasa interpretasi dengan tujuan umum, interaktif, berorientasi objek, dan bahasa program tingkat tinggi. Python ditulis oleh Guido van Rossum pada tahun 1985 – 1990 (Anonim, Python Tutorial, 2015). Python adalah salah satu bahasa pemrograman yang populer yang digunakan untuk program *standalone* dan penulisan skrip aplikasi dalam skala besar. Python merupakan bahasa pemrograman yang gratis, ringan, dan mudah digunakan (Lutz, 2007).

**2.6 Django**

Django adalah salah satu *web framework* yang paling populer yang digunakan saat ini. Django digunakan oleh beberapa *website* besar, seperti Pinterest, Instagram, Disqus, dan NASA. Dengan beberapa baris kode, memungkinkan untuk membangun *website* yang fungsional dan aman secara cepat, dan dapat dikembangkan ke skala besar untuk menangani jutaan pengguna (Ravindran, 2015). Django, seperti *web* framework yang lain, yang dibangun oleh bahasa pemrograman pokok, sedangkan Django dibangun menggunakan bahasa Python (Alchin, 2013).

### 2.6.1 Django Object Relational Mapping (ORM)

*Django ORM* merupakan suatu teknik yang digunakan untuk mengkonversi *query* *SQL* *database* menjadi bahasa Python, sehingga Django ORM menjembatani antara program dengan *database*. Dengan menggunakan Django ORM memiliki beberapa keuntungan, yaitu:

* Terfokus dalam menulis skrip dengan bahasa Python, sehingga tidak kesulitan jika tidak menguasai *query SQL* *database* yang digunakan.
* Memanipulasi *database* ke dalam bentuk objek-objek, sehingga mempermudah dalam mengolah *database*.
* Mempercepat pengembangan program.
* Menyederhanakan fungsi-fungsi *query SQL* yang kompeks yang melibatkan *query* dari beberapa tabel sekaligus.
* Jika ingin mengganti *engine database*, tidak perlu mengubah skrip kode yang telah ditulis.

## 2.7 Basis Data

Merupakan suatu kumpulan data terhubung (*interrelated data*) yang disimpan secara bersama-sama pada suatu media, tanpa ada suatu kerangkapan data (kalaupun ada maka kerangkapan data tersebut harus seminimal mungkin dan terkontrol (*controlled redundancy*), data disimpan dengan cara-cara tertentu sehingga mudah digunakan atau ditampilkan kembali; data dapat digunakan oleh satu atau lebih program-program aplikasi secara optimal dan tanpa mengalami ketergantungan dengan program yang akan menggunakannya; data disimpan sedemikian rupa sehingga proses penambahan, pengambilan dan modifikasi data dapat dilakukan dengan mudah dan terkontrol (Sutanta, 2004).

### 2.7.1 MySQL

MySQL merupakan *database* yang paling populer digunakan untuk membangun aplikasi web yang menggunakan *database* sebagai sumber dan pengelola datanya. Kepopuleran MySQL dimungkinkan karena kemudahannya untuk digunakan, cepat secara kinerja *query,* dan mencukupi untuk kebutuhan *database* perusahaan-perusahaan skala menengah kecil. MySQL merupakan *database* yang di gunakan oleh situs-situs terkemuka di *internet* untuk menyimpan datanya (Sidik, 2005, p.1).

*Software database* MySQL merupakan *software* manajemen *database* yang *open source,* sebelumnya merupakan *software* manajemen *database* yang *shareware. Shareware* adalah suatu *software* yang dapat didistribusikan secara bebas untuk keperluan penggunaan secara pribadi, tetapi jika digunakan secara komersial maka pemakai harus mempunyai lisendi dari pembuatnya. *Software open source* menjadikan *software* dapat didistribusikan secara bebas dan dapat dipergunakan untuk keperluan pribadi maupun komersial, termasuk didalamnya *source code software* tersebut (Sidik, 2005, p.2).

## 2.8 UML

*Unified Modeling Language (UML)* adalah keluarga notasi rafis yang didukung oleh *meta-model* tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO). Definisi ini merupakan definis yang sederhana. Pada kenyataannya, pendapat orang-orang tentang UML berbeda satu sama lain. Hal ini dikarenakan oleh sejarahnya sendiri dan oleh perbedaan persepsi tetang apa yang membuat sebuah proses rancang-bangun perangkat lunak efektif (Fowler, 2004, p. 1).

### 2.8.1 Use Case Diagram

Diagram *use case* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Hal yang ditekankan pada diagram ini adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* menyatakan sebauh aktivitas atas perkerjaan tertentu, misalnya *login* ke sistem, meng-*create* sebuah daftar belanja, dan lain sebagainya. Aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu (Tohari, 2014, p. 48).

Berikut ini adalah daftar simbol-simbol pada *Use Case Diagram*:

Tabel 2.1 Simbol-simbol pada *Use Case Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *Actor* | Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*. |
| 2 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*). |
| 3 |  | *Generalization* | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 4 |  | *Include* | Menspesifikasikan bahwa *use case* sumber secara *eksplisit*. |
| 5 |  | *Extend* | Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan. |
| 6 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 7 |  | *System* | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
| 8 |  | *Use Case* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 9 |  | *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |
| 10 |  | *Note* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi |

Sumber (Amalia, 2016)

### 2.8.2 *Activity* Diagram

*Activity* diagram adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja. Dalam beberapa hal, diagram ini memainkan peran mirip sebuah diagram alir, tetapi perbedaan prinsip antara diagram ini dan notasi diagram alir adalah diagram ini mendukung behavior paralel (Fowler, 2004, p. 163).

Berikut ini adalah daftar simbol-simbol pada *Activity Diagram:*

Tabel 2.2 Simbol-simbol pada *Activity Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| **1** |  | *Actifity* | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain |
| **2** |  | *Action* | State dari sistem yang mencerminkan eksekusi dari suatu aksi |
| **3** |  | *Initial Node* | Bagaimana objek dibentuk atau diawali. |
| **4** |  | *Actifity Final Node* | Bagaimana objek dibentuk dan dihancurkan |
| **5** |  | *Fork Node* | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran |

Sumber (Amalia, 2016)

### 2.8.3 Sequence Diagram

*Sequence Diagram* menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untu kmenunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem. dalam UML, objek, pada diagram *sequence* digambarkan dengan segi empat, yang berisi nama dari objek yang digarisbawahi (Tohari, 2014, p. 101).

Berikut ini adalah daftar simbol-simbol pada *Sequence Diagram:*

Tabel 2.3 Simbol-simbol pada *Sequence Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *LifeLine* | Objek *entity*, antarmuka yang saling berinteraksi. |
| 2 |  | *Message* | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi |
| 3 |  | *Message* | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi |

Sumber (Amalia, 2016)

### 2.8.4 Class Diagram

Kelas (*class*) adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan perancangan berorientasi objek. Kelas menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi) (Tohari, 2014, p. 83).

Berikut ini adalah daftar simbol-simbol pada *Class Diagram:*

Tabel 2.4 Simbol-simbol pada *Class Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *Generalization* | Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 2 |  | *Nary Association* | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek. |
| 3 |  | *Class* | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama. |
| 4 |  | *Collaboration* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 5 |  | *Realization* | Operasi yang benar-benar dilakukan oleh suatu objek. |
| 6 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri |
| 7 |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya |

Sumber (Amalia, 2016)

**2.9 Sistem Pendukung Keputusan**

**2.9.1 Konsep Dasar Sistem Pendukung Keputusan**

Keputusan merupakan kegiatan memilih suatu strategi atau tindakan dalam pemecahan masalah tersebut. Tindakan memilih strategi atau aksi yang diyakini manajer akan memberikan solusi terbaik atas sesuatu itu disebut pengambilan keputusan.

Tujuan dari keputusn adalah untuk mencapai target atau aksi tertentu yang harus dilakukan.

Kriteria atau ciri-ciri dari keputusan adalah :

1. Banyak pilihan.

2. Ada kendala atau syarat.

3. Mengikuti suatu pola/model tingkah laku, baik yang terstruktur maupun tidak terstruktur.

4. Banyak input/variabel.

5. Ada faktor resiko.

6. Dibutuhkan kecepatan, ketepatan, dan keakuratan (Kusrini, 2007, p.7)

**2.9.2 Pengertian Sistem Pendukung Keputusan**

Menurut Alter (2002), *Decision Support Sistem* (DSS) atau sistem pendukung keputusan merupakan sistem informasi interaktif yang menyediakan informasi pemodelan dan pemanipulasian data. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semi terstruktur dan situasi yang tidak terstruktur, dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat.

Bonczek (1980) mendefinisikan sistem pendukung keputusan adalah sebagai sebuah sistem berbasis komputer yang terdiri dari atas komponen-komponen antara lain komponen sistem bahasa (*language*), komponen sistem pengetahuan (*knowledge*) dan komponen sistem pemrosesan masalah.

Turban dan Aronson (1998) mendefinisikan sistem penunjang keputusan adalah sebagai sistem yang digunakan untuk mendukung danmembantu pihak manajemen pengambilan keputusan pada kondisi semi terstruktur dan tidak terstruktur. (dalam Kursini, 2007, p.15).

**2.9.3 Tahapan Dalam Pengambilan Keputusan**

Menurut Herbet A.Simon (Turban dkk, 2005), tahap-taha yang harus dilalui dalam proses pengambilan keputusan sebagai berikut :

1. Tahap Penelusuran (*Inteligence*)

Tahap ini merupakan proses penelusuran dan pendekatan dari lingkup proplematika serta proses pengenalan masalah. Data masukan diperoleh, diproses dan diuji dalam rangka mengidentifikasikan masalah.

2. Tahap Perancangan (*Design*)

Tahap ini merupakan proses pengembangan dan perancangan alternatif tindakan/solusi yang dapat di ambil. Ini merupakan representasi kejadian nyata yang di sederhanakan sehingga diperlukan proses validasi dan verifikasi untuk mengetahui keakuratan model dalam meneliti masalah yang ada.

3. Tahap Pemilihan (*choice*)

Tahap ini dilakukan pemilihan terhadap berbagai alternative solusi yang di munculkan pada tahap perancangan agar ditentukan/dengan memperhatikan kriteria-kriteria berdasarkan tujuan yang akan di capai.

4. Tahap Implementasi (*Implementation*)

Tahap ini di lakukan penerapan terhadaprancangan sistem yang telah dipilih pada tahap pemilihan.

**2.9.4 Komponen Sistem Pendukung Keputusan**

Aplikasi Sistem Pendukung Keputusan terdiri dari beberapa subsistem:

1. Subsistem Manajemen Data

Memasukan satu database yang berisi data yang relevan untuk suatu situasi dan dikelola oleh perangkat lunak yang disebut sistem manajemen database (DBMS/*Data Base Manajement System*). Subsistem manajemen data bisa diinterkoneksikan dengan data *warehouse* perusahaan, suatu repositori untuk data perusahaan yang relevan dengan pengambilan keputusan.

2. Subsistem Manajemen Model

Merupakan paket perangkat lunak yang memasukan model keuangan, statistik, ilmu manajemen, atau model kuantitatif lain yang memberikan kapabilitas analitik dan manajemen peangkat lunak yang tepat. Bahasa-bahasa pemodelan untuk membangun model-model kostum juga dimasukkan. Perangkat lunak itu sering sistem manajemen basis model. Komponen tersebut bisa dikoneksikan ke penyimpanan atau *eksternal* yang ada pada model.

3. Subsistem Antarmuka Pengguna

Pengguna berkomunikasi dengan memerintahkan sistem pendukung keputusan melalui subsistem tersebut. Pengguna adalah bagian yang dipertimbangkan dari sistem. Para peneliti menegaskan bahwa beberapa kontribusi unik dari sistem pendukung keputusan berasal dari interaksi yang intensif antara komputer dan pembuat keputusan.

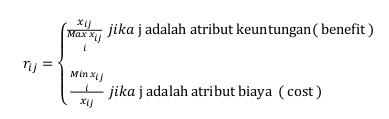
4. Subsistem Manajemen Berbasis Pengetahuan

Subsistem tersebut mendukung semua subsistem lain atau bertindak langsung sebagai suatu komponen insependen dan bersifat opsional. (Kursini, 2007, p.25).

**2.10 Metode SAW**

**2.10.1 Definisi Metode SAW**

Metode *Simple Additive Weighting Method (SAW)* sering juga dikenal istilah metode penjumlahan terbobot. Konsep dasar metode *Simple Additive Weighting Metod (SAW)* adalah mencaripenjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut. Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternative yang ada (Mude, 2016).

(2.1)

Dimana rij adalah rating kinerja ternormalisasi dari alternatif Ai pada atribut Cj; i=1,2,…,m dan j=1,2,…,n. Nilai preferensi untuk setiap alternatif ( Vi) diberikan. Nilai Vi yang lebih besar mengindikasikan bahwa alternatif Ai lebih terpilih.

(2.1)

**2.12.1 Teknik Metode SAW**

* 1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan.
  2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.

1. Membuat matriks keputusan berdasarkan kriteria, kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut sehingga diperoleh matriks ternormalisasi R.
2. Hasil akhir diperoleh dari proses perangkingan yaitu penjumlahan dari perkalian matrik ternormalisasi R dengan Vector bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternative terbaik sebagai solusi (dalam Putra, 2011).

**2.11 Metode SMART**

**2.11.1 Definisi Metode *SMART***

*SMART* (*Simple Multi - Attribute Rating Technique*) merupakan metode pengambilan keputusan multi kriteria yang dikembangkan oleh Edward pada tahun 1977. Teknik pengambilan keputusan *multi* kriteria ini didasarkan pada teori bahwa setiap alternatif terdiri dari sejumlah kriteria yang memiliki nilai-nilai dan setiap kriteria memiliki bobot yang menggambarkan seberapa penting ia dibandingkan dengan kriteria lain. Pembobotan ini digunakan untuk menilai setiap alternatif agar diperoleh alternatif terbaik. *SMART* menggunakan *linier additive* model untuk meramal nilai setiap alternatif. *SMART* merupakan metode pengambilan keputusan yang fleksibel. *SMART* lebih banyak digunakan karena kesederhanaannya dalam merespon kebutuhan pembuat keputusan dan caranya menganalisa respon (Atiqah, 2013, p.75).

*SMART* merupakan metode dalam pengambilankeputusan *multi* atribu. Teknik pengambilan keputusan multiatribut ini digunaakn untuk mendukung pembuat keputusan dalam memilih beberapa alternatif. Setiap alternatif keputusan harus memiliki sebuat alternatif yang sesuai dengan tujuan yang dirumuskan. Setiap alternatif terdiri dari sekumpulan atribut dan setiap atribut mempunyai nilai-nilai. Nilai ini dirata-rata sengan skala tertentu. Setiap atribut mempunyai bobot yang menggambarkan seberapa penting skala tertentu. Setiap atribut mempunyai bobot yang menggambarkan seberapa penting suatu atribut dibandingkan dengan atribut lain. Pembobotan dan pemberian peringkat ini digunakan untuk menilai setiap alternatif agar diperoleh alternative terbaik (Yunitarini, Rizka, 2013, p. 46).

*SMART* menggunakan *linier adaptive* model untuk meramal nilai setiap alternatif. *SMART* lebih banyak digunakan karena kesederhanaannya dalam merespon kebutuhan pembuat keputusan dan caranya menganalisa respon. Analisis yang terbaik adalah transparan sehingga metode ini memberikan pemahaman masalah yang tinggi dan dapat diterima oleh pembuat keputusan. Pembobotan pada SMART mengunakan skala 0 sampai 1, sehingga mempermudah perhitungan dan perbandingan nilai pada masing-masing alternatif. Model yang digunakan dalam *SMART* yaitu (Yunitarini, Rika. 2013, p. 46):

**(1)**

Pembobotan yang dipakai di dalam motode *SMART* ada 3 jenis yaitu pembobotan secara langsung (*direct weighting*), pembobotan swing (*swing weighting*), pembobotan centroid (*centroid weighting)* (Yunitarini, Rika. 2013, p. 47)*.*

**2.11.2 Teknik Metode *SMART***

Adapun teknik atau langkah-langkah dalam proses metode *SMART* (Yunitarini, Rika. 2013, p. 46), antara lain :

1. Identifikasi user yang nantinya bertanggung jawab dalam mengambil keputusan.

2. Identidikasi permasalahan yang ada dengan melihat akar permasalahan dan batasan-batasan yang ada agar nantinya tidak menyimpang dari tujuan yang ingin dicapai.

3. Identifikasi alternatif yang ada untuk mencapai tujuan dari sistem yang akan dibuat.

4. Identifikasi kriteria-kriteria yang akan mendukung pengambilan keputusan.

5. Memberikan peringkat atau bobot untuk setiap kriteria. Pemberian peringkat atau bobot ini ditentukan oleh user dan sistem akan memberikan bobot skala default dari peringkat yang dimasukan.

6. Memberikan penilaian setiap kriteria untuk setiap alternatif. Juga dilakukan oleh user dimana penilaian setiap kriteria dapat bersifat kualitatif maupun kuantitatif untuk setiap alternatifnya.

7. Mengembangkan *Single-Atribute Utilities* yang mencerminkan seberapa baik setiap alternatif dilihat dari setiap kriteria. Tahap ini adalah memberikan suatu nilai pada semua kriteria untuk semua alternatif dengan nilai yang berskala 0 sampai 1.

8. Menghitung penilaian terhadap setiap alternative.

**2.13 Olahraga**

Aktifitas fisik juga merupakan kerja fisik yang menyangkut system lokomotor tubuh yang ditujukan dalam menjalankan aktifitas hidup sehari - harinya. Suatu aktifitas fisik yang memiliki tujuan tertentu dan dilakukan dengan aturan – aturan tertentu secara sistematis seperti adanya aturan waktu, target denyut nadi, jumlah pengulangan gerakan dan lain - lain, hal ini disebut latihan. Sedangkan yang dimaksud dengan olahraga adalah latihan yang dilakukan dengan mengandung unsur rekreasi (Lesmana, 2003).

Olahraga adalah suatu bentuk kegiatan fisik yang dapat meningkatkan kebugaran jasmani. Dalam olahraga tidak hanya melibatkan sistem muskuloskeletal semata, namun juga mengikutsertakan sistem lain seperti sistem kardiovaskular, system respirasi, sistem ekskresi, sistem saraf dan masih banyak lagi. Olahraga mempunyai arti penting dalam memelihara kesehatan dan menyembuhan tubuhyang tidak sehat (Mutohir & Maksum, 2007).

**2.14 Panahan**

Panahan adalah seni, keahlian serta praktik membidik dengan busur dan anak panah (Barrett, 1987).

Panahan (Inggris:Archery) adalah suatu kegiatan menggunakan busur panah untuk menembakkan anak panah. Bukti-bukti menunjukkan bahwa sejarah panahan telah dimulai sejak 5.000 tahun yang lalu yang awalnya digunakan untuk berburu dan kemudian berkembang sebagai senjata dalam pertempuran dan kemudian sebagai olahraga ketepatan. Seseorang yang gemar atau merupakan ahli dalam memanah disebut juga sebagai pemanah (wikipedia.org).

# BAB III

# ANALISIS DAN PERANCANGAN SISTEM

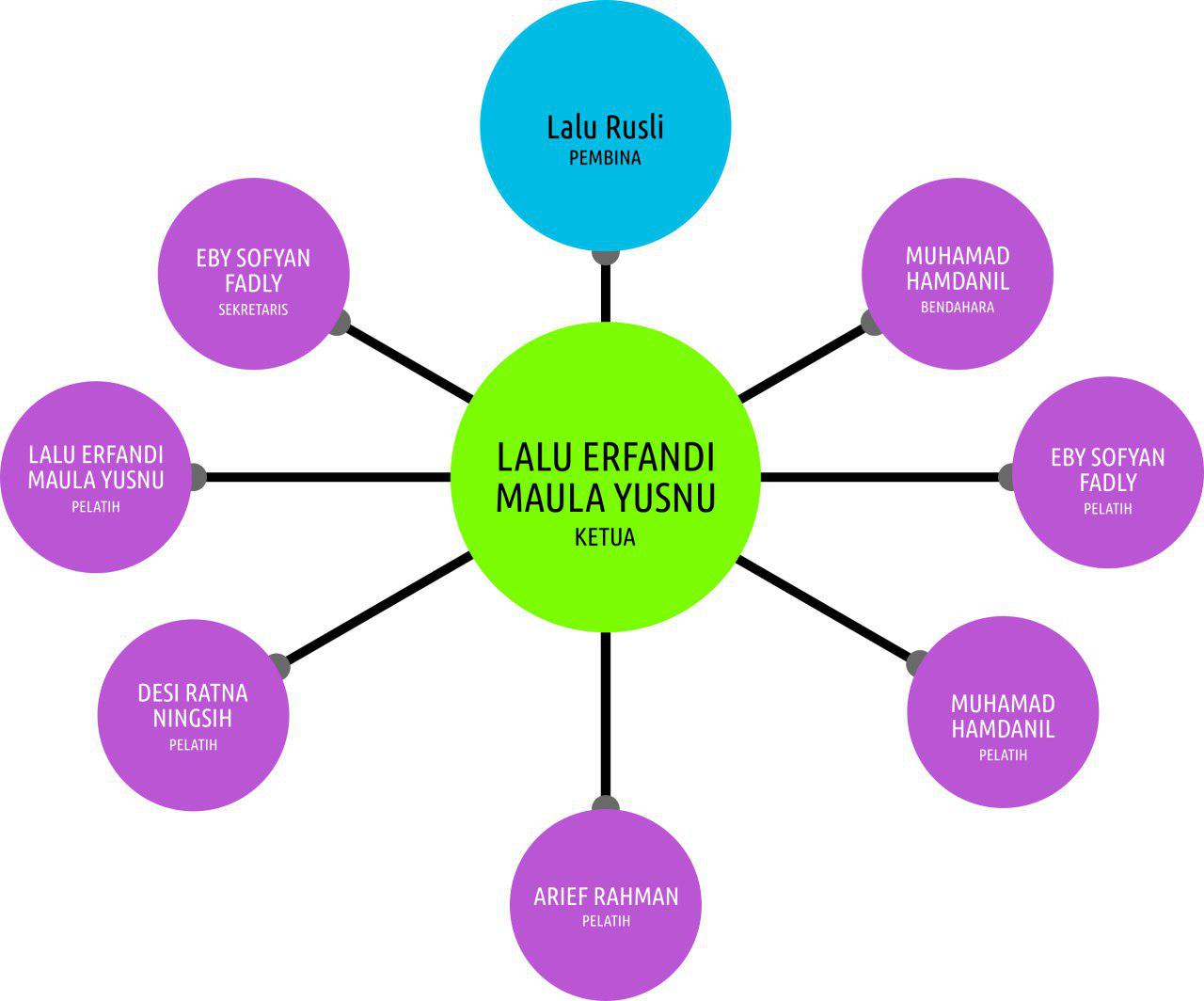
**3.1 Tinjauan Organisasi**

**3.1.1 Struktur Organisasi**

**STRUKTUR ORGANISASI**

**KLUB PANAHAN MUSLIM ARRIHU *ARCHERY***

**MATARAM NTB**

****

**Gambar 3.1 Struktur Organisasi Klub Panahan Muslim Arrihu Archery**

**3.1.2 Visi – Misi**

“Menjadi Sarana Pelatihan, Penelitian dan Pengembangan Olahraga Panahan Terbaik untuk Umat Islam Dalam Rangka Menghidupkan Sunnah Rasulullah”.

1. Mencetak Atlit panahan tradisional dan moderen yang handal dalam segala medan.

2. Melakukan penelitian dan pengembangan teknik dan peralatan panahan tradiosional dan modern.

**3.2 Metodologi**

Metodologi didefinisikan sebagai kaidah yang digunakan untuk memberikan panduan dan arah tujuan yang benar sepanjang proses pembangunan system (Jogiyanto. 2005). Berdasarkan pembahasan pada sebelumnya, bahwa metodologi yang digunakan Penulis adalah metode *Waterfall*. Metode *waterfall* dinilai cocok untuk memenuhi kebutuhan perencanaan dan analisa yang lebih detail untuk pebuatan aplikasi sistem pendukung keputusan untuk kelulusan dan perengkingan ujian kenaikan tingkat atlet panahan menggunakan metode *SMART.* rbasis *client server* serta menentukan kelulusan menggunakan metode *SMART*. Berikut penjelasan dari masing-masing tahapan dalam model *waterfall*.

1. Perancangan Sistem (*System* *Engineering*)

Proses untuk melakukan perancangan sistem penulis melihat dan mencari apa yang dibutuhkan oleh sistem yang akan dibangun serta untuk diterapkan ke dalam perangkat lunak yang akan dibuat. Untuk mendapatkan informasi yang dibutuhkan maka penulis menggunakan langkah sebagai berikut:

1. Dokumen. Penulis melakukan penelitian dengan mengumpulkan dan memanfaatkan dokumen-dokumen atau file-file mengenai kurikulum, form-form UKT, hal-hal yang dibutuhkan dalam proses UKT dan hal-hal lainnya tentang Arrihu *Archery* yang dapat membantu penulis dalam melakukan penelitian.
2. Observasi, mengunjungi langsung lokasi tempat penilitian tentang proses dan tata cara UKT pada Arrihu *Archery*.
3. Studi Literatur yaitu penelitian yang dilaksanakan dengan mempelajari konsep, metode, teknik, maupun informasi dari berbagai sumber seperti internet, buku jurnal, maupun artikel ilmiah lainnya yang berkaitan dengan aplikasi yang dibangun.
4. Analisa Kebutuhan (*Analysis*)

Setelah melakukan pengumpulan informasi, maka tahap yang akan dilakukan selanjutnya adalah menganalisa atau mengetahui ruang lingkup informasi, fungsi-fungsi yang dibutuhkan, kemajuan kinerja yang ingin dihasilkan dan perancangan antarmuka pemakai perangkat lunak (*software*).

1. Perancangan (*Design*)

Merupakan proses bertahap yang memfokuskan pada empat bagian penting, yaitu: struktur data, arsitektur perangkat lunak, detil prosedur, dan karakteristik antar muka pemakai.

1. Pengkodingan (*Coding*)

Proses penulisan bahasa pemrograman agar perangkat lunak tersebut dapat dijalankan oleh mesin atau komputer dengan menggunakan bahasa pemrograman PHP dan database *MySql*.

1. Pengujian (*Testing*)

Menguji kode program yang telah dibuat dengan tujuan untuk memastikan bahwa input yang digunakan akan menghasilkan output yang sesuai, dan apakah semua fungsi dapat dipergunakan dengan baik.

**3.3 Identifikasi Masalah**

**3.3.1** **Penjelasan Masalah**

Masalah yang terjadi di klub panahan muslim Arrihu *Archery* adalah tidak adanya suatu sistem yang menangani UKT dan tidak adanya Sumber Daya Manusia (SDM) yang membuat sistem tersebut, sehingga membutuhkan waktu lama dalam kalkulasi akumulasi hasil dari UKT didalam menentukan hasil kelulusan dan perengkingan, dari setiap nilai UKT yang diperoleh atlet, pelatih dalam mengimputkan hasil UKT masih menggunakan exel untuk menentukan kelulusan.

**3.3.4 Solusi Masalah**

Berdasarkan hasil identifikasi di lokasi penelitian klub panahan muslim Arrihu *Archery*, dokumen-dokumen serta file-file yang diperoleh dapat mendukung dalam menganalisa dan pembangunan perangkat lunak yang menerapan metode SAW dan SMART, dari kedua metode tersebut akan dianalisa manakah dari metode SAW dan SMART yang cocok untuk pembangunan sistem pendukung keputusan ujian kenaikan tingkat atlet panahan yang menentukan kelulusan dan perengkingan atlet Arrihu *Archery*.

**3.4 Analisa Kebutuhan Sistem**

Analisis kebutuhan sistem dilakukan untuk perkiraan kebutuhan sistem baru. Adapun analisis kebutuhan sistem meliputi:

1. Perangkat Keras

Untuk menjalankan sistem ini diperlukan beberapa perangkat keras yaitu:

* RAM 4GB
* *Hard disk* 500GB
* CPU *Core* 3 Duo

1. Perangkat Lunak

Untuk menjalankan sistem ini diperlukan beberapa perangkat lunak yaitu:

* Sistem Operasi Linux Ubuntu untuk *server* aplikasi *Web* *Client*.
* Ngix sebagai *Web* *Server*.
* *Database* SQL.
* *Web* Browser untuk mengakses aplikasi *Web* *Client*.

1. Pengguna

Agar perangkat lunak yang dibangun dapat berjalan dengan baik, maka diperlukan beberapa pengguna antara lain:

* *Admin*, yaitu pengguna yang menangani kebutuhan aplikasi *server control*. Kemampuan yang harus dimiliki *admin* yaitu:

1. Dapat mengoperasikan komputer dan perangkat lainnya dengan baik
2. Dapat bekerja pada sistem operasi Windows maupun Linux.
3. Dapat mengoprasikan aplikasi *server control* dengan baik dan benar.

* Pelatih dan Atlet, yaitu pengguna yang akan menggunakan aplikasi ini. Kemampuan yang harus dimiliki antara lain:

1. Dapat mengoperasikan komputer dengan baik.
2. Dapat mengoperasikan perangkat lunak *web browser* dengan baik.

**3.5 Analisa Perhitungan Manual Metode SAW**

Metode Simple Additive Weighting (SAW) sering juga dikenal istilah metode penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut (Fishburn, 1967).

**3.5.1 Kriteria dan Bobot**

Dalam metode SAW terdapat beberapa kriteria yang memiliki bobotnya masing-masing untuk mendapatkan nilai hasil seleksi para atlet. Adapun kriterianya adalah sebagai berikut:

**Tabel 3.1: Tabel Jenis Kriteria**

|  |  |
| --- | --- |
| Kriteria | Keterangan |
| C1 | Akurasi |
| C2 | Kecepatan |
| C3 | Teknik |
| C4 | Pengetahuan |
| C5 | Mental |
| C6 | Fisik |

Proses perangkingan dengan menggunakan bobot yang telah diberikan oleh pengambil keputusan.

**Tabel 3.2: Tabel Bobot Kriteria Tingkat Umum**

|  |  |
| --- | --- |
| Kriteria (C) | Bobot (W) |
| C1 | 15% (Akurasi, Ujian Bersifat Praktik) |
| C2 | 15% (Kecepatan, Ujian Bersifat Praktik) |
| C3 | 16% (Teknik, Ujian Bersifat Praktik) |
| C4 | 19% (Pengetahuan, Ujian Bersifat Teori/Tertulis) |
| C5 | 18% (Mental, Ujian Bersifat Teori/Tertulis) |
| C6 | 17% (Fisik, Ujian Bersifat Praktik) |

**Tabel 3.3: Tabel Bobot Kriteria Tingkat *Novice***

|  |  |
| --- | --- |
| Kriteria (C) | Bobot (W) |
| C1 | 15% (Akurasi, Ujian Bersifat Praktik) |
| C2 | 15% (Kecepatan, Ujian Bersifat Praktik) |
| C3 | 18% (Teknik, Ujian Bersifat Praktik) |
| C4 | 16% (Pengetahuan, Ujian Bersifat Teori/Tertulis) |
| C5 | 19% (Mental, Ujian Bersifat Teori/Tertulis) |
| C6 | 17% (Fisik, Ujian Bersifat Praktik) |

**Tabel 3.4: Tabel Bobot Kriteria Tingkat *Intermediate***

|  |  |
| --- | --- |
| Kriteria (C) | Bobot (W) |
| C1 | 17% (Akurasi, Ujian Bersifat Praktik) |
| C2 | 15% (Kecepatan, Ujian Bersifat Praktik) |
| C3 | 19% (Teknik, Ujian Bersifat Praktik) |
| C4 | 15% (Pengetahuan, Ujian Bersifat Teori/Tertulis) |
| C5 | 16% (Mental, Ujian Bersifat Teori/Tertulis) |
| C6 | 18% (Fisik, Ujian Bersifat Praktik) |

**Tabel 3.5: Tabel Bobot Kriteria Tingkat *Advance***

|  |  |
| --- | --- |
| Kriteria (C) | Bobot (W) |
| C1 | 15% (Akurasi, Ujian Bersifat Praktik) |
| C2 | 16% (Kecepatan, Ujian Bersifat Praktik) |
| C3 | 17% (Teknik, Ujian Bersifat Praktik) |
| C4 | 19% (Pengetahuan, Ujian Bersifat Teori/Tertulis) |
| C5 | 18% (Mental, Ujian Bersifat Teori/Tertulis) |
| C6 | 15% (Fisik, Ujian Bersifat Praktik) |

**Tabel 3.6: Tabel Bobot Kriteria Tingkat *Warden***

|  |  |
| --- | --- |
| Kriteria (C) | Bobot (W) |
| C1 | 15% (Akurasi, Ujian Bersifat Praktik) |
| C2 | 16% (Kecepatan, Ujian Bersifat Praktik) |
| C3 | 17% (Teknik, Ujian Bersifat Praktik) |
| C4 | 19% (Pengetahuan, Ujian Bersifat Teori/Tertulis) |
| C5 | 18% (Mental, Ujian Bersifat Teori/Tertulis) |
| C6 | 15% (Fisik, Ujian Bersifat Praktik) |

**3.5.2 Perhitungan Menggunakan Metode SAW**

Adapun langkah-langkah teknik perhitungan metode SAW, yaitu sebagai berikut:

**Langkah 1. Menentukan Kriteria dan Bobot**

Pada metode SMART, langkah pertama yang dibutuhkan yaitu menentukan jumlah kriteria. Penulis akan mengambil sampel kriteria tingkat umum, jumlah kriteria yang ditentukan penulis yaitu sebanyak 6 (enam) kriteria. Adapun jenis dari 6 kriteria tersebut adalah sebagai berikut:

**Tabel 3.7: Tabel Langkah 1**

|  |  |  |
| --- | --- | --- |
| Kriteria | Keterangan | Bobot |
| C1 | Akurasi, Ujian Bersifat Praktik | 15% |
| C2 | Kecepatan, Ujian Bersifat Praktik | 15% |
| C3 | Teknik, Ujian Bersifat Praktik | 16% |
| C4 | Pengetahuan, Ujian Bersifat Teori/Tertulis | 19% |
| C5 | Mental, Ujian Bersifat Teori/Tertulis | 18% |
| C6 | Fisik, Ujian Bersifat Praktik | 17% |

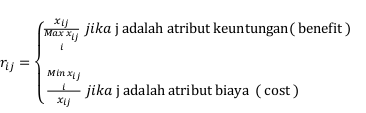
**Langkah 2. Menentukan Variable Nilai Kriteria**

**Tabel 3.8: Tabel Langkah 1**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Alternatif | C1 | C2 | C3 | C4 | C5 | C6 |
| 1. | CA0001 | 80 | 70 | 60 | 80 | 70 | 90 |
| 2. | CA0002 | 90 | 70 | 80 | 60 | 80 | 70 |
| 3. | CA0003 | 80 | 70 | 60 | 80 | 90 | 60 |

Dari tabel diatas data dapat diubah kedalam matriks keputusan X sebagai berikut :

Menormalisasikan matrix X menjadi r berdasarkan rumus sebagai berikut:

(**1**)

Ket:

* *Rij* = nilai rating kinerja normalisasi
* *Xij* = nilai atribut yang dimiliki dari setiap kriteria
* *Max Xij* = nilai terbesar dari setiap kriteria
* *Min Xij* = nilai terkecil dari setiap kriteria
* Benefit = nilai terbesar adalah terbaik
* Cost = nilai terkecil adalah terbaik

Penyelesaiaan

**Tabel 3.9: Tabel Langkah Normalisasi**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| r11 = = 0.8 | r12 = = 1 | r13 = = 0.7 | r14 = = 1 | r15 = = 0.7 | R16 = = 1 |
| r21 = = 1 | R22 = = 1 | R23 = = 1 | r 24= = 0.7 | R25 = = 0.8 | R26 = = 0.7 |
| r31 = = 0.8 | R32 = = 1 | R33 = = 0.7 | R34 = = 1 | R35 = = 1 | R36 = = 0.6 |

Matrix R:

**Tabel 3.10: Tabel Matrix R**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Alternatif | C1 | C2 | C3 | C4 | C5 | C6 |
| 1. | CA0001 | 0.8 | 1 | 0.7 | 1 | 0.7 | 1 |
| 2. | CA0002 | 1 | 1 | 1 | 0.7 | 0.8 | 0.7 |
| 3. | CA0003 | 0.8 | 1 | 0.7 | 1 | 1 | 0.6 |

**Langkah 3. Melakukan proses perhitungan kelulusan dan perengkingan dari setiap atlet**

Langkah ketiga yaitumelakukan proses perhitungan kelulusan dan perengkingan dari kriteria yang sudah dihitung dari proses sebeumnya. Adapun rumus yang dingunakan sebagai berikut:

(**2**)

Ket:

* *Vi* = rangking untuk setiap alternatif
* *Wij* = nilai bobot dari setiap kriteria
* *rij* = nilai rating kinerja ternormalisasi

Penyelesaian:

V1 = 0.8(15) + 1(15) + 0.7(16) + 1(19) + 0.7(18) + 1(17)

= 12 + 15 + 11.2 + 19 + 12.6 + 17

= 86.8

V1 = 1(15) + 1(15) + 1(16) + 0.7(19) + 0.8(18) + 1(17)

= 15 + 15 + 16 + 13.5 + 14.4 +17

= 90.9

V1 = 0.8(15) + 1(15) + 0.7(16) + 1(19) + 0.7(18) + 1(17)

= 12 + 15 + 16.7 + 19 + 12.6 +17

= 92.3

Dari nilai akhir yang didapat, dan yang memiliki nilai paling besar yaitu dengan nilai akhir 92.3 dengan alternatif CA0003 dibandingkan dengan alternative lainnya, dan semua alternative dinyatakan lulus, minimum nilai kelulusan adalah di atas 70.

**3.6 Analisa Perhitungan Manual Metode SMART**

Dalam proses penentuan penilaian rangking dari UKT ini dengan menggunakan salah satu metode dalam sistem pendukung keputusan yaitu metode *Simple Multy Attribute Rating Technique* (SMART). Pada hakekatnya SMART merupakan suatu model pengambilan keputusan yang komprehensif dengan memperhitungkan hal-hal yang bersifat kualitatif dan kuantitatif.

**3.5.1 Kriteria dan Bobot**

Dalam metode SMART dibutuhkan beberapa kriteria yang memiliki bobotnya masing-masing untuk mendapatkan nilai hasil seleksi para atlet. Adapun kriterianya adalah sebagai berikut:

**Tabel 3.11: Tabel Jenis Kriteria**

|  |  |
| --- | --- |
| Kriteria | Keterangan |
| C1 | Akurasi |
| C2 | Kecepatan |
| C3 | Teknik |
| C4 | Pengetahuan |
| C5 | Mental |
| C6 | Fisik |

Proses perangkingan dengan menggunakan bobot yang telah diberikan oleh pengambil keputusan.

**Tabel 3.12: Tabel Bobot Kriteria Tingkat Umum**

|  |  |
| --- | --- |
| Kriteria (C) | Bobot (W) |
| C1 | 15% (Akurasi, Ujian Bersifat Praktik) |
| C2 | 15% (Kecepatan, Ujian Bersifat Praktik) |
| C3 | 16% (Teknik, Ujian Bersifat Praktik) |
| C4 | 19% (Pengetahuan, Ujian Bersifat Teori/Tertulis) |
| C5 | 18% (Mental, Ujian Bersifat Teori/Tertulis) |
| C6 | 17% (Fisik, Ujian Bersifat Praktik) |

**Tabel 3.13: Tabel Bobot Kriteria Tingkat *Novice***

|  |  |
| --- | --- |
| Kriteria (C) | Bobot (W) |
| C1 | 15% (Akurasi, Ujian Bersifat Praktik) |
| C2 | 15% (Kecepatan, Ujian Bersifat Praktik) |
| C3 | 18% (Teknik, Ujian Bersifat Praktik) |
| C4 | 16% (Pengetahuan, Ujian Bersifat Teori/Tertulis) |
| C5 | 19% (Mental, Ujian Bersifat Teori/Tertulis) |
| C6 | 17% (Fisik, Ujian Bersifat Praktik) |

**Tabel 3.14: Tabel Bobot Kriteria Tingkat *Intermediate***

|  |  |
| --- | --- |
| Kriteria (C) | Bobot (W) |
| C1 | 17% (Akurasi, Ujian Bersifat Praktik) |
| C2 | 15% (Kecepatan, Ujian Bersifat Praktik) |
| C3 | 19% (Teknik, Ujian Bersifat Praktik) |
| C4 | 15% (Pengetahuan, Ujian Bersifat Teori/Tertulis) |
| C5 | 16% (Mental, Ujian Bersifat Teori/Tertulis) |
| C6 | 18% (Fisik, Ujian Bersifat Praktik) |

**Tabel 3.15: Tabel Bobot Kriteria Tingkat *Advance***

|  |  |
| --- | --- |
| Kriteria (C) | Bobot (W) |
| C1 | 15% (Akurasi, Ujian Bersifat Praktik) |
| C2 | 16% (Kecepatan, Ujian Bersifat Praktik) |
| C3 | 17% (Teknik, Ujian Bersifat Praktik) |
| C4 | 19% (Pengetahuan, Ujian Bersifat Teori/Tertulis) |
| C5 | 18% (Mental, Ujian Bersifat Teori/Tertulis) |
| C6 | 15% (Fisik, Ujian Bersifat Praktik) |

**Tabel 3.16: Tabel Bobot Kriteria Tingkat *Warden***

|  |  |
| --- | --- |
| Kriteria (C) | Bobot (W) |
| C1 | 15% (Akurasi, Ujian Bersifat Praktik) |
| C2 | 16% (Kecepatan, Ujian Bersifat Praktik) |
| C3 | 17% (Teknik, Ujian Bersifat Praktik) |
| C4 | 19% (Pengetahuan, Ujian Bersifat Teori/Tertulis) |
| C5 | 18% (Mental, Ujian Bersifat Teori/Tertulis) |
| C6 | 15% (Fisik, Ujian Bersifat Praktik) |

**3.5.2 Perhitungan Menggunakan Metode SMART**

Adapun langkah-langkah teknik perhitungan metode SMART, yaitu sebagai berikut:

**Langkah 1. Menentukan Jumlah Kriteria**

Pada metode SMART, langkah pertama yang dibutuhkan yaitu menentukan jumlah kriteria. Penulis akan mengambil sampel kriteria tingkat umum, jumlah kriteria yang ditentukan penulis yaitu sebanyak 6 (enam) kriteria. Adapun jenis dari 6 kriteria tersebut adalah sebagai berikut:

**Tabel 3.17: Tabel Langkah 1**

|  |  |  |
| --- | --- | --- |
| Kriteria | Keterangan | Bobot |
| C1 | Akurasi, Ujian Bersifat Praktik | 15% |
| C2 | Kecepatan, Ujian Bersifat Praktik | 15% |
| C3 | Teknik, Ujian Bersifat Praktik | 16% |
| C4 | Pengetahuan, Ujian Bersifat Teori/Tertulis | 19% |
| C5 | Mental, Ujian Bersifat Teori/Tertulis | 18% |
| C6 | Fisik, Ujian Bersifat Praktik | 17% |

**Langkah 2. Hitung Normalisasi Bobot**

Langkah kedua menghitung nilai normalisasi dari bobot-bobot yang sudah ditentukan oleh penulis. Adapun rumus yang digunakan untuk menghitung normalisasi bobot yaitu sebagai berikut:



**(1)**

Ket:

* *nwj* = normalisasi bobot kriteria ke-j
* k = jumlah kriteria
* *wn* = bobot kriteria ke-n

**Tabel 3.18: Tabel Langkah 2**

|  |  |  |
| --- | --- | --- |
| Kriteria | Bobot | Normalisasi |
| C1 | 15% | C1 = 0,15  0,15 + 0,15 + 0,16 + 0,19 + 0,18 + 0,17  = 0,15  1  = 0,15 |
| C2 | 15% | C2 = 0,15  0,15 + 0,15 + 0,16 + 0,19 + 0,18 + 0,17  = 0,15  1  = 0,15 |
| C3 | 16% | C3 = 0,16  0,15 + 0,15 + 0,16 + 0,19 + 0,18 + 0,17  = 0,16  1  = 0,16 |
| C4 | 19% | C4 = 0,19  0,15 + 0,15 + 0,16 + 0,19 + 0,18 + 0,17  = 0,19  1  = 0,19 |
| C5 | 18% | C4 = 0,18  0,15 + 0,15 + 0,16 + 0,19 + 0,18 + 0,17  = 0,18  1  = 0,18 |
| C6 | 17% | C4 = 0,17  0,15 + 0,15 + 0,16 + 0,19 + 0,18 + 0,17  = 0,17  1  = 0,17 |

**Langkah 3. Memberikan nilai kriteria untuk masing-masing *alternative* (atlet)**

Pada langkah ketiga, penulis menentukan jumlah *alternative* (atlet) yang akan diujikan serta penulis memberikan nilai kriteria untuk masing-masing atlet. Nilai pada tabel (Tabel 3.9) di bawah, didapatkan dari hasil kalkulasi akumulasi nilai atlet dari masing-masing kriteria.

**Tabel 3.19: Tabel Langkah 3**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Alternatif | C1 | C2 | C3 | C4 | C5 | C6 |
| 1. | CA0001 | 80 | 70 | 60 | 80 | 70 | 90 |
| 2. | CA0002 | 90 | 70 | 80 | 60 | 80 | 70 |
| 3. | CA0003 | 80 | 70 | 60 | 80 | 90 | 60 |

**Langkah 4. Hitung nilai utility untuk setiap nilai kriteria**

Langkah keempat yaitu menghitung nilai *utility* setiap nilai kriteria yang sudah ditentukan masing-masing. Adapun rumus yang digunakan untuk menghitung utilitas nilai kriteria yaitu sebagai berikut:



**(2)**

Ket:

* *ui(ai)* = nilai utility kriteria ke-j untuk atlet ke-i
* *cmax* = nilai kriteria maksimal
* *cmin* = nilai kriteria minimal
* *cout i* = nilai kriteria ke-i

**Tabel 3.20: Tabel Langkah 4**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Alternatif | C1 | C2 | C3 | C4 | C5 | C6 |
| 1. | CA0001 | 0,3 | 0,6 | 1 | 0,3 | 0,6 | 0 |
| 2. | CA0002 | 0 | 0,6 | 0,3 | 1 | 0,3 | 0,6 |
| 3. | CA0003 | 0,3 | 0,6 | 1 | 0,3 | 0 | 1 |

**Langkah 5. Hitung nilai akhir**

Untuk menentukan rangking akhir dari perhitungan menggunakan metode *SMART*, yaitu dengan cara nilail normalisasi kriteria masing-masing dikalikan dengan nilai utilitas masing-masing *alternative*. Adapun rumus yang digunakan yaitu sebagai berikut beserta tabel hasil perhitungan nilai akhir:

**(3)**



**Tabel 3.21: Tabel Langkah 5**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Alternatif | C1 | C2 | C3 | C4 | C5 | C6 | Total Nilai |
| 1. | CA0001 | 0,05 | 0,1 | 0,16 | 0,06 | 0,12 | 0 | 0,49 |
| 2. | CA0002 | 0 | 0,1 | 0,05 | 0,19 | 0,06 | 0,11 | 0,51 |
| 3. | CA0003 | 0,05 | 0,1 | 0,16 | 0,63 | 0 | 0,17 | 0,54 |

Dari nilai akhir yang didapat, dan yang memiliki nilai paling besar yaitu dengan nilai akhir 0,54 dengan alternatif CA0003 dibandingkan dengan alternative lainnya.

**3.6 Analisa Data**

**3.6.1 Struktur *Database***

1. Tabel u*ser*

Tabel 3.22 Struktur Tabel *user*

|  |  |  |  |
| --- | --- | --- | --- |
| Nama *Field* | Tipe | Ukuran | Ket |
| id | Int | 11 | id *user* |
| password | Varchar | 128 | *Password user* |
| is\_superuser | TinyInt | 1 | Hak akses admin |
| username | Varchar | 150 | username |
| is\_staff | TinyInt | 1 | Hak akses user |

2. Tabel *profile*

Tabel 3.23 Struktur Tabel *profile*

|  |  |  |  |
| --- | --- | --- | --- |
| Nama *Field* | Tipe | Ukuran | Ket |
| id | Int | 11 | Id *profile* |
| created | DateTime | - | Tanggal data dibuat |
| modified | DateTime | - | Tanggal data diubah |
| name | Varchar | 255 | Nama *profile* |
| address | LongText | - | Alamat *profile* |
| gender | Varchat | 25 | Gender *profile* |
| born\_place | Varchar | 100 | Tempat lahir *profile* |
| born\_date | Date | - | Tanggal lahir *profile* |
| religion | Varchar | 100 | Agama *profile* |
| identity\_number | Varchar | 100 | Nomer identitas *profile* |
| phone | Varchar | 15 | Nomer telpon profile |
| email | Varchar | 100 | Email *profile* |
| picture | Varchar | 100 | Gambar *profile* |
| level\_id | Int | 11 | *Level* id *profile* |
| organization\_id | Int | 11 | Organisasi *profile* |
| physic\_id | Int | 11 | Id *physic profile* |
| user\_id | Int | 11 | Id *user* |

3. Tabel *physic*

Tabel 3.24 Struktur Tabel *physic*

|  |  |  |  |
| --- | --- | --- | --- |
| Nama *Field* | Tipe | Ukuran | Ket |
| id | Int | 11 | id *physic* |
| body\_weight | Varchar | 5 | Berat badan atlet |
| body\_height | Varchar | 5 | Tinggi badan atlet |
| arm\_span | Varchar | 5 | *Arm span* atlet |
| full\_draw | Varchar | 5 | *Full draw* atlet |
| over\_draw | Varchar | 5 | *Over draw* atlet |
| hospital\_history | LongText | - | Riwayat penyakit atlet |
| blood\_group | Varchar | 5 | Golongan darat atlet |

4. Tabel *organization*

Tabel 3.25 Struktur Tabel *organization*

|  |  |  |  |
| --- | --- | --- | --- |
| Nama *Field* | Tipe | Ukuran | Ket |
| id | Int | 11 | Id *organization* |
| created | DateTime | - | Tanggal data dibuat |
| modified | DateTime | - | Tanggal data diubah |
| name | Varchar | 100 | Nama organisasi |
| description | LongText | - | Deskripsi organisasi |

5. Tabel *leveling*

Tabel 3.26 Struktur Tabel *leveling*

|  |  |  |  |
| --- | --- | --- | --- |
| Nama *Field* | Tipe | Ukuran | Ket |
| id | Int | 11 | Id *level* |
| created | DateTime | - | Tanggal data dibuat |
| modified | DateTime | - | Tanggal data diubah |
| accuracy\_point | Double | - | Poin dari akurasi |
| speed\_point | Double | - | Poin dari kecepatan |
| technique\_point | Double | - | Poin dari teknik |
| knowledge\_point | Double | - | Poin dari penetahuan |
| physic\_point | Double | - | Poin dari fisik |
| mental\_point | Double | - | Poin dari mental |
| lev\_info\_id | Int | 11 | *Leveling* info |
| profile\_id | Int | 11 | Profile id |

6. Tabel *division*

Tabel 3.27 Struktur Tabel *devision*

|  |  |  |  |
| --- | --- | --- | --- |
| Nama *Field* | Tipe | Ukuran | Ket |
| id | Int | 11 | Id *devision* |
| created | DateTime | - | Tanggal data dibuat |
| modified | DateTime | - | Tanggal data diubah |
| name | Varchar | 100 | Nama devisi |
| description | LongText | - | Deskripsi devisi |

7. Tabel *leveling\_info*

Tabel 3.28 Struktur Tabel *leveling\_info*

|  |  |  |  |
| --- | --- | --- | --- |
| Nama *Field* | Tipe | Ukuran | Ket |
| id | Int | 11 | Id *leveling* |
| created | DateTime | - | Tanggal data dibuat |
| modified | DateTime | - | Tanggal data diubah |
| location | Varchar | 255 | Lokasi *leveling* |
| accuracy\_weight | Double | - |  |
| speed\_weight | Double | - |  |
| technique\_weight | Double | - |  |
| knowledge\_weight | Double | - |  |
| physic\_weight | Double | - |  |
| mental\_weight | Double | - |  |
| division | Varchar | 100 |  |
| level | Varchar | 100 |  |
| pass\_score | Double | - |  |
| date | Date | - | Tanggal *leveling* |

8. Tabel *settings*

Tabel 3.29 Struktur Tabel *setting*

|  |  |  |  |
| --- | --- | --- | --- |
| Nama *Field* | Tipe | Ukuran | Ket |
| id | Int | 11 | Id *settings* |
| created | DateTime | - | Tanggal data dibuat |
| modified | DateTime | - | Data data diubah |
| property | Varchar | 100 | Property *settings* |
| value | Varchar | 100 | Nilai Settings |

9. Tabel *level*

Tabel 3.30 Struktur Tabel *level*

|  |  |  |  |
| --- | --- | --- | --- |
| Nama *Field* | Tipe | Ukuran | Ket |
| id | Int | 11 | Id *settings* |
| created | DateTime | - | Tanggal data dibuat |
| modified | DateTime | - | Data data diubah |
| name | Varchar | 100 | Nama level |
| description | LongText | - | Description level |
| division\_id | Int | 11 | Division\_id level |

10. Tabel *assesment\_weight*

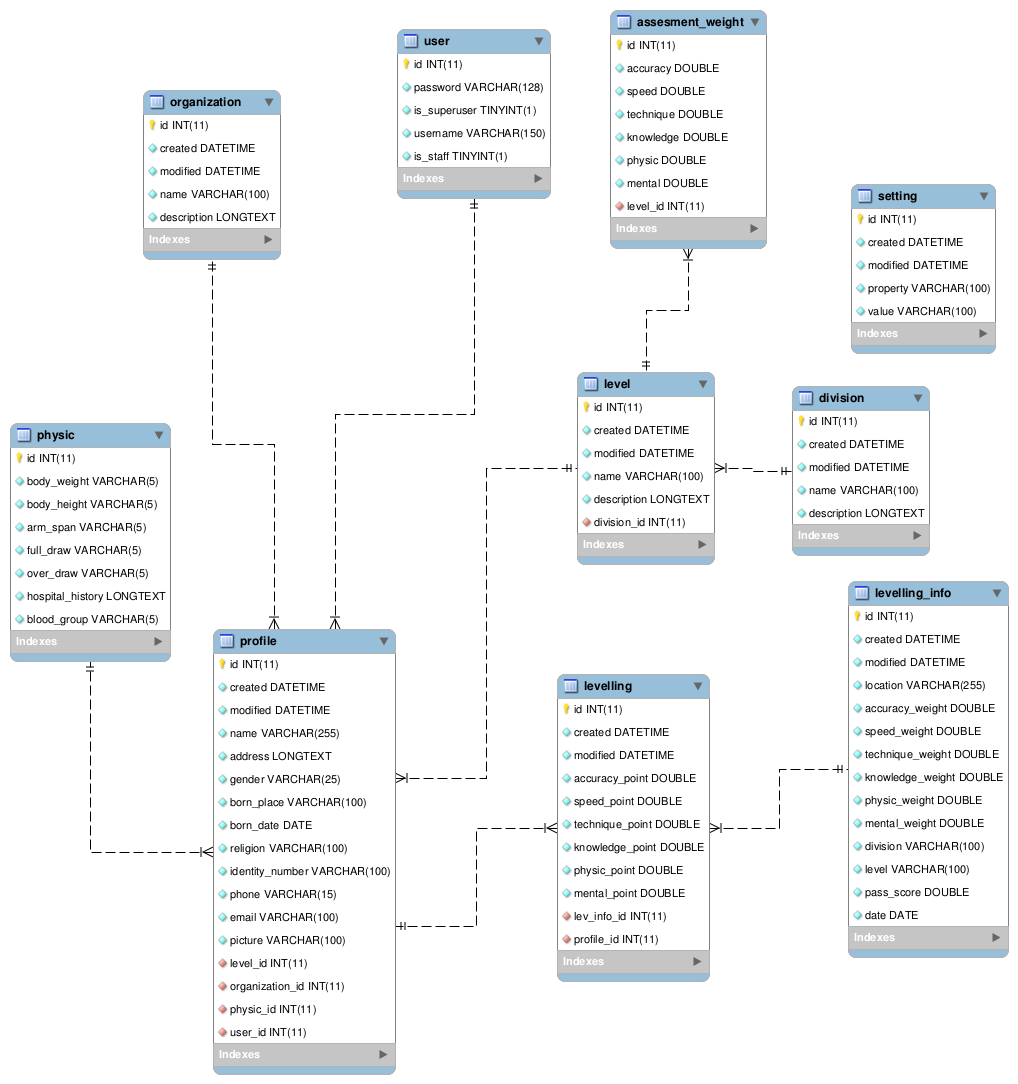
Tabel 3.31 Struktur Tabel assessment\_weight

|  |  |  |  |
| --- | --- | --- | --- |
| Nama *Field* | Tipe | Ukuran | Ket |
| id | Int | 11 | Id *assesment\_weight* |
| accuracy | Double |  |  |
| speed | Double |  |  |
| technique | Double |  |  |
| knowledge | Double |  |  |
| physic | Double |  |  |
| mental | Double |  |  |
| level\_id | Int | 11 | level\_id level |

**3.6.2 Entity Relationship Diagram (ERD)**

Pada diagram relasi entitas ini menggambarkan bagaimana tabel-tabel dalam *database* berelasi antara tabel yang satu dengan tabel yang lain.

Adapun diagram relasi entitas dapat digambarkan sebagai berikut:

****

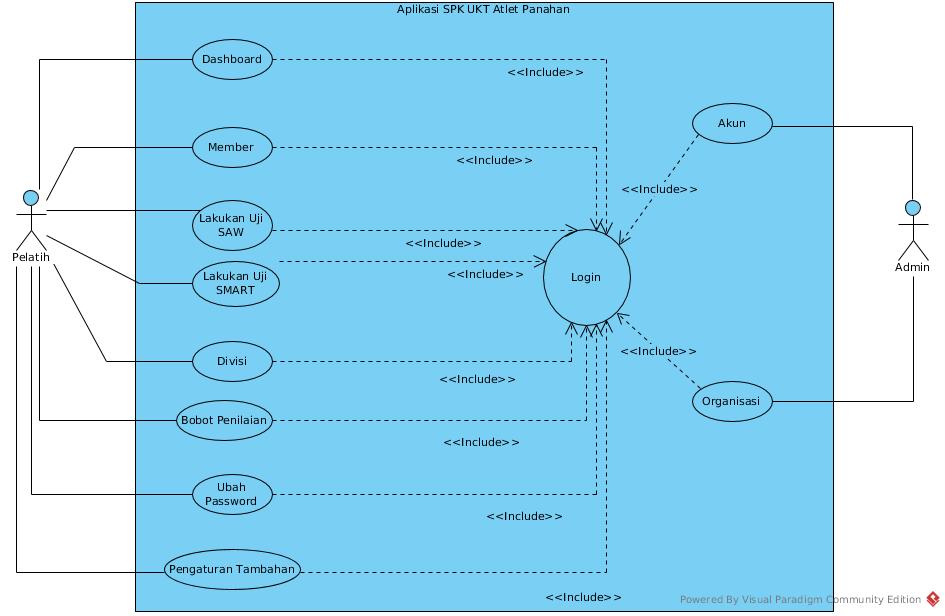
Gambar 3.2 Entity Relationship Diagram (ERD)

**3.7 Design Sistem**

Pada desain sistem ini akan menjelaskan mengenai Desain Sistem, , *Use Case Diagram Class Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Desain Interface*  pada aplikasi *Web* SPK.

**3.7.1 Use Case Diagram**

Dalam perancangan sistem ini penulis menggambarkan *use case* diagram sebagai beriut :

****

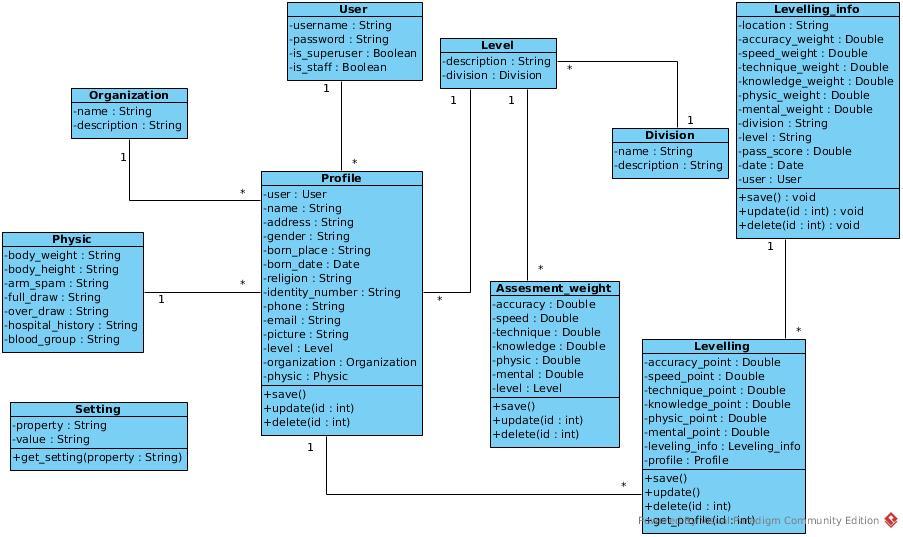
Gambar 3.3 Use Case Diagram

Keterangan:

1. Aktor yang berperan dalam sistem ini adalah Pelatih dan Admin
2. Include pada gambar artinya Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya. Jadi untuk bisa mengakses aplikasi, Pelatih dan Admin harus melakukan *login*.
3. Setelah *login*, Pelatin bisa melakukan melihat dashboard statistic para atlet, menambahkan member baru, melakukan UKT sampai melakukan pengaturan akun.
4. Setelah *login*, Admin bisa melakukan mengakses pemanbahan organisasi dan pembuatan akun baru.

**3.7.2 Class Diagram**

Dalam per*model*an sistem ini, terbentuk *class* digram sebagai berikut:

****

**Gambar 3.3 Use Class Diagran**

Keterangan:

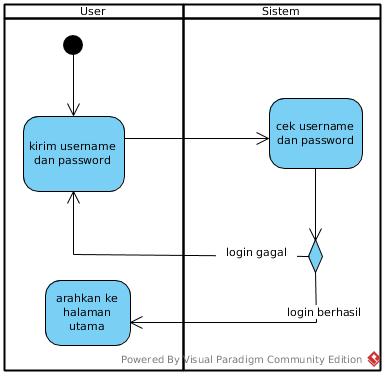
* Class *User*, *Level, Organization, Physic dan Levelying* akan meng-*extend* class *Profile*, sehingga masing-masing *profile* akan sebagai salah satu *User*, *Level, Organization, Physic dan Levelying*.
* Class *Division* dan *Assessment\_weight* akan meng*extend* ke class *Level*, sehingga *Division* dan *Assessment\_weight* dapat mengakses *Level*.
* Class *Levelying\_info* akan meng-*extend* ke class *Level*, sehingga Levelying\_info bias memiliki data dari *Level*.

**3.7.3 *Activity* Diagram**

Dalam perancangan sistem ini penulis menggambarkan *activity* diagram sebagai beriut :

* + - * 1. *Activity* Diagram proses *login*

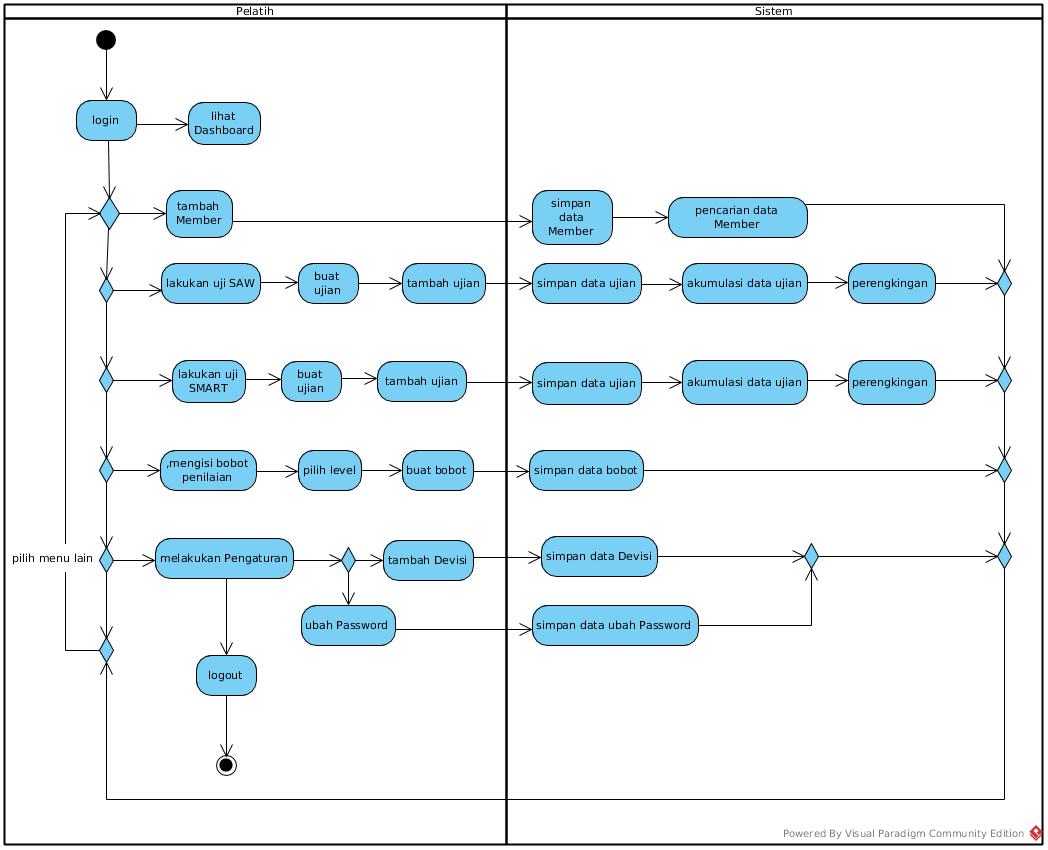
*User* melakukan *login* dengan mengirimkan *username* dan *password*, kemudian sistem akan mengecek apakah *login* berhasil atau gagal, jika *login* berhasil maka *user* akan diarahkan ke halaman utama.

****

Gambar 3.4 *Activity* Diagram proses *Login*

* 1. *Activity* Diagram pada Pelatih

Pelatih harus melakukan *login* terlebih dahulu sebelum berinteraksi dengan sistem, aktifitas utama yang bisa dilakukan Pelatih terhadap sistem adalah Pelatin bisa melakukan melihat dashboard statistic para atlet, menambahkan member baru, melakukan UKT sampai melakukan pengaturan akun.

****

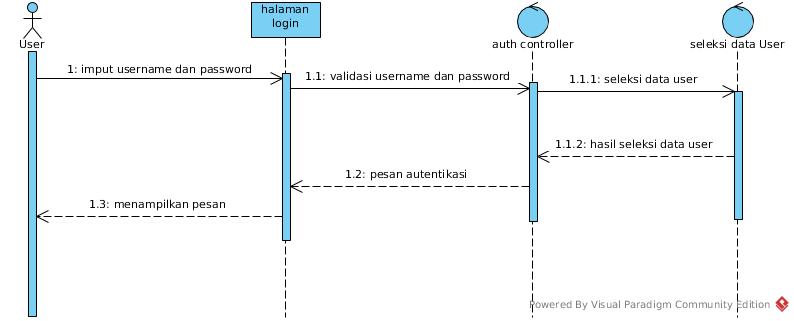
**Gambar 3.4 *Activity* Diagram Pelatih**

**3.7.4 Sequence Diagram**

Dalam perancangan sistem ini penulis menggambarkan sequence diagram sebagai beriut :

1. Squence Diagram proses *login*

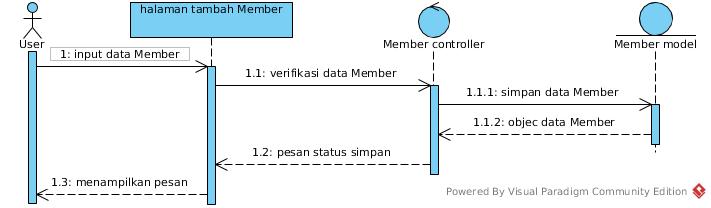
*User* mengirim *request data* berupa *username* dan *password*, kemudian data yang dikirim akan ditangkap oleh bagian *Controller* untuk divalidasi dengan seleksi data user yang terhubung ke entitas *database*. Jika *login* berhasil, maka *Control* akan mengembalikan *response*  ke *user*.

****

**Gambar 3.5 Sequence Diagram Proses *Login***

1. Squence Diagram proses tambahMember

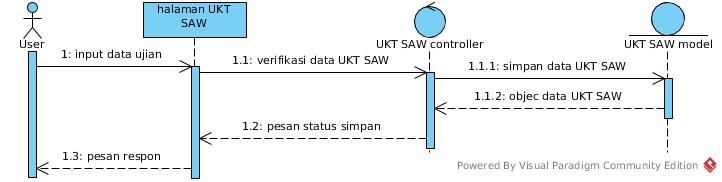
*User* yang *login* sebagai pelatih mengirim data *resuest* berupa data member, kemudian data yang dikirim akan ditangkap oleh bagian *Controller* untuk divalidasi dengan member model. Jika proses menambahkan member berhasil, maka sistem akan mengembalikan respon.

****

**Gambar 3.6 Sequence Diagram Proses Tambah *Member***

1. Squence Diagram proses Ujian Kenaikan Tingkat menggunakan metode SAW

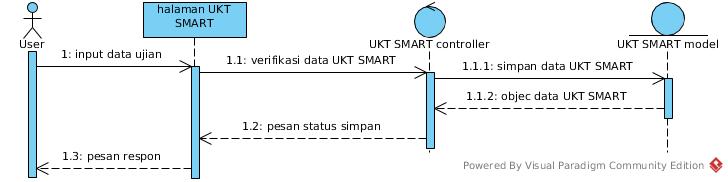
*User* yang *login* sebagai pelatih mengirim data *resuest* berupa data UKT, kemudian data yang dikirim akan ditangkap oleh bagian *Controller* untuk divalidasi dengan UKT SAW model. Jika proses UKT dengan metode SAW berhasil, maka sistem akan mengembalikan respon.

****

**Gambar 3.7 Sequence Diagram Proses UKT metode SAW**

1. Squence Diagram proses Ujian Kenaikan Tingkat menggunakan metode SMART

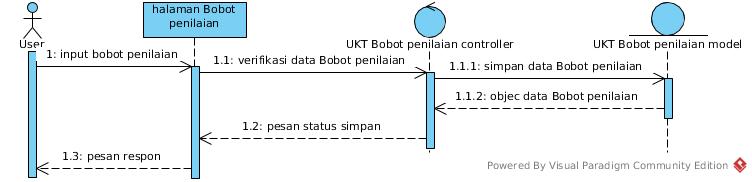
*User* yang *login* sebagai pelatih mengirim data *resuest* berupa data UKT, kemudian data yang dikirim akan ditangkap oleh bagian *Controller* untuk divalidasi dengan UKT SMART model. Jika proses UKT menggunakan metode SMART berhasil, maka sistem akan mengembalikan respon.

****

**Gambar 3.8 Sequence Diagram Proses UKT metode SMART**

1. Squence Diagram proses Bobot Penilaian

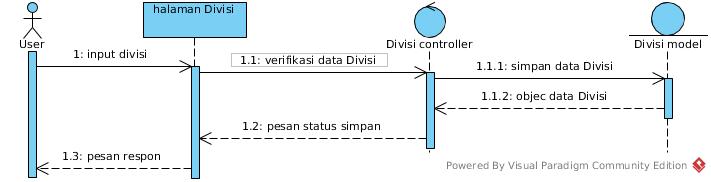
*User* yang *login* sebagai pelatih mengirim data *resuest* berupa data bobot penilaian, kemudian data yang dikirim akan ditangkap oleh bagian *Controller* untuk divalidasi dengan bobot penilaian model. Jika proses menambahkan bobot penilaian berhasil, maka sistem akan mengembalikan respon.

****

**Gambar 3.8 Sequence Diagram Proses Bobot Penilaian**

1. Squence Diagram proses Penambahan Divisi

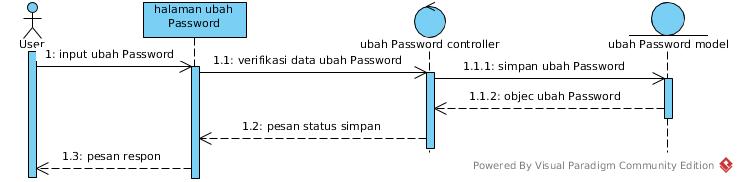
*User* yang *login* sebagai pelatih mengirim data *resuest* berupa data penambahan divisi, kemudian data yang dikirim akan ditangkap oleh bagian *Controller* untuk divalidasi dengan divisi model. Jika proses penambahan divisi berhasil, maka sistem akan mengembalikan respon.

****

**Gambar 3.9 Sequence Diagram Proses Penambahan Divisi**

1. Squence Diagram proses Ubah Password

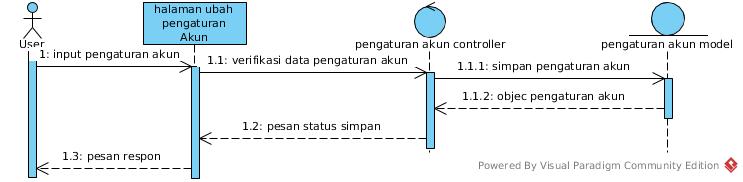
*User* yang *login* sebagai pelatih mengirim data *resuest* berupa data ubah password, kemudian data yang dikirim akan ditangkap oleh bagian *Controller* untuk divalidasi dengan ubah password model. Jika proses ubah password berhasil, maka sistem akan mengembalikan respon.

****

**Gambar 3.9 Sequence Diagram Proses Ubah Password**

1. Squence Diagram proses Pengaturan Tambahan

*User* yang *login* sebagai pelatih mengirim data *resuest* berupa data pengaturan akun, kemudian data yang dikirim akan ditangkap oleh bagian *Controller* untuk divalidasi dengan pengaturan akun model. Jika proses pengaturan akun berhasil, maka sistem akan mengembalikan respon.

****

**Gambar 3.10 Sequence Diagram Proses Pengaturan Tambahan**

**3.7.5 Desain *Interface***

Adapun rancangan *Interface*  perangkat lunak ini secara garis besar dirancang akan mengakses data-data dari *server*, rancangan *Interface*  ini akan digunakan oleh Pelatih dan Admin. Berikut ini merupakan rancangan *Interface*  pada aplikasi web:

* + 1. Desain halaman *login*

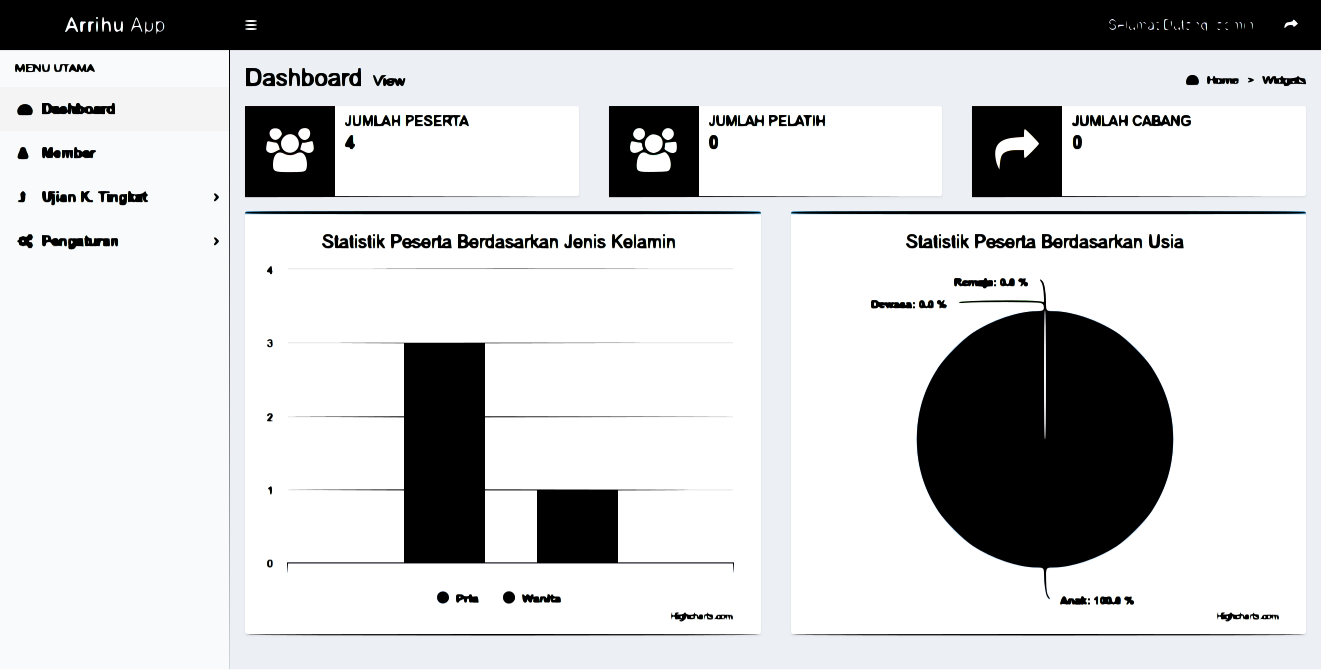
Pada halaman *login* ini menyediakan *form* yag memiliki *input*an berupa *username* dan *password* identitas pengguna. Halaman *login* ini digunakan oleh Pelatih dan Admin sebelum menjalankan aplikasi. Berikut desain halaman *login*:



**Gambar 3.11 Desain Halaman *Login***

* + 1. Desain halaman dashboard

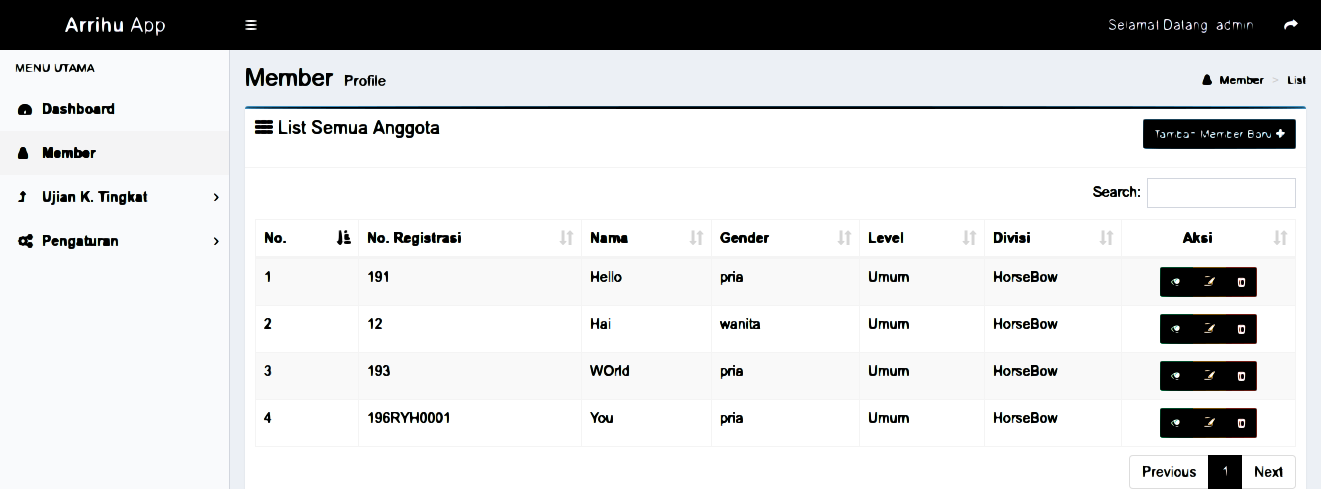
Setelah *login* berhasil, akan menampilkan halaman *dashboard.* Pada halaman *dashboard* ini menampilkan grafik statistik peserta berdasarkan jenis kelamin dan statistik berdasarkan usia dari database Arrihu *Archery*. Berikut desain halaman *dashboard*:



**Gambar 3.12 Desain Halaman *Dashboard***

* + 1. Desain halaman member

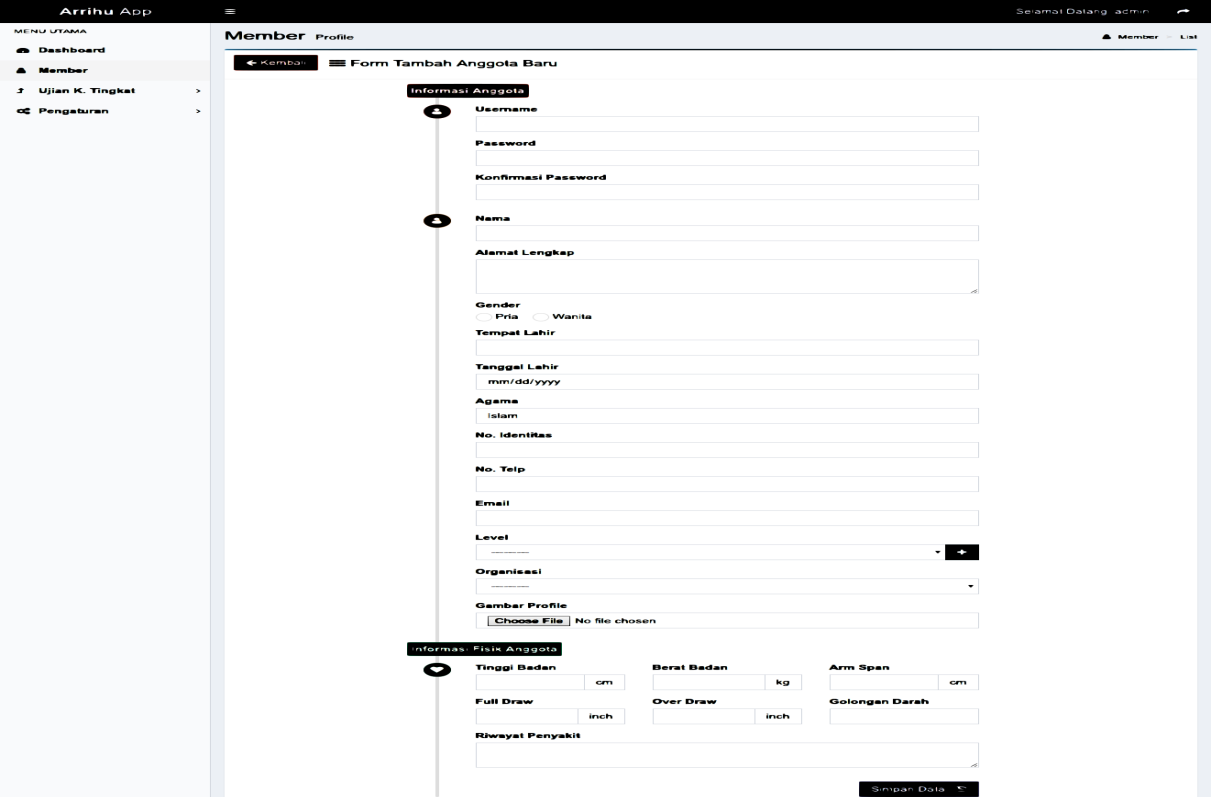
Pada halaman member pelatih akan menginputkan data peserta yang akan mengikuti UKT, setelah itu list semua anggota akan di tampilkan pada halaman member. Berikut desain halaman member:



**Gambar 3.13 Desain Halaman Member**

* + 1. Desain halaman tambah member

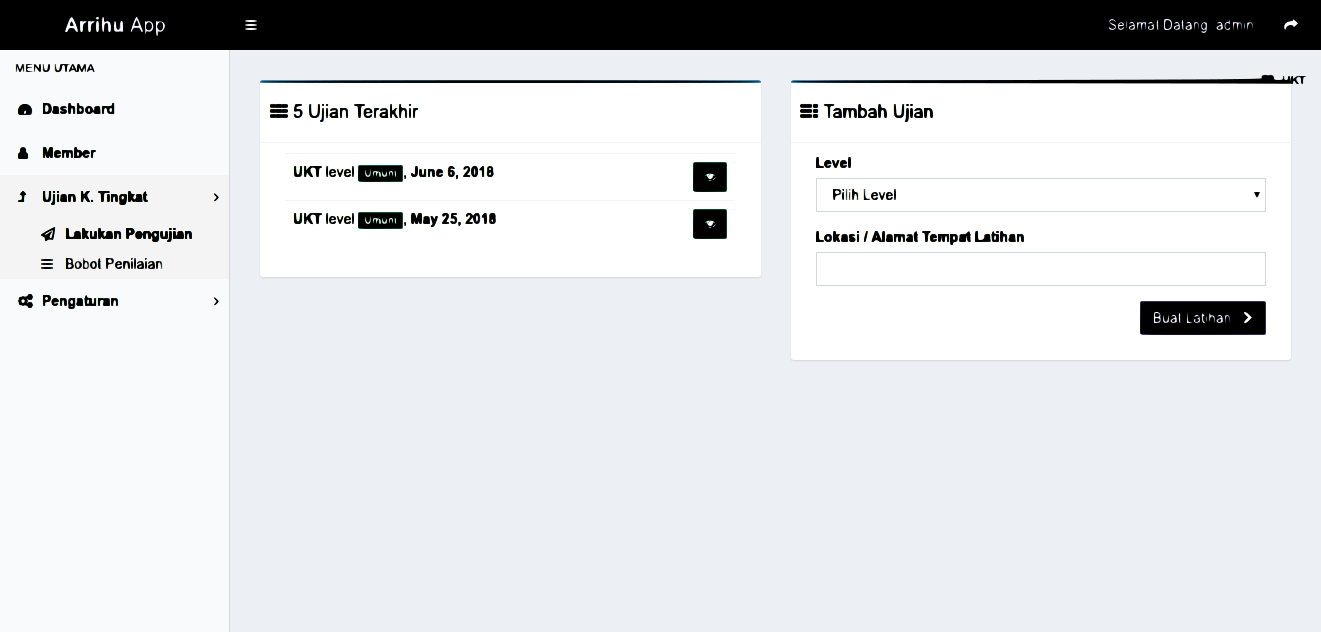
Pada halaman tambah member, data peserta UKT yang akan melakukan ujian akan di inputkan oleh pelatih ke sistem. Berikut desain halaman tambah member:



**Gambar 3.14 Desain Halaman Tambah Member**

* + 1. Desain halaman lakukan pengujian

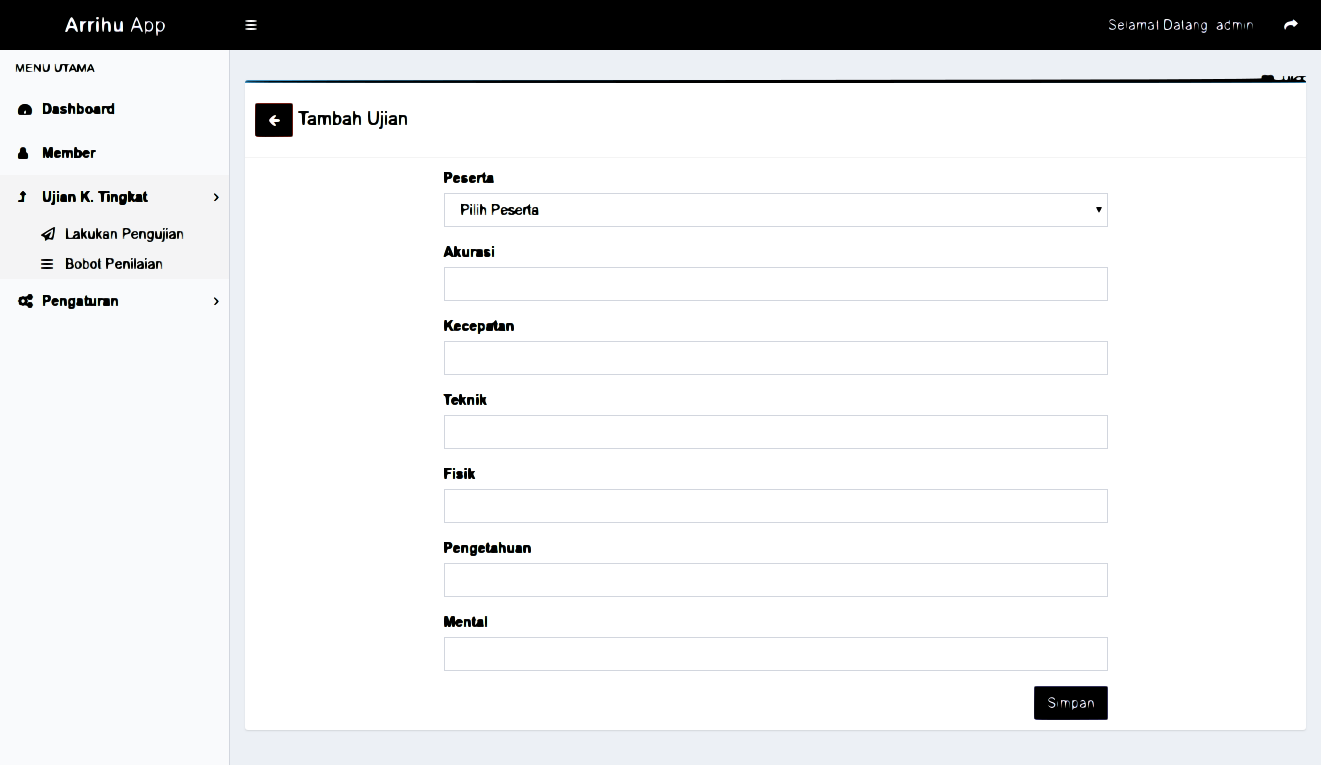
Pelatih akan menginputkan *level* dan lokasi peserta yang akan melaksanakan UKT ke sistem. Berikut desain halaman lakukan pengujian:



**Gambar 3.15 Desain Halaman Lakukan Pengujian**

* + 1. Desain halaman tambah ujian

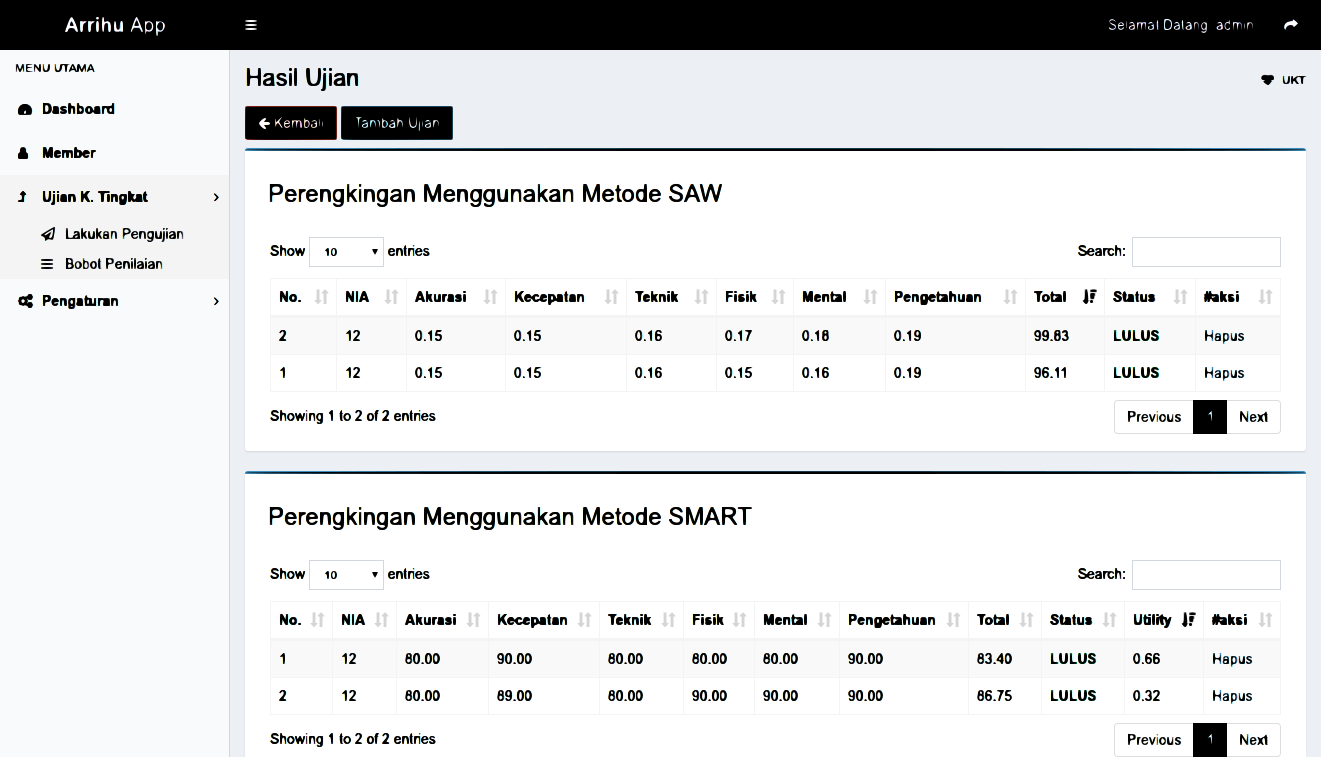
Pada halaman tambah ujian pelatih akan menginputkan *score* atlet setelah diujikan. Berikut desain halaman tambah ujian:



**Gambar 3.16 Desain Halaman Tambah Ujian**

* + 1. Desain halaman hasil ujian

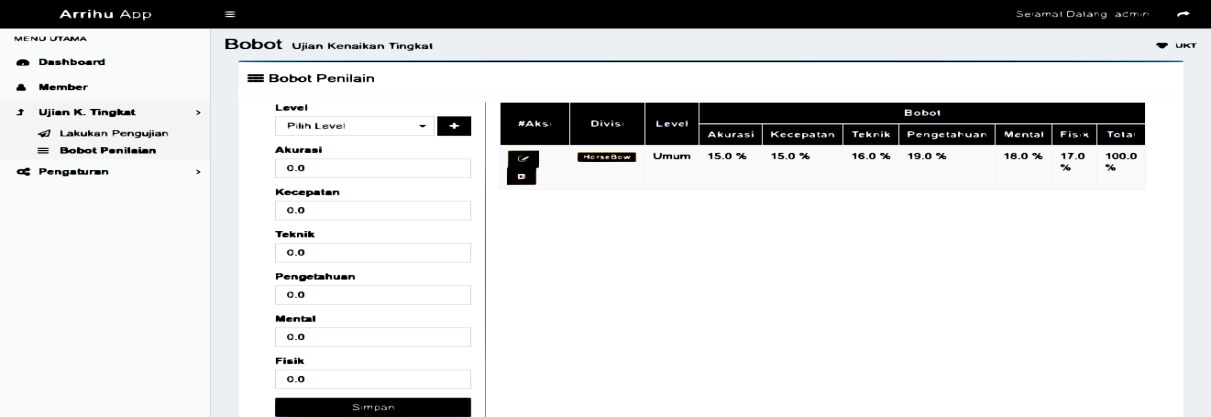
Pada halaman hasil ujian berisi informasi mengenai hasil kelulusan dan perengkingan menggunakan metode SAW dan metode SMART. Berikut desain halaman hasil ujian:



**Gambar 3.17 Desain Halaman Hasil Ujian**

* + 1. Desain halaman bobot penilaian

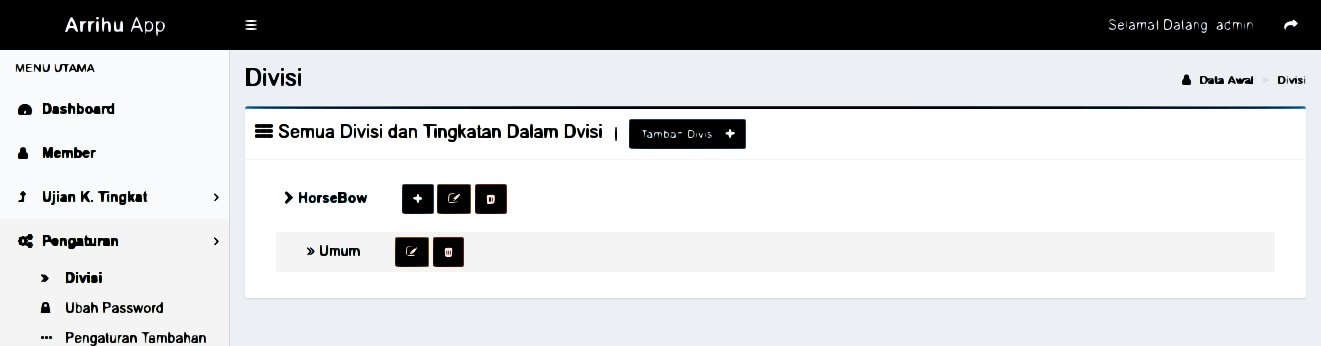
Pada halaman ini menampilkan informasi pengimputan nilai bobot dari setiap kriteria yang akan diinputkan oleh pelatih. Berikut desain halaman bobot penilaian:



**Gambar 3.18 Desain Halaman Bobot Penilaian**

* + 1. Desain halaman divisi

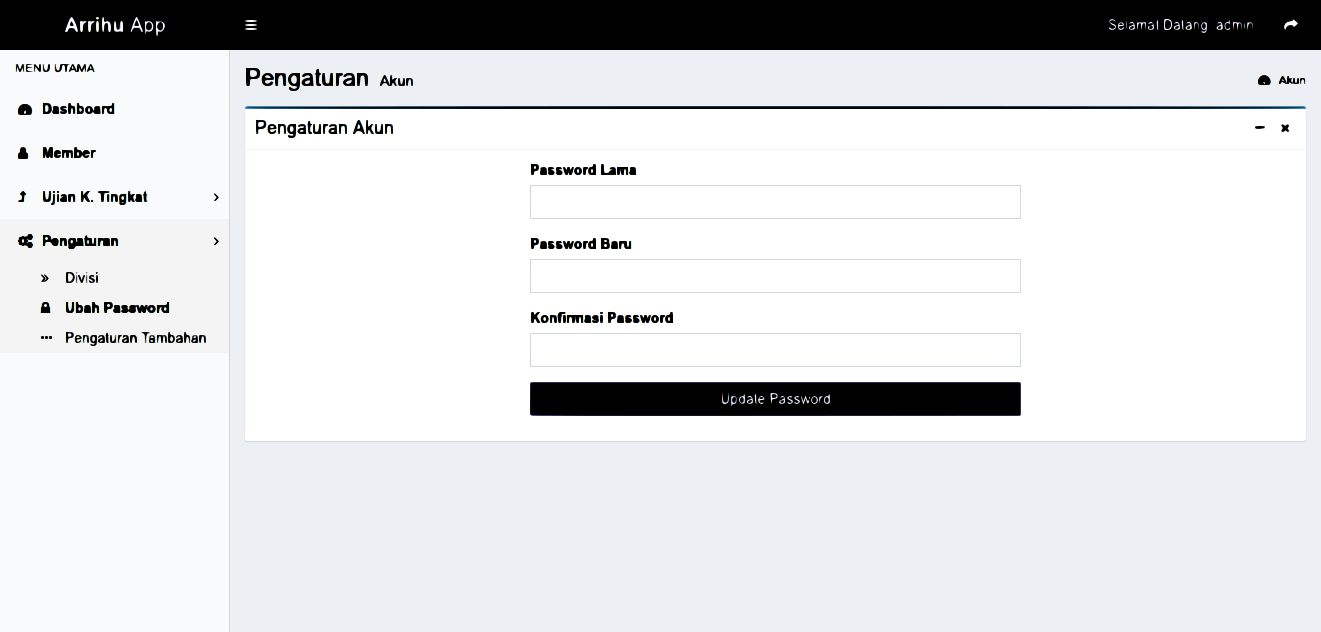
Pada halaman divisi pelatih akan menginputkan divisibesertatingkatannya. Berikut desain halaman divisi:



**Gambar 3.19 Desain Halaman Divisi**

* + 1. Desain halaman ubah *password*

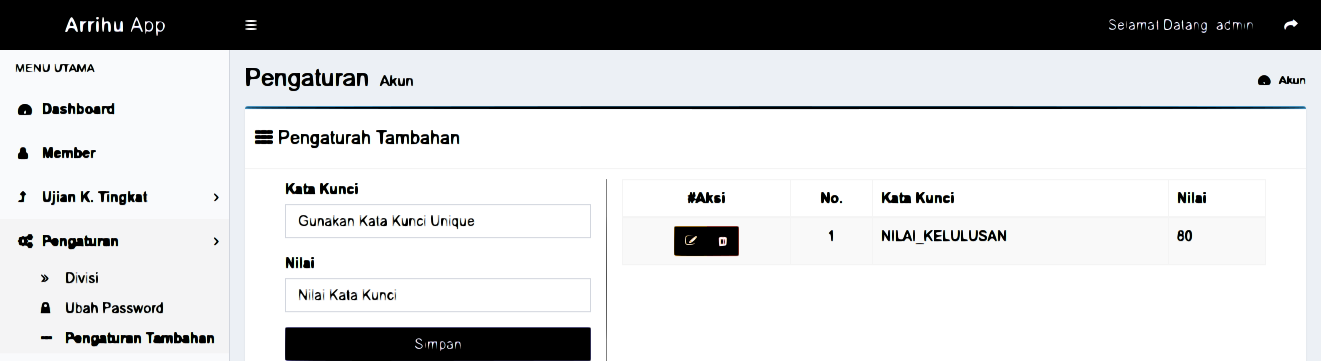
Pada halaman ini user dapat mengubah *password* lama menjadi *password* baru. Berikut desain halaman ubah *password*:



**Gambar 3.20 Desain Halaman Ubah *Password***

* + 1. Desain halaman pengaturan tambahan

Pada halaman pengaturan tambahan pelatih dapat mengubah minimum batas nilai kelulusan pada atlet yang mengikuti UKT. Berikut desain halaman pengaturan tambahan:



**Gambar 3.21 Desain Halaman Pengaturan Tambahan**

# BAB IV

# HASIL DAN PEMBAHASAN

Sebelum membahas penjelasan program, pada bab ini penulis akan menjelaskan tahap-tahap yang perlu dilakukan dalam implementasi.

**4.1 Persiapan Implementasi**

**4.1.1 Persiapan Software dan Hardware**

Pada tahapan ini dilakukan beberapa persiapan sebelum melakukan implementasi program adapun tahap-tahap yang dilakukan adalah :

1. Instalasi *MySQL Workbench*  untuk memudahkan desain database.
2. Kemudian instal *Visual Studio Code* sebagai *text editor.*
3. Instalasi paket-paket program yang dibutuhkan untuk sistem operasi Linux.
4. Instalasi Python versi 3.0 untuk meng-*compile* program dan Django versi 1.11.7 sebagai *webserver.*