**SISTEM PENDUKUNG KEPUTUSAN PEMBERIAN BONUS BERDASARKAN PENILAIAN KINERJA GURU DENGAN METODE S.A.W**

**( Study kasus SMK PUTRA SATRIA )**

**SKRIPSI**



**OLEH :**

**TITO ANDRIANTO**

**2013142047**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS PAMULANG**

**TANGERANG SELATAN**

**2017**

# 

# LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : TITO ANDRIANTO

Nim : 2013142047

Fakultas : Teknik

Program Studi : Teknik Informatika

Jenjang Pendidikan : Strata 1

Menyatakan bahwa skripsi yang saya buat dengan judul :

**SISTEM PENDUKUNG KEPUTUSAN PEMBERIAN BONUS BERDASARKAN PENILAIAN KINERJA GURU DENGAN METODE *SIMPLE ADDIVITY WIGHTING* (SAW)*.***

1. Sesungguhnya skripsi yang saya susun ini merupakan hasil karya tulis saya sendiri. Adapun bagian-bagian tertentu dalam skripsi yang saya peroleh dari hasil karya tulis orang lain, telah saya tuliskan sumbernya dengan jelas, sesuai dengan kaidah penulisan ilmiah.
2. Saya izinkan untuk dikelola oleh Universitas Pamulang sesuai dengan norma hukum dan etik yang berlaku.

Pernyataan ini saya buat dengan penuh tanggung jawab dan saya bersedia menerima kosenkuensi apapun sesuai aturan yang berlaku apabila dikemudian hari pernyataan ini tidak benar.

Tanggerang Selatan, 2 Desember 2016

(Tito Andrianto)

# LEMBAR PERSETUJUAN

NIM :2013142047

Nama :TITO ANDRIANTO

Program Studi :TEKNIK INFORMATIKA

Fakultas :TEKNIK

Jenjang Pendidikan :STRATA 1

Judul Skripsi :**SISTEM PENDUKUNG KEPUTUSAN** **PEMBERIAN BONUS PENILAIAN KINERJA GURU DENGAN METODE *SIMPLE*** ***ADDIVITY WIGHTING* (SAW)**

Skripsi ini telah diperiksa dan disetujui.

Tanggerang Selatan, 2 Desember 2016

|  |
| --- |
| Surtikanti, S.Kom., M.Kom |
| Pembimbing |

Mengetahui,

|  |
| --- |
| Hendro Waryanto, S.Si., M.M. |
| KaProdi Teknik Informatika |

# LEMBAR PENGESAHAN

NIM : 2013142047

Nama :TITO ANDRIANTO

Program Studi :TEKNIK INFORMATIKA

Fakultas :TEKNIK

Jenjang Pendidikan :STRATA 1

Judul Skripsi :**SISTEM PENDUKUNG KEPUTUSAN PEMBERIAN BONUS BERDASARKAN PENILAIAN KINERJA GURU DENGAN METODE *SIMPLE ADDIVITY WIGHTING* (SAW)**

Skripsi ini telah dipertahankan dihadapan dewan penguji ujian skripsi Fakultas Teknik, Program Studi Teknik Informatika dan dinyatakan LULUS.

Tanggerang Selatan, 2 Desember 2016

|  |
| --- |
|  |
| Penguji I |

|  |
| --- |
|  |
| Penguji II |

|  |
| --- |
| Surtikanti, S.Kom., M.Kom |
| Pembimbing |

Mengetahui,

|  |
| --- |
| Hendro Waryanto, S.Si., M.M. |
| KaProdi Teknik Informatika |

# KATA PENGANTAR



Alhamdulillah dan puji syukur kehadirat Allah SWT karena berkat rahmat dan karunia-Nya jugalah sehingga penulisan skripsi dengan judul **“SISTEM PENDUKUNG KEPUTUSAN PEMBERIAN BONUS BERDASARKAN PENILAIAN KINERJA GURU DENGAN METODE SAW (STUDI KASUS SMK PUTRA SATRIA) ”**ini dapat diselesaikan. Penulisan skripsi ini dimaksudkan sebagai salah satu syarat untuk menyelesaikan studi program strata 1 pada Fakultas Teknik Informatika Universitas Pamulang.

Dalam penyelesaian skripsi ini, penulis telah berusaha dengan segala kemampuan yang penulis miliki dan tentunya dengan bimbingan dosen-dosen pembimbing yang telah sudi meluangkan waktu untuk membimbing penulis. Untuk itu penulis mengucapkan terima kasih banyak kepada seluruh pihak yang telah membantu penulis dalam menyelesaikan skripsi ini, terutama kepada yang terhormat:

1. Bapak Dr.H. Dayat Hidayat, M.M., selaku Rektor Universitas Pamulang.
2. Bapak Ir Dadang Kurnia, M.M., selaku Dekan Fakultas Teknik Universitas Pamulang.
3. Bapak Hendro Waryanto, S.Si., M.M., selaku ketua program studi Teknik Informatika Universitas Pamulang.
4. Terima Kasih Orang Tua yang telah mendukung dan mendoakan agar kelarnya skripsi ini.
5. Ibu Surtikanti, S.Kom., M.Kom. selaku Dosen Pembimbing.
6. Teman-teman seperjuangan Teknik Informatika 2013 “Muhammad januar fauzy, Muthahharoh Zuhro, Anila Retna Anjani, Andika Pangestu, Rizky Dwiyana H,Yoga Bagus Syaputra, Muhammad Anggi Kurniawan,Imam Maulana Ibrahim,dan teman-teman yang tidak biasa di sebutkan satu-persatu”, terima kasih atas doa dan dukunganya.
7. Terimakasih Fitri Duwinyanti yang telah memberi semangat dalam menyelesaikan skripsi ini.

Atas bantuan yang telah diberikan, semoga Allah SWT memberikan balasan yang setimpal. Jujur penulis akui bahwa skripsi ini tidak luput dari kesalahan dan sangat jauh dari kesempurnaan, walaupun demikian penulis berharap semoga hasil pemikiran yang tertuang dalam skripsi ini dapat bermanfaat bagi kita semua. Amin.

Pamulang, 2 Desember 2016

**Penulis**

**ABSTRAK**

Pemberian bonus merupakan salah satu metode yang banyak digunakan sebagai sebuah bentuk penghargaan kepada pekerjaannya yang kinerjanya selama ini dapat dianggap memuaskan oleh perusahaan. Begitu juga halnya dengan SMK PUTRA SATRIA yang memberikan penghargaan kepada staff dan guru sebagai salah satu bentuk penghargaan atas kinerjanya selama ini.

Namun bonus tahunan ini hanya diberikan kepada staff dan guru yang dianggap berprestasi. Untuk itu diperlukan suatu sistem pendukung keputusan (SPK) atau *decision support system* yang dapat memperhitungkan segala kriteria yang mendukung dan guna membantu mempermudah proses pengambilan keputusan. Sistem pendukung keputusan ini menggunakan metode *Simple Additive wighting* ( SAW ).

Persoalan pendukung keputusan pada dasarnya adalah bentuk pemilihan dari berbagai alternative tindakan yang mungkin dipilih diantaranya kedesiplinan atau jumlah absen dalam setahun, masa kerja, kerajinan, dan hasil kerja dalam setahun yang prosesnya melalui mekanisme tertentu, dengan harapan akan menghasilkan sebuah keputusan yang terbaik.

**Kata kunci : Penilaian kinerja guru , *Simple Additive Wighting.***

# DAFTAR ISI

[LEMBAR PERNYATAAN i](#_Toc487402519)

[LEMBAR PERSETUJUAN ii](#_Toc487402520)

[LEMBAR PENGESAHAN iii](#_Toc487402521)

[KATA PENGANTAR iv](#_Toc487402522)

[DAFTAR ISI viii](#_Toc487402523)

[DAFTAR GAMBAR xii](#_Toc487402524)

[DAFTAR TABEL xv](#_Toc487402525)

[BAB I 1](#_Toc487402526)

[1.1 Latar Belakang 1](#_Toc487402527)

[1.2 Identifikasi Masalah 3](#_Toc487402528)

[1.3 Rumusan Masalah 3](#_Toc487402529)

[1.4 Batasan Masalah 3](#_Toc487402530)

[1.5 Tujuan Penelitian 4](#_Toc487402531)

[1.6 Manfaat Penelitian 4](#_Toc487402532)

[1.7 Sistematika Penulisan 4](#_Toc487402533)

[BAB II 6](#_Toc487402534)

[2.1 Landasan Teori 6](#_Toc487402535)

[2.1.1 Pengertian Bonus 6](#_Toc487402536)

[2.1.2 Definisi Guru 7](#_Toc487402537)

[2.1.3 Teori Sistem Pendukung Keputusan 8](#_Toc487402538)

[2.1.4 Komponen Sistem Pendukung Keputusan 10](#_Toc487402539)

[2.1.5 Pengertian Fuzzy 10](#_Toc487402540)

[2.1.5.1 Fuzzy Multiple Attribute Decision Making (FMADM) 11](#_Toc487402541)

[2.1.5.2 Metode-Metode Penyelesaian MADM 12](#_Toc487402542)

[2.1.6 Simple Additive Weighting( SAW ) 13](#_Toc487402543)

[2.1.7 Desktop 15](#_Toc487402544)

[2.1.8 Pengembangan Sistem 16](#_Toc487402545)

[2.1.8.1 UML ( Unified Modeling Language ) 16](#_Toc487402546)

[2.1.8.1.1 Tujuan Pemanfaatan UML 17](#_Toc487402547)

[2.1.8.1.2 Object Oriented Program (OOP) 18](#_Toc487402548)

[2.1.8.1.3 Komponen-komponen UML 19](#_Toc487402549)

[2.1.8.2 Basis Data 26](#_Toc487402550)

[2.1.8.2.1 Tipe Database 28](#_Toc487402551)

[2.1.9 Software Pendukung 33](#_Toc487402552)

[2.1.9.1 Java 33](#_Toc487402553)

[2.1.9.2 NetBeans 35](#_Toc487402554)

[2.1.9.3 Mysql 36](#_Toc487402555)

[2.1.9.4 Xampp 38](#_Toc487402556)

[2.1.10 Metode Pengujian Sistem 38](#_Toc487402557)

[2.1.10.1 Whitebox Testing 38](#_Toc487402558)

[2.1.10.2 Blackbox Testing 38](#_Toc487402559)

[3.1 Analisa 40](#_Toc487402560)

[3.1.1 Analisa Sistem Saat ini 40](#_Toc487402561)

[3.1.2 Evaluasi Sistem Saat Ini 41](#_Toc487402562)

[3.1.3 Analisa Sistem Yang Diusulkan 42](#_Toc487402563)

[3.1.4 Analisa Sistem Pendukung Keputusan dengan Metode SAW 43](#_Toc487402564)

[3.2 Perancangan Sistem 51](#_Toc487402565)

[3.2.1 Perancangan Basis Data 51](#_Toc487402566)

[3.2.1.1 Entity Relationship Diagram ( ERD ) 51](#_Toc487402567)

[3.2.1.2 Transformasi ERD ke LRS 52](#_Toc487402568)

[3.2.1.3 Logical Record Structure (LRS) 52](#_Toc487402569)

[3.2.1.4 Spesifikasi Basis Data 53](#_Toc487402570)

[3.2.2 Perancangan Aplikasi 55](#_Toc487402571)

[3.2.2.1 Use Case Diagram 55](#_Toc487402572)

[3.2.2.1 Activity Diagram 56](#_Toc487402573)

[3.2.2.2 Squence Diagram 62](#_Toc487402574)

[3.2.2.3 Class Diagram 68](#_Toc487402575)

[3.2.2.2 Rancangan UserInterface 68](#_Toc487402576)

[BAB IV 76](#_Toc487402577)

[4.1 Implementasi 76](#_Toc487402578)

[4.1.1 Implementasi Aplikasi 76](#_Toc487402579)

[4.1.2 Spesifikasi Perangkat Lunak Yang Digunakan 76](#_Toc487402580)

[4.1.2.1 Spesifikasi Perangkat Keras Yang Digunakan 76](#_Toc487402581)

[4.2 Implementasi Antar Muka ( *Interface* ) 77](#_Toc487402582)

[4.2.1 Implementasi Menu Utama 77](#_Toc487402583)

[4.2.2 Implementasi Menu Master Data 77](#_Toc487402584)

[4.2.3 Implementasi Menu Data Guru 78](#_Toc487402585)

[4.2.4 Implementasi Menu Bobot Tiap Kriteria 79](#_Toc487402586)

[4.2.5 Implementasi Menu user 80](#_Toc487402587)

[4.2.6 Implementasi Menu Pemberian Bonus 81](#_Toc487402588)

[4.3 Implementasi Antar Muka 82](#_Toc487402589)

[4.3.1 Implementasi Tampilan Awal 83](#_Toc487402590)

[4.3.2 Implementasi Tampilan Login 83](#_Toc487402591)

[4.3.3 Implementasi Tampilan Halaman Utama 84](#_Toc487402592)

[4.3.4 Implementasi Tampilan Data Guru 85](#_Toc487402593)

[4.3.5 Implementasi Tampilan Bobot Tiap Kriteria 85](#_Toc487402594)

[4.3.6 Implementasi Tampilan Data User 86](#_Toc487402595)

[4.3.7 Implementasi Tampilan Pemberian Bonus 86](#_Toc487402596)

[4.3.8 Implementasi Tampilan Lihat Guru Mendapatkan Bonus 87](#_Toc487402597)

[4.3.9 Implementasi Tampilan Cetak Hasil 88](#_Toc487402598)

[4.4 Pengujian Black Box Dan White Box 88](#_Toc487402599)

[4.4.1 Black Box 88](#_Toc487402600)

[4.4.2 White Box 95](#_Toc487402601)

[BAB V 107](#_Toc487402602)

[5.1 KESIMPULAN 107](#_Toc487402603)

[5.1 SARAN 107](#_Toc487402604)

[DAFTAR PUSTAKA 108](#_Toc487402605)

# DAFTAR GAMBAR

[Gambar 2. 1 Kerangka Konsep SPK **Error! Bookmark not defined.**](#_Toc468308138)

[Gambar 2. 2 Diagram Alir SPK Penilaian Kinerja Degan Metode SAW **Error! Bookmark not defined.**](#_Toc468308139)

[Gambar 2. 3 Relasi *One – to - one* **Error! Bookmark not defined.**](#_Toc468308140)

[Gambar 2. 4 Relasi *One – to* - *many* **Error! Bookmark not defined.**](#_Toc468308141)

[Gambar 2.5 Relasi *Many – to* - *many* **Error! Bookmark not defined.**](#_Toc468308142)

[Gambar 2.6 *Tanformasi one – to one* **Error! Bookmark not defined.**](#_Toc468308143)

[Gambar 2.7 Tran*form*asi*One – to - many* **Error! Bookmark not defined.**](#_Toc468308144)

[Gambar 2.8 Tranformasi *many – to - many* **Error! Bookmark not defined.**](#_Toc468308145)

[Gambar 2.9 LRS dengan*transformasi 1:1* **Error! Bookmark not defined.**](#_Toc468308146)

[Gambar 2. 10 *LRS dengan transformasi 1:M / M : 1* **Error! Bookmark not defined.**](#_Toc468308147)

[Gambar 2. 11 *LRS dengan transformasi M : N* **Error! Bookmark not defined.**](#_Toc468308148)

[Gambar 3. 1*Use Case Diagram* Saat ini **Error! Bookmark not defined.**](#_Toc468308149)

[Gambar 3. 2Sistem diusulkan untuk admin **Error! Bookmark not defined.**](#_Toc468308150)

[Gambar 3. 3 Sistem Diusulkan Untuk Kepala Sekolah **Error! Bookmark not defined.**](#_Toc468308151)

[Gambar 3. 4*Entity Relationship Diagram* ( ERD ) **Error! Bookmark not defined.**](#_Toc468308152)

[Gambar 3. 5 Transformasi ERD ke LRS **Error! Bookmark not defined.**](#_Toc468308153)

[Gambar 3. 6*Logical Record Structure* ( LRS ) **Error! Bookmark not defined.**](#_Toc468308154)

[Gambar 3. 7 *Use Case Diagram* **Error! Bookmark not defined.**](#_Toc468308155)

[Gambar 3. 8*Activity Diagram Login* **Error! Bookmark not defined.**](#_Toc468308156)

[Gambar 3. 9*Activity Diagram* Kelola Data Guru **Error! Bookmark not defined.**](#_Toc468308157)

[Gambar 3. 10*Activity Diagram* Kelola Bobot Kriteria **Error! Bookmark not defined.**](#_Toc468308158)

[Gambar 3. 11*Activity Diagram* Kelola User **Error! Bookmark not defined.**](#_Toc468308159)

[Gambar 3. 12*Activity Diagram* Kelola Pemberian Bonus **Error! Bookmark not defined.**](#_Toc468308160)

[Gambar 3. 13*Activity Diagram* Melihat Hasil Guru **Error! Bookmark not defined.**](#_Toc468308161)

[Gambar 3. 14 *Activity Diagram Logout* **Error! Bookmark not defined.**](#_Toc468308162)

[Gambar 3. 15*Squence Diagram Login* **Error! Bookmark not defined.**](#_Toc468308163)

[Gambar 3. 16*Squence Diagram* Kelola Data Guru **Error! Bookmark not defined.**](#_Toc468308164)

[Gambar 3. 17*Squence Diagram* Kelola Bobot Kriteria **Error! Bookmark not defined.**](#_Toc468308165)

[Gambar 3. 18*Squence Diagram* Kelola Data *User* **Error! Bookmark not defined.**](#_Toc468308166)

[Gambar 3. 19*Squence Diagram* Kelola Pemilihan Guru Terbaik **Error! Bookmark not defined.**](#_Toc468308167)

[Gambar 3. 20*Squence Diagram* Melihat Hasil Guru **Error! Bookmark not defined.**](#_Toc468308168)

[Gambar 3. 21*Squence Diagram Logout* **Error! Bookmark not defined.**](#_Toc468308169)

[Gambar 3. 22*Class Diagram* **Error! Bookmark not defined.**](#_Toc468308170)

[Gambar 3. 23Rancangan *User Interface* Lampiran Awal **Error! Bookmark not defined.**](#_Toc468308171)

[Gambar 3. 24Rancangan *User Interface* Halaman *Login* **Error! Bookmark not defined.**](#_Toc468308172)

[Gambar 3. 25Rancangan User Interface Halaman Utama **Error! Bookmark not defined.**](#_Toc468308173)

[Gambar 3. 26Rancangan User Interface Halaman Login **Error! Bookmark not defined.**](#_Toc468308174)

[Gambar 3. 27Rancangan User Interface Kelola Data Guru **Error! Bookmark not defined.**](#_Toc468308175)

[Gambar 3. 28Rancangan User Interface Kelola Bobot Kriteria **Error! Bookmark not defined.**](#_Toc468308176)

[Gambar 3. 29Rancangan User Interface Kelola Data User **Error! Bookmark not defined.**](#_Toc468308177)

[Gambar 3. 30Rancangan User Interface Penilaian Pemberian Bonus **Error! Bookmark not defined.**](#_Toc468308178)

[Gambar 3. 31 Rancangan User Interface Lihat Penilaian Mendapatkan Bonus **Error! Bookmark not defined.**](#_Toc468308178)

[Gambar 4. 1 Tampilan Awal **Error! Bookmark not defined.**](#_Toc468307696)

[Gambar 4. 2 Tampilan Login **Error! Bookmark not defined.**](#_Toc468307697)

[Gambar 4. 3 Halaman Utama **Error! Bookmark not defined.**](#_Toc468307698)

[Gambar 4. 4 Tampilan Data Guru **Error! Bookmark not defined.**](#_Toc468307699)

[Gambar 4. 5 Tampilan Bobot tiap Kriteria **Error! Bookmark not defined.**](#_Toc468307700)

[Gambar 4. 6 Tampilan Data User **Error! Bookmark not defined.**](#_Toc468307701)

[Gambar 4. 6 Tampilan Pemberian Bonus **Error! Bookmark not defined.**](#_Toc468307701)

[Gambar 4. 6 Tampilan Lihat Hasi Guru Mendapatkan Bonus **Error! Bookmark not defined.**](#_Toc468307701)

[Gambar 4. 6 Tampilan Cetak Hasil **Error! Bookmark not defined.**](#_Toc468307701)

# DAFTAR TABEL

[Tabel 2. 1 Keterangan Simbol – simbol *Use Case Diagram* **Error! Bookmark not defined.**](#_Toc468308697)

[Tabel 2. 2 Simbol – simbol *Activity Diagram* **Error! Bookmark not defined.**](#_Toc468308698)

[Tabel 2. 3 Simbol – simbol *Squence Diagram* **Error! Bookmark not defined.**](#_Toc468308699)

[Tabel 2. 4*Simbol – simbol Class Diagram* **Error! Bookmark not defined.**](#_Toc468308700)

[Tabel 2. 5 *Simbol – simbol Entity Relationship Diagram* ( ERD ) **Error! Bookmark not defined.**](#_Toc468308700)

[Tabel 3.1 Kriteria **Error! Bookmark not defined.**](#_Toc468435034)

[Tabel 3. 2 Variabel dan Bobot ( Nilai ) **Error! Bookmark not defined.**](#_Toc468435035)

[Tabel 3. 3 Nilai Rata – Rata Absensi **Error! Bookmark not defined.**](#_Toc468435036)

[Tabel 3.4 Nilai Rata – Rata Kedisiplinan **Error! Bookmark not defined.**](#_Toc468435037)

[Tabel 3.5 Nilai Rata – Rata Tanggung Jawab **Error! Bookmark not defined.**](#_Toc468435038)

[Tabel 3.6 Nilai Rata – Rata Kemampuan Motivasi **Error! Bookmark not defined.**](#_Toc468435039)

[Tabel 3. 7 Nilai Rata – Rata Prestasi **Error! Bookmark not defined.**](#_Toc468435040)

[Tabel 3.8 Data Guru Pemberian Bonus **Error! Bookmark not defined.**](#_Toc468435041)

[Tabel 3.9 Menunjukkan rating kecocokan dari setiap alternatif pada setiap kriteria **Error! Bookmark not defined.**](#_Toc468435042)

[Tabel 3.10 Bobot Untuk Pemberian Bonus **Error! Bookmark not defined.**](#_Toc468435043)

[Tabel 3.11 Kriteria ( Benefit ) **Error! Bookmark not defined.**](#_Toc468435044)

[Tabel 3. 12 Nilai Total integer **Error! Bookmark not defined.**](#_Toc468435045)

[Tabel 3. 14 Tabel User **Error! Bookmark not defined.**](#_Toc468435046)

[Tabel 3. 15 Tabel Guru **Error! Bookmark not defined.**](#_Toc468435046)

[Tabel 3. 16 Tabel Kriteria **Error! Bookmark not defined.**](#_Toc468435046)

[Tabel 3. 17 Tabel Bobot Kriteria **Error! Bookmark not defined.**](#_Toc468435046)

[Tabel 4. 1 Kebutuhan Perangkat Keras Komputer Yang digunakan **Error! Bookmark not defined.**](#_Toc468265330)

[Tabel 4. 2 Implementasi Menu Utama **Error! Bookmark not defined.**](#_Toc468265332)

[Tabel 4. 3 Implementasi Menu Master Data **Error! Bookmark not defined.**](#_Toc468265333)

[Tabel 4. 4 Implementasi Menu data Guru **Error! Bookmark not defined.**](#_Toc468265334)

[Tabel 4. 5Implementasi Menu Bobot Tiap Kriteria **Error! Bookmark not defined.**](#_Toc468265335)

[Tabel 4. 6 Implementasi Menu User **Error! Bookmark not defined.**](#_Toc468265336)

[Tabel 4. 7 Implementasi Menu Pemberian Bonus **Error! Bookmark not defined.**](#_Toc468265337)

[Tabel 4. 8 Implementasi Menu Lihat hasil Guru Mendapatkan Bonus **Error! Bookmark not defined.**](#_Toc468265338)

[Tabel 4. 9 Rencana Pengujian **Error! Bookmark not defined.**](#_Toc468265339)

[Tabel 4.10 Pengujian Login **Error! Bookmark not defined.**](#_Toc468265340)

[Tabel 4.11 Pengujian Data Guru **Error! Bookmark not defined.**](#_Toc468265341)

[Tabel 4.12 Pengujian Pemberian bobot tiap kriteria **Error! Bookmark not defined.**](#_Toc468265342)

[Tabel 4.13 Pengujian Data User **Error! Bookmark not defined.**](#_Toc468265343)

[Tabel 4. 14 Pengujian Pemberian Bonus **Error! Bookmark not defined.**](#_Toc468265344)

[Tabel 4. 15 Pengujian *Whitebox* Sistem pemberian Bonus **Error! Bookmark not defined.**](#_Toc468265345)

# BAB I

**PENDAHULUAN**

## Latar Belakang

Di zaman Teknologi Informasi ini (TI), berkembang pesatnya teknologi dalam berbagai hal dimana segala sesuatu telah berjalan bersama teknologi banyak berpengaruh dalam kehidupan sehari - hari. Oleh karena itu sudah seharusnya setiap perusahaan memiliki sebuah sistem informasi yang baik. Dengan adanya sistem informasi yang baik maka akan memberikan kemudahan dalam mengakses berbagai hal yang berhubungan dengan perusahaan tersebut.

SMK PUTRA SATRIA adalah suatu sekolah swasta dengan biaya yang terjangkau untuk kalangan bawah yang beralamat di Jl Ciledug Raya No 46 kelurahan Petukangan Utara kecamanatan Pesanggrahan Jakarta Selatan. Smk Putra Satria berdiri pada tahun 1988 dengan memiliki akreditas A pada tahun 2015 dan memiliki visi yang menjadikan tamatan SMK PUTRA SATRIA yang beriman, bertaqwa dan bermoral tinggi, unggul dalam mutu, terampil dalam teknologi, professional dan berjiwa wiraswasta.

Penelitian ini membahas tentang sistem pendukung keputusan pemberian bonus pada Guru untuk setiap periodenya. Sistem penentuan pemberian bonus guru pada SMK PUTRA SATRIA sampai saat ini masih dilakukan secara konvesional. Dengan mengecek satu – persatu kriteria yang dijadikan pedoman dasar dalam pengambilan keputusan siapakah pegawai yang memperoleh bonus pada periode kali ini, membuat banyaknya waktu yang tersita dalam memilih, menimbang hingga memutuskan siapakah guru yang layak diberikan bonus pada periode kali ini.

Guru merupakan suatu kesatuan dalam kegiatan belajar-mengajar yang membantu para siswa untuk mencapai pembelajaran yang baik. Penilaian kinerja guru merupakan proses analisis yang dalam rangka menghasilkan pengajaran yang baik. Kualitas pengajaran dan standar akademik perlu dievaluasi dan ditingkatkan karena pendidikan sangat penting untuk kita semua.

SMK PUTRA SATRIA secara rutin melakukan penilaian kinerja rutin terhadap seluruh guru. Proses penilaian terhadap kinerja guru di SMK PUTRA SATRIA belum begitu detail (rinci), dan masih menggunakan sistem konvesional. Dengan penilaian yang tidak rinci tersebut, dikhawatirkan akan adanya penilaian yang bersifat yang dapat menimbulkan kecemburuan sosial bagi guru yang tidak menerima prestasi. Masalah yang terjadi dalam proses evaluasi (penilaian) kinerja guru diantaranya adalah subyektifitas pengambilan keputusan akan terasa, terutama jika beberapa guru yang ada memiliki kemampuan (dan beberapa pertimbangan lain yang tidak jauh berbeda.

*Decision Support System ( DSS )* pertama kali dinyatakan oleh Micheal S. Scott Morton pada tahun 1970 dengan istilah “*Management Decision System*” (Sprague and Watson : 1993 : 4) (Turban: 1995) (McLeod: 1995). Setelah pernyataan tersebut beberapa perusahaan dan perguruan tinggi melakukan riset dan mengembangkan konsep *Decision Support System.* Pada Dasarnya *DSS* dirancang untuk mendukung seluruh tahap pengambilan keputusan melalui dari mengidentifikasi masalah, memilih data yang relevan, menentukan pendekatan yang digunakan dalam proses pengambil keputusan sampai mengavaluasi pemilihan alternative.

Kriterian untuk menentukan staff, karyawan atau guru yang memiliki prioritas untuk mendapatkan bonus gaji meliputi kedisiplinan atau jumlah absen dalam setahun, masa kerja, kerajinan, dan hasil kerja dalam setahun. Selain masih menggunakan sistem konvensional, kedekatan kepala sekolah atau pemilik yayasan dengan guru sering kali menghasilkan keputusan yang berbeda dari yang semestinya hal ini menyebabkan hasil keputusannya menjadi tidak tepat.

Untuk menunjang sistem pendukung keputusan tersebut maka penulis menggunakan suatu metode yang dapat digunakan untuk membantu sistem pendukung keputusan tersebut. Metode yang dipakai dalam mendukung keputusan pemberian bonus pada guru adalah metode *Simple Additive weighiting* karena metode ini dapat menyeleksi guru dari berbagai alternative lainya yang berhak mendapatkan bonus dari instansi atau sekolahan tersebut.

Metode *Simple Additive Weightting* ini dipilih karena dapat menentukan nilai bobot untuk setiap atribut yang akan menyeleksi alternative terbaik dari sejumlah altermate lainnya, dalam hal ini alternative yang dimaksud adalah yang berhak menerima bonus berdasarkan kriteria – kriteria yang telah ditentukan. Dengan metode tersebut diharapkan penilaian akan lebih akurat. Karena, didasarkan pada nilai kriteria yang sudah ditentukan sehingga akan mendapatkan hasil yang lebih akurat terhadap siapa yang akan menerima bonus tersebut(Zulkilfi, Sariffudin;, 2016).

## Identifikasi Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka penulis melakukan identifikasi masalah seperti berikut ini:

1. Belum adanya sistem pendukung keputusan dalam membantu melakukan pemberian bonus menurut kinerja guru.
2. Pengambilan keputusan secara manual akan menghasilkan penilaian yang tidak objektif sehingga tidak tepat.

## Rumusan Masalah

Adapun beberapa permasalahan yang akan dibahas dalam skripsi ini adalah sebagai berikut :

1. Bagaimana menentukan kriteria sistem pendukung keputusan guru yang berhak menerima bonus menggunakan bahasa pemrograman ?
2. Menerapkan penjumlahan terbobot metode *Simple Additive Weightting* pada sistem pendukung keputusan penerimaan bonus pada guru ?

## Batasan Masalah

Agar pembahasan tidak menyimpang dari pokok permaslahan yang telah di rumuskan maka dapat diambil batasan masalah sebagai berikut :

1. Sistem pendukung keputusan dibuat dengan ruang lingkup pemberian bonus berdasarkan kriteria pada SMK Putra Satria.
2. Sistem pendukung keputusan pemberian bonus ini dirancang untuk memberi masukkan kepada ketua yayasan, siapa guru atau staff yang berhak untuk mendapatkan bonus tersebut.

## Tujuan Penelitian

1. Merancang sistem pendukung keputusan guru atau staff yang berhak menerima bonus menggunakan bahasa pemrograman.
2. Untuk menerapkan metode *Simple Additive Wightting* pada sistem penggambil keputusan ( SPK ).

## Manfaat Penelitian

Adapun manfaat dalam penelitian ini adalah sebagai berikut :

1. Bagi Peneliti

Dapat menambah wawasan tentang ilmu Sistem Pendukung Keputusan menggunakan metode *Simple Additive Weightting* ( SAW ).

1. Bagi Instansi

Dapat membantu pihak sekolah dalam mengambil keputusan pemilihan guru terbaik di SMK PUTRA SATRIA Jakarta Selatan.

1. Bagi Akademik

Dapat menambah sumber referensi bagi peneliti yang akan melakukan penelitian tentang sistem pendukung keputusan pemberian bonus menurut kinerja guru.

## Sistematika Penulisan

Uraian singkat mengenai struktur penulisan pada masing-masing bab adalah sebagai berikut:

|  |  |
| --- | --- |
| **BAB I**  **BAB II**  **BAB III**  **BAB IV**  **BAB V**  **BAB VI** | **PENDAHULUAN**  Membahas Latar Belakang Masalah, Identifikasi Masalah, Batasan Masalah, Tujuan Penelitian, Metodelogi Penelitian serta Sistematika Penulisan.  **LANDASAN TEORI**  Memaparkan teori-teori yang didapat dari sumber-sumber yang relevan untuk digunakan sebagai panduan dalam penelitian serta penyusunan Skripsi.  **ANALISIS SISTEM**  Menjelaskan tentang gambaran sistem serta deskripsi dari hasil analisis sistem yang akan dijadikan sebagai petunjuk untuk perancangan pada tahapan berikutnya.  **PERANCANGAN SISTEM**  Berisi tentang perancangan sistem serta komponen-komponen pemodelan sistem yang digunakan, yaituPerancanganSistemdengan *UML*, Perancangan Data,Perancangan Arsitektural, Perancangan Prosedural dan Perancangan Antarmuka.  **IMPLEMENTASI SISTEM**  Bab ini menjelaskan mengenai kebutuhan *hardware*, *software* serta *tools* lainnya. Dalam bab ini juga dibahas mengenai arsitektur dan proses pengembangan aplikasi berbasis dekstop, yaitu berupa pembuatan user interface, pembuatan controller untuk mengatur hubungan user interface dengan basis data dan pembuatan model untuk proses basis data.  **PENUTUP**  Mengemukakan kesimpulan yang diambil dari hasil penelitian dan penulisan skripsi ini, serta saran-saran untuk pengembangan selanjutnya, agar dapat dilakukan perbaikan-perbaikan di masa yang akan dating. |

# BAB II

**TINJAUAN PUSTAKA**

## Landasan Teori

### Pengertian Bonus

Bonus, bonus tahunan atau gaji ke-13 merupakan tambahan uang yang diberikan kepada pekerja selain gaji. Biasanya bonus tahunan diberikan sesuai dengan prestasi dan kemampuan/keuntungan perusahaan di akhir tahun. Bonus tahunan penting bagi perusahaan.

Bonus adalah sejumlah uang yang ditambahkan ke [gaji karyawan](http://www.gajimu.com/main/gaji/cek-gaji), biasanya diperuntukkan bagi karyawan sebagai hadiah untuk karena mereka telah melakukan pekerjaan dengan baik. Apabila pembayaran gaji pokok biasanya dilakukan setiap bulan, maka pembayaran bonus dilakukan secara bervariasi, tergantung pada kriteria-kriteria seperti : omset tahunan perusahaan, atau jumlah pelanggan yang diperoleh, atau nilai saham perusahaan saat ini. Dengan demikian pembayaran bonus dapat bertindak sebagai insentif bagi para pekerja agar termotivasi untuk mencari keuntungan bagi keberhasilan ekonomi perusahaan mereka.

Bonus dapat pula dibagi menjadi beberapa jenis yaitu :

1. Bonus Retensi

Bonus Retensi adalah pembayaran insentif yang digunakan untuk mencegah [karyawan meninggalkan perusahaan](http://www.gajimu.com/main/pekerjaan-yanglayak/kontrak-kerja/pemutusan-hubungan-kerja/pemutusan-hubungan-kerja-phk). Biasanya karyawan diminta untuk menandatangani perjanjian yang menyatakan mereka akan tetap bekerja untuk jangka waktu tertentu atau sampai selesainya suatu tugas atau proyek tertentu agar memenuhi syarat untuk mendapatkan bonus.

1. Bonus Tahunan

Bonus Tahunan adalah sebuah pembayaran kompensasi variabel, biasanya dalam bentuk uang tunai, yang diberikan kepada karyawan jika kinerja tahunan perusahaan melebihi target keuangan dan non-keuangan yang ditentukan. Ukuran bonus umumnya dinyatakan sebagai persentase dari [gaji pokok](http://www.gajimu.com/main/pekerjaan-yanglayak/upah-kerja/pertanyaan-mengenai-gaji-atau-upah-kerja-1/pertanyaan-mengenai-gaji-atau-upah-kerja) dan mungkin memiliki minimum yang dijamin dan maksimum tertentu

1. Bonus Akhir Tahun

Bonus Akhir Tahun adalah adalah pembayaran yang terkadang diberikan kepada karyawan pada akhir tahun ketika karyawan dan/atau perusahaan berkinerja sangat baik.

1. Tanteim

Tantiem adalah bagian keuntungan perusahaan yang dihadiahkan kepada karyawan, yang baru dapat diberikan bila perusahaan memperoleh laba bersih sebagaimana ditentukan dalam Pasal 70 ayat (1) UU No. 40 Tahun 2007 tentang Perseroan Terbatas (UU PT). Dalam Surat Edaran Direktur Jenderal Pajak No. SE-16/PJ.44/1992 Tentang Pembagian Bonus, Gratifikasi, Jasa Produksi Dan Tantiem disebutkan bahwa, Tantiem merupakan bagian keuntungan yang diberikan kepada[Direksi dan Komisaris oleh pemegang saham](http://www.gajimu.com/main/gaji/gaji-pejabat-negara-ri/gaji-direksi-bumn/daftar-gaji-jajaran-direksi-badan-usaha-milik-negara-bumn) yang didasarkan pada suatu prosentase/jumlah tertentu dari [laba perusahaan setelah kena pajak](http://www.gajimu.com/main/gaji/pajak-penghasilan/tarif-pajak-penghasilan-pribadi/tarif-pajak-penghasilan-pribadi).

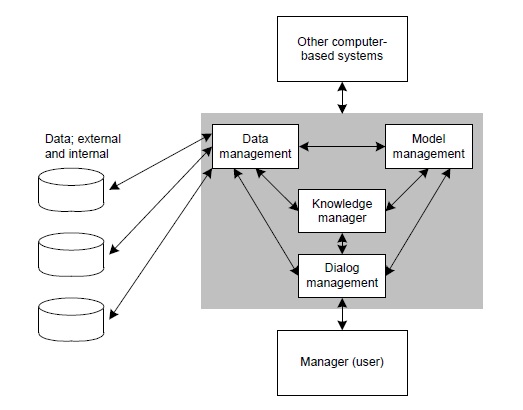
### Definisi Guru

Guru adalah semua orang yang berwenang dan bertanggung jawab terhadap pendidikan murid-murid, baik secara individual maupun klasikal, baik di sekolah maupun luar sekolah”. Ini berarti bahwa seorang **guru**, minimal harus memiliki dasar-dasar kompetensi sebagai wewenang dan kemampuan dalam menjalankan tugas. Berdasarkan uraian di atas, dapatlah dipahami bahwa kompetensi **guru** merupakan suatu kemampuan yang mutlak dimiliki oleh seorang **guru**, baik dari segi pengetahuan, keterampilan dan kemampuan serta tanggung jawab terhadap murid-murid yang di asuhnya,sehingga tugasnya sebagai seorang pendidik dapat terlaksana dengan baik.  
Definisi Guru menurut para ahli:

* **Definisi Guru menurut Noor Jamaluddin (1978: 1)** - **Guru** adalah pendidik, yaitu orang dewasa yang bertanggung jawab memberi bimbingan atau bantuan kepada anak didik dalam [perkembangan](http://definisimu.blogspot.com/2012/08/definisi-perkembangan.html) jasmani dan rohaninya agar mencapai kedewasaannya, mampu berdiri sendiri dapat melaksanakan tugasnya sebagai makhluk Allah khalifah di muka bumi, sebagai makhluk sosial dan individu yang sanggup berdiri sendiri.
* [**Definisi Guru**](http://definisimu.blogspot.com/2012/09/definisi-guru.html)**menurut Peraturan Pemerintah** - **Guru** adalah jabatan fungsional, yaitu kedudukan yang menunjukkan tugas, tanggung jawab, wewenang, dan hak seorang PNS dalam suatu organisasi yang dalam pelaksanaan tugasnya didasarkan keahlian atau keterampilan tertentu serta bersifat mandiri.
* **Definisi Guru menurut Keputusan Men.Pan** - **Guru** adalah Pegawai Negeri Sipil yang diberi tugas, wewenang dan tanggung jawab oleh pejabat yang berwenang untuk melaksanakan pendidikan di sekolah.
* [**Definisi**](http://definisimu.blogspot.com/)**Guru menurut Undang-undang No. 14 tahun 2005** - **Guru** adalah pendidik profesional dengan tugas utama mendidik, mengajar, membimbing, mengarahkan, melatih, menilai, dan mengevaluasi peserta didik pada pendidikan anak usia dini jalur pendidikan formal, pendidikan dasar, dan [pendidikan](http://definisimu.blogspot.com/2012/07/definisi-pendidikan.html) menengah.

### Teori Sistem Pendukung Keputusan

Pada awal tahun 1970an, Scott-Morton merumuskan konsep SPK yang pertama. Mendefinisikan bahwa SPK adalah sistem interaktif berbasis komputer yang membantu mengambil keputusan memanfaatkan data dan model untuk memecahkan masalah yang tidak terukur. (Gorry and Scott-Morton, 1971; Effraim, 2007).

 Kerangka konsep sistem pendukung keputusan yang terdiri dari data; eksternal dan internal, computer di luar sistem, manajemen model, pengetahuan manajer, dialog manajemen dan manajer ( *user* ) seperti gambar di bawah .( Effraim, 2007)

**Gambar 2.1** Kerangka Konsep SPK

Langkah yang dilakukan dalam SPK penilaian kinerja dengan menggunakan metode SAW bisa dilihat pada gambar.

INPUT KRITERIA DAN NILAI

PERSYARATAN PENILAIAN KINERJA

INPUT DATA KARYAWAN

ANALISIS DENGAN SAW

TAMPILKAN HASIL

**Gambar 2.2 Diagram Alir SPK penilaian kinerja dengan metode SAW**

Input kriteria dan nilai dilakukan untuk memberi nilai perbandingan sehingga menghasilkan nilai prioritas local ataupun prioritas global. Persyaratan penilaian kinerja dilakukan untuk memberikan syarat- syarat yang digunakan sebelum proses penilaian kinerja dilakukan input data karyawan lalu dihitung dengan menggunakan metode SAW dan ditampilkan hasil karyawan yang lulus dan ditunda.(Taufik, Rohmat, 2012)

### Komponen Sistem Pendukung Keputusan

Sistem pendukung keputusan harus mencangkup tiga komponen

utama yaitu (Kusrini, 2007):

1. Subsistem data *(Database)*

Subsistem data merupakan komponen sistem pendukung keputusan sebagai penyedia data bagi sistem. Data disimpan data suatu rangkaian data yang diorganisasikan oleh suatu sistem yaitu sistem manajemen *(databasemanagement system).* Pangkalan data dalam sistem pendukung keputusan berasal dari dua sumber, yaitu sumber internal (dari dalam organisasi atau perusahaan) dan sumber eksternal (dari luar organisasi atau perusahaan).

1. Subsistem model

Model adalah suatu peniruan dari alam nyata. Penholahan berbagai model dilakukan dalam pengakalan model dilakukan secara fleksibel untuk membantu pengguna dalam memodifikasi dan menyempurnakan model.

1. Subsistem dialog

Subsistem dialog adalah fasilitas yang mampu mengintegrasikan sistem yang terpasang dengan pengguna secara interaktif. Melalui subsistem inilah sistem diartikulasi dan diimplementasi sehingga pengguna dapat berkomunikasi dengan sistem yang dirancang.

### Pengertian Fuzzy

Teori tentang *fuzzy* atau himpunan samar pertama kali dikembangkan oleh Lotfi Zadeh sekitar tahun 1965 pada sebuah makalah yang berjudul ’*Fuzzy Sets*’, sejak pertengahan 1970 para peneliti jepang berhasil mengaplikasikan teori ini kedalam berbagai permasalahan praktis (Kusumadewi S, 2003). *Fuzzy* mempresentasikan dan menangani masalah ketidakpastian yang dalam hal ini bias berarti keraguan, ketidakpastian, kekurangan kelengkapan in*form*asi dan kebenaran yang bersifat sebagian. Kata *Fuzzy* merupakan kata sifat yang berarti kabur, tidak jelas. *Fuzziness* atau kekaburan atau ketidakjelasan atau ketidakpastian selalu meliputi keseharian manusia.

Himpunan *crips*didefinisikan himpunan yang membagi sekelompok individu kedalam dua kategori yaitu anggota dan bukan anggota (Kusumadewi S, 2006).Himpunan *fuzzy*didasarkan pada gagasan untuk memperluas jangkauan fungsi karakteristik sedemikian hingga tersebut mencangkup bilangan real pada interval [0,1]. Nilai keanggotaan menunjukan bahwa suatu item dalam semesta pembicaraan tidak hanya berada pada 0 dan 1, namun juga nilai terletak diantaranya. Dengan kata lain, nilai keberadaan suatu sistem tidak hanya bernilai benar atau salah. Nilai 0 menunjukan salah nilai 1 menunjukan benar dan masih ada nilai-nilai yang terletak antara benar dan salah.

Penerapan *fuzzy logic* sering digunakan, ada beberapa alasan mengapa orang menggunakan logika *fuzzy* antara lain :

1. Konsep logika *fuzzy* mudah dimengerti. Konsep matematis yang mendasari penalaran *fuzzy* sederhana dan mudah dimengerti.
2. Logika *fuzzy* sangat fleksibel.
3. Logika *fuzzy* memiliki toleransi terhadap data-data yang tidak tepat.
4. Logika *fuzzy* mampu memodelkan fungsi-fungsi nonlinier yang sangat kompleks.

#### Fuzzy Multiple Attribute Decision Making (FMADM)

*Fuzzy Multiple Decision Making* (FMADM) adalah suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternative dengan kriteria tertentu(Sri Kusumadewi, 2006). Inti dari FMADM adalah menentukan nilai bobot dari setiap atribut, kemudian dilanjutkan dengan proses perangkingan yang menyelesaikan alternative yang sudah diberikan.

Pada dasarnya terdapat tiga untuk mencari nilai bobot atribut, pendekatan subjektif dan pendekatan objektif. Masing-masing pendekatan memiliki kelebihan dan kelemahan. Pada pendekatan subjektif, nilai bobot disarankan subjektifitas dari para pengambil keputusan, sehingga factor perangkingan alternatif dapat ditentukan secara bebas. Sedangkan pada pendekatan objektif, nilai bobot dihitung secara objektif, nilai bobot dihitung secara matematis sehingga mengabaikan subjektifitas dari pengambilan keputusan. Ada beberapa metode untuk menyelesaikan masalah FMADM antara lain (Lestari, 2013) :

1. *Simple Additive Weighting* (SAW)
2. *Weighted Product* (WP)
3. *ELECTRE (Elemination Et Choix TRaduisant La realitE)*
4. TOPSIS *(Technique for Order Preference by Similarity to Ideal Solution)*
5. AHP (*Analiytic Hierarchy Process)*

#### Metode-Metode Penyelesaian MADM

Beberapa metode yang dapat digunkan untuk menyelesaikan masalah dalam MADM (*Multi Attribute Decision Making)* (Kusumadewi S, 2006) sebagai berikut :

1. *Simple Additive Weighting* (SAW)

Metode 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.

1. *Weighted Product* (WP)

Metode WP menggunakan perkalian untuk menghubungkan rating atribut, dimana rating setiap atribut harus dipangkatkan dulu dengan bobot atribut yang bersangkutan.

1. *ELECTRE (Elemination Et Choix TRaduisant La realitE)*

Metode ELECTRE didasarka pada konsep perangkingan melalui perbandingan berpasangan antar alternatif pada criteria yang sesuai.

1. TOPSIS *(Technique for Order Preference by Similarity to Ideal Solution)*

Metode TOPSIS didasarkan pada konsep dimana alternatif terpilih yang terbaik tidak hanya memiliki jarak terpendek dari solusi ideal positif, namun juga memiliki jarak terpanjang dari solusi ideal negative.

1. AHP (*Analiytic Hierarchy Process)*

Metode AHP adalah metode yang digunakan untuk memecahkan suatu situasi yang kompleks dan tidak terstruktur kedalam beberapa komponen dalam susunan yang hierarki, dengan memberi nilai subjektif tentang pentingnya setiap variabel secara relatif, dan menetapkan variabel mana yang memiliki prioritas paling tinggi guna mempengaruhi hasil pada situasi tersebut.

### Simple Additive Weighting( SAW )

Metode SAW sering dikenal dengan istilah metode penjumlahan terbobot. Konsep dasar metode SAW (*Simple Additive Weighting*) adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut. Metode SAW dapat membantu dalam pengambilan keputusan suatu kasus, akan tetapi perhitungan dengan menggunakan metode SAW ini hanya yang menghasilkan nilai terbesar yang akan terpilih sebagai alternatif yang terbaik. Perhitungan akan sesuai dengan metode ini apabila alternatif yang terpilih memenuhi kriteria yang telah ditentukan. Metode SAW ini lebih efisien karena waktu yang dibutuhkan dalam perhitungan lebih singkat. Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada.

Metode SAW sering juga dikenal dengan istilah metode penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif ada semua atribut (Sri, 2006). Metode SAW membutuhkan proses normalisasi matrik keputusan (X) ke suatu skala dapat diperbandingkan dengan semua sating alternative yang ada (Kusuma, 2006). *Metode Simple Additive Weighting* ini mempunyai beberapa kelebihan dan kekurangan, diantaranya yaitu:

1. Kelebihan:
2. Menentukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan proses perangkingan yang akan menyeleksi alternative terbaik dari sejumlah alternative
3. Penilain akan lebih tepat karena didasarkan pada nilai kriteria dan bobot preferensi yang sudah ditentukan.
4. Kekurangan:
5. Perhitungan dilakukan dengan menggunakan bilangan crisp.
6. Adanya perbedaan perhitungan normalisasi matriks sesuai dengan nilai atribut (antara nilai benefit dan cost).

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) (MacCrimmon, 1968).

Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada. Metode ini merupakan metode yang paling terkenal dan paling banyak digunakan dalam menghadapi situasi Multiple Attribute Decision Making (MADM). MADM itu sendiri merupakan suatu metode yang digunakan untuk mencari alternatifoptimal dari sejumlah alternatif dengan kriteria tertentu. Metode SAW ini mengharuskan pembuat keputusan menentukan bobot bagi setiap atribut. Skor total untuk alternatif diperoleh dengan menjumlahkan seluruh hasil perkalian antara rating (yang dapat dibandingkan lintas atribut) dan bobot tiap atribut. Rating tiap atribut haruslah bebas dimensi dalam arti telah melewati proses normalisasi matriks sebelumnya.

Langkah Penyelesaian Simple Additive Weighting (SAW)

Langkah Penyelesaian SAW sebagai berikut :

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu Ci.
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
3. Membuat matriks keputusan berdasarkan kriteria(Ci), kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi $R.
4. Hasil akhir diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vektor bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (Ai)sebagai solusi.

* Formula untuk melakukan normalisasi tersebut adalah sebagai berikut:



* rij adalah penilaian kinerja ternormalisasi dari alternatif Ai pada atribut Cj; i=1,2,...,m dan j=1,2,...,n.
* Nilai preferensi untuk setiap alternatif (Vi) diberikan sebagai:



Nilai Vi yang lebih besar mengindikasikan bahwa alternatif Ai lebih terpilih.(Fauziah, Wakhidatul;, 2015)

### Desktop

*Desktop base* merupakan suatu aplikasi yang dapat berjalan sendiri atau independen tanpa menggunakan *browser* atau koneksi *internet* di suatu komputer dengan *operation system* tertentu. Biasanya aplikasi *desktop* ini harus di *install* terlebih dahulu serta filenya berekstensi *.exe* (untuk Windows) atau *.sh* (untuk Linux).

1. Keunggulan aplikasi *desktop base* adalah seperti berikut:
2. Lebih cepat, karena *desktop base* tidak membutuhkan waktu tambahan dalam meng-*compile* barisan kode pemrograman.
3. Proses pengamanan pada *desktop* lebih mudah daripada *web*, karena *web* hubungannya langsung ke dunia luar, dan resiko keamanannya pun menyangkut proses *online*.
4. Tidak mudah diserang *hacker* (peretas) karena aplikasi tersebut di *install* di setiap komputer.
5. *Desktop* lebih memudahkan pengguna dari segi kegunaan, dimana dengan jumlah komponen yang lebih banyak dan bervariasi dibanding *web*.
6. Kelemahan aplikasi dekstop base adalah seperti berikut:
7. Hanya dapat diakses lewat komputer yang telah ter-*install* aplikasi.
8. Butuh spesifikasi *hardware* komputer yang selaras dengan kebutuhan sistem. Semakin rumit aplikasi maka semakin membutuhkan spesifikasi komputer yang tinggi.
9. Tidak dapat di-*install* diberbagai macam Sistem Operasi. Biasanya aplikasi harus dibuat berbeda untuk setiap sistem operasi.

### Pengembangan Sistem

#### UML ( Unified Modeling Language )

UML ( *Unified Modeling Language* ) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual (Braun, et. al. 2001). Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek (Whitten, et. al. 2004).

Sejarah UML sendiri terbagi dalam dua fase; sebelum dan sesudah munculnya UML. Dalam fase sebelum, UML sebenarnya sudah mulai diperkenalkan sejak tahun 1990an namun notasi yang dikembangkan oleh para ahli analisis dan desain berbeda-beda, sehingga dapat dikatakan belum memiliki standarisasi.

Fase kedua; dilandasi dengan pemikiran untuk mempersatukan metode tersebut dan dimotori oleh Object Management Group (OMG) maka pengembangan UML dimulai pada akhir tahun 1994 ketika Grady Booch dengan metode OOD (*Object-Oriented Design*), Jim Rumbaugh dengan metode OMT (*Object Modelling Technique*) mereka ini bekerja pada Rasional Software Corporation dan Ivar Jacobson dengan metode OOSE (*Object-Oriented Software Engineering*) yang bekerja pada perusahaan Objectory Rasional.

Sebagai pencetus metode-metode tersebut mereka bertiga berinisiatif untuk menciptakanbahasa pemodelan terpadu sehingga pada tahun 1996 mereka berhasil merilis UML versi 0.9 dan 0.91 melalui *Request for Proposal* (RFP) yang dikeluarkan oleh OMG (Braun, et.al. 2001).

Kemudian pada Januari 1997 IBM, ObjecTime, Platinum Technology, Ptech, Taskon, Reich Technologies dan Softeam juga menanggapi *Request for Proposal* (RFP) yang dikeluarkan oleh OMG tersebut dan menyatakan kesediaan untuk bergabung.

Perusahaan-perusahaan ini menyumbangkan ide-ide mereka, dan bersama para mitra menghasilkan UML revisi 1.1. Fokus dari UML versi rilis 1.1 ini adalah untuk meningkatkan kejelasan UML Semantik versi rilis 1.0. Hingga

saat ini UML versi terbaru adalah versi 2.0.

Saat ini sebagian besar para perancang sistem informasi dalam menggambarkan informasi dengan memanfaatkan UML diagram dengan tujuan utama untuk membantu tim proyek berkomunikasi, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau pembuat program.

Secara filosofi UML diilhami oleh konsep yang telah ada yaitu konsep permodelan Object Oriented karena konsep ini menganalogikan sistem seperti kehidupan nyata yang didominasi olehobyek dan digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik.

##### Tujuan Pemanfaatan UML

Tujuan dari penggunaan diagram seperti diungkapan oleh Schmuller J. (2004), “*The purpose of the diagrams is to present multiple views of a system; this set of multiple views is called a model*”.

Berikut tujuan utama dalam desain UML adalah (Sugrue J. 2009) :

1. Menyediakan bagi pengguna (analisis dan desain sistem) suatu bahasa pemodelan visual yang ekspresif sehingga mereka dapat mengembangkan dan melakukan pertukaran model data yang bermakna.
2. Menyediakan mekanisme yang spesialisasi untuk memperluas konsep inti.
3. Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu.
4. Memberikan dasar formal untuk pemahaman bahasa pemodelan.
5. Mendorong pertumbuhan pasar terhadap penggunaan alat desain sistem yang berorientasi objek (OO).
6. Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola dan komponen terhadap suatu sistem.
7. Memiliki integrasi praktik terbaik.

##### Object Oriented Program (OOP)

*Object Oriented Program* (OOP)merupakan paradigma baru dalam rekayasa software yang didasarkan pada obyek dan kelas. (Ronald J.N., 1996). Diakui para ahli bahwa *objectoriented* merupakan metodologi terbaik yang ada saat ini dalam rekayasa software. Object-oriented memandang software bagian per bagian dan menggambarkan satu bagian tersebut dalam satu obyek.

Teknologi obyek menganalogikan sistem aplikasi seperti kehidupan nyata yang didominasi oleh obyek. Dengan demikian keunggulan teknologi obyek adalah bahwa model yang dibuat akan sangat mendekati dunia nyata yang masalahnya akan dipecahkan oleh sistem yang dibangun. Model obyek, atribut dan perlakuannya bisa langsung diambil dari obyek yang ada di dunia nyata.

Ada 4 (empat) prinsip dasar dari pemrograman berorientasi obyek yang menjadi dasar kemunculan UML, yaitu abstraksi, enkapsulasi, modularitas dan hirarki. Berikut dijelaskan satu persatu secara singkat.

1. Abstraksi memfokuskan perhatian pada karakteristik obyek yang paling penting dan paling dominan yang bisa digunakan untuk membedakan obyek tersebut dari obyek lainnya.
2. Enkapsulasi menyembunyikan banyak hal yang terdapat dalam obyek yang tidak perlu diketahui oleh obyek lain. Dalam praktek pemrograman, enkapsulasi diwujudkan dengan membuat suatu kelas interface yang akan dipanggil oleh obyek lain, sementara didalam obyek yang dipanggil terdapat kelas lain yang mengimplementasikan apa yang terdapat dalam kelas interface.
3. Modularitas membagi sistem yang rumit menjadi bagian-bagian yang lebih kecil yang bisa mempermudah developer memahami dan mengelola obyek tersebut.
4. Hirarki berhubungan dengan abstraksi dan modularitas, yaitu pembagian berdasarkan urutan dan pengelompokkan tertentu. Misalnya untuk menentukan obyek mana yang berada pada kelompok yang sama, obyek mana yang merupakan komponen dari obyek yang memiliki hirarki lebih tinggi. Semakin rendah hirarki obyek berarti semakin jauh abstraksi dilakukan terhadap suatu obyek.

##### Komponen-komponen UML

Sejauh ini para pakar merasa lebih mudah dalam menganalisa dan mendesain atau memodelkan suatu sistem karena UML memiliki seperangkat aturan dan notasi dalam bentuk grafis yang cukup spesifik (Sugrue J. 2009).Komponen atau notasi UML diturunkan dari 3 (tiga) notasi yang telah ada sebelumnya yaitu Grady Booch, OOD (*Object-Oriented Design*), Jim Rumbaugh, OMT (*Object Modelling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).(, Mulawarman;, 2011)

1. ***Use Case Diagram***

Setelah pemrograman berorientasi objek menjadi mapan pada tahun 1990-an, para ahli metodologi mulai mengadopsi beberapa konsepnya dan membuatnya menjadi analisis sistem berorientasi objek. Pada awalnya metode ini tidak bisa menggantikan analisis terstruktur secara keseluruhan. Ada aspek yang tidak terpenuhi yaitu dalam hal spesifikasi sistem eksternal atau kebutuhan pengguna. Sebagian analis mengadopsi sebagian komponen metode berorientasi objek dan memadukannya dengan analisis terstruktur. Hal ini dilakukan karena perwakilan pengguna tidak dapat mengerti dokumentasi yang dihasilkan jika menggunakan analisis berorientasi objek murni.

Pada tahun 1992, Ivar Jacobson memperkenalkan use case yang jika diamati serupa dengan cara naratif dari tahun 1960-an dalam melakukan spesifikasi sistem aplikasi. Metode ini berhasil melengkapi metodologi berorientasi objek dari sisi analisis kebutuhan. Dengan adanya use case maka sisi lemah metodologi berorientasi objek dapat dilengkapi.

Sejak tahun 1990-an, use case dengan cepat menjadi praktek yang paling banyak digunakan dalam menangkap kebutuhan fungsional. Hal ini khususnya terjadi pada komunitas berorientasi objek di mana metode ini berasal. Penerapannya tidak hanya terbatas pada sistem berorientasi objek saja karena use case sebenarnya tidak bersifat berorientasi objek.

*Use Case Diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Shalahuddin & Rosa, 2011:130). *Use Case Diagram* digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (*user*), sehingga pembuatan *use case diagram* lebih dititik beratkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Sebuah *use case diagram* merepresentasikan sebuah interaksi antara aktor dengan sistem.

**Tabel 2.1.Keterangan Simbol-Simbol *Use Case Diagram***

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Gambar** | **Nama** | **Keterangan** |
| 1. |  | *Actor* | Menspesifikasikan himpunan 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 sebagai elemen yang tidak mandiri (*independent*). |
| 3. |  | *Generalization* | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada diatasnya (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 *actor*. |
| 9. |  | *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerjasama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |
| 10. |  | *Note* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi. |

Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggilkan setiap kali *use case* yang meng-*include­* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common.* Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain*.*

1. ***Activity Diagram***

*Activity Diagram* menjelaskan bahwa diagram aktivitas atau *activitydiagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau sebuah proses. Yang perlu diperhatikan di sini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dilakukan oleh sistem (Shalahuddin &Rosa, 2011:134).

**Tabel 2.2 Simbol-simbol *Activity Diagram***

|  |  |  |
| --- | --- | --- |
| **No.** | **Simbol** | **Keterangan** |
|  | Description: act_simbol_start.jpg | ***Start*** merupakan status awal sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
|  | Description: act_simbol_activity.jpg | ***Activity*** merupakan aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
|  | Description: act_simbol_decision.jpg | ***Decision*** merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
|  | Description: act_simbol_swimlane.jpg | ***Swimlane*** digunakan untuk memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |
|  | Description: act_simbol_finish.jpg | ***Finish*** merupakan status akhir sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
|  | Description: act_simbol_penggabungan.jpg | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
|  | Description: act_simbol_join.jpg | ***Join*** digunakan untuk menunjukkan kegiatan yang digabungkan |
|  | Description: act_simbol_fork.jpg | ***Fork*** digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel. |

1. ***Squence Diagram***

*Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendesktripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu (Shalahuddin &Rosa, 2011:137).

**Tabel 2.3 Simbol-simbol *Sequence Diagram***

|  |  |  |
| --- | --- | --- |
| **No.** | **Simbol** | **Keterangan** |
|  |  | ***Actor*** adalah pesan dari seseorang atau sistem lain yang bertukar informasi dengan sistem yang lainnya, kemudian *lifeline* berhentu atau mulai pada titik yang tepat. |
|  |  | ***Object Life Line*** menunjukkan keberadaan dari sebuah objek terhadap waktu. Yaitu objek dibuat atau dihilangkan selama suatu periode waktu diagram ditampilkan, kemudian *lifeline* berhenti atau mulai pada titik yang tepat. |
|  |  | ***Activation*** menampilkan periode waktu selama sebuah objek atau aktor melakukan aksi. Dalam *object lifeline*, *activation* berada diatas *lifeline* dalam bentuk kotak persegi panjang, bagian atas dari kotak merupakan inisialisasi waktu dimulainya suatu kegiatan dan yang dibawah merupakan akhir dari waktu. |
|  |  | ***Message*** adalah komunikasi antar objek yang membawa informasi dan hasil pada sebuah aksi. *Message* menyampaikan dari *lifeline* sebuah objek kepada *lifeline* yang lain, kecuali pada kasus sebuah *message* dari objek kepada objek itu sendiri, atau dengan kata lain *message* dimulai dan berakhir pada *lifeline* yang sama. |
|  | Description: sq_boundary.jpg | ***Boundary*** digunakan untuk menggambarkan sebuah form. |
|  | Description: sq_control.jpg | ***Control*** digunakan untuk menghubungkan *boundary* dengan tabel. |
|  | Description: sq_entity.jpg | ***Entity*** digunakan untuk menggambarkan hubungan kegiatan yang akan dilakukan. |

1. ***Class Diagram***

*Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh satu kelas, sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas (Shalahuddin &Rosa, 2011:122).

*Class diagram* adalah sebuah spesifikasi yang akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut / properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode / fungsi). *Class diagram* menggambarkan struktur dan deskripsi *Class, Package* dan objek beserta hubungan satu sama lain seperti pewarisan, asosiasi, dan lain-lain.

**Tabel 2.4 Simbol-simbol *Class Diagram***

|  |  |  |
| --- | --- | --- |
| **No.** | **Simbol** | **Keterangan** |
|  | Description: class_package.jpg | ***Package*** merupakan sebuah bungkusan dari satu atau lebih kelas. |
|  |  | ***Class*** digambarkan dengan bentuk persegi panjang yang dibagi kedalam ruang-ruang terpisah yang terdiri dari nama *class*, atribut, dan operasi-operasinya. |
|  |  | ***Associations*** adalah representasi / gambaran relasi statis diantara *class-class.* |
|  | Description: class_interface.jpg | ***Interface*** sama dengan *interface* dalam pemrograman berorientasi objek. |
|  |  | ***Associations berarah*** merupakan relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity.* |
|  |  | ***Generalisasi*** merupakan relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus). |
|  |  | ***Depedency*** merupakan relasi antar kelas dengan makna kebergantungan antar kelas. |
|  |  | ***Aggregation*** merupakan relasi antar kelas dengan makna semua-bagian (*whole-part*). |

#### Basis Data

Database atau sering kita kenal basis data merupakan sekumpulan data yang tersusun dan tersimpan rapi dalam computer, dan dapat diolah maupun dimanipulasi dengan menggunakan software atau perangkat lunak untuk dijadikan sebagai informasi. Database adalah kumpulan informasi atau data yang tersimpan secara sistematis sehingga temu kembali informasinya menjadi mudah dan cepat (Kusmayadi 2011).

(Fathansyah 2004, 2) menyatakan beberapa pengertian database yaitu sebagai berikut :

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganiasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansasi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan file/table/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

Menurut Connolly dan Begg (2010, 65), database adalah sekumpulan data tersebar yang berhubungan secara logis, dan penjelasan dari data ini dirancang untuk memenuhi kebutuhan informasi dari suatu organisasi.

Menurut Inmon (2005, 493), database adalah sekumpulan data yang saling berhubungan yang disimpan (biasanya dengan redudansi yang terkontrol dan terbatas) berdasarkan skema. Sebuah database dapat melayani single atau multiple applications.

Menurut Gottschalk dan Saether dalam jurnal (2010, 41), database adalah

sekumpulan data yang terorganisir untuk mendukung banyak aplikasi secara efisien dengan memusatkan data dan mengontrol data redundant.

Menurut (Al-Bahra Bin Ladjamudin 2005, 129 ) terdapat beberapa definisi basis data dari beberapa orang ahli basis data sebagai berikut :

1. Database adalah sekumpulan data store (bisa dalam jumlah yang sangat

besar) yang tersimpan dalam magnetic disk, optical disk, magnetic drum

atau media penyimpanan sekunder lainnya.

1. Database adalah sekumpulan program – program aplikasi umum yang

bersifat “*batch*” yang mengeksekusi dan memproses data secara umum

(seperti pencarian, peremajaan, penambahan, dan penghapusan terhadap

data ).

1. Database terdiri dari data yang akan digunakan atau diperuntukan terhadap banyak user, dimana masing – masing user ( baik menggunakan teknik pemrosesan yang bersifat *batch* atau *on-line*) akan menggunakan data tersebut sesuai dengan tugas dan fungsinya, dan user lain dapat juga

menggunakan data tersebut dalam waktu yang bersamaan.

1. Database adalah koneksi terpadu dari data – data yang saling berkaitan dari suatu enterprise ( perusahaan, instansi pemerintah atau swasta ).

Database adalah koleksi atau kumpulan data yang mekanis, terbagi/shared, terdefinisi secara formal dan dikontrol terpusat pada organisasi (*Everest*). Selanjutnya menurut C.J. Date database adalah koleksi “data operasional” yang tersimpan dan dipakai oleh sistem aplikasi dari suatu organisasi.

1. Data input adalah data yang masuk dari luar sistem
2. Data output adalah data yang dihasilkan sistem
3. Data operasional adalah data yang tersimpan pada sistem

Dari pengertian di atas dapat dinyatakan bahwa database yaitu kumpulan-kumpulan data yang berisi informasi yang terhubung satu sama lain yang diorganisasikan dengan struktur tertentu serta dapat ditemukan dengan mudah dan cepat menggunakan bantuan komputer.(Kusnayadi;, 2011)

##### Tipe Database

Untuk menyimpan ataupun mengambil data dari basis data kita memerlukan perangkat lunak yang sering disebut dengan DBMS (system managemen basis data). Adapun tipe database ada sekurang-kurangnya 12 tipe yaitu antara lain:

1. Analytical database, yang merupakan database untuk menyimpan informasi dan data yang diambil dari operasional dan eksternal database. Database ini terdiri dari data dan informasi yang diringkas dan paling banyak dibutuhkan oleh suatu organisasi manajemen maupun End-user lainnya.
2. Operational database ialah database yang menyimpan data secara rinci yang dibutuhkan untuk mendukung operasi dari seluruh organisasi. Biasa juga disebut dengan SADB (subject-area databases), transaksi, dan produksi database. Contohnya seperti : database inventaris, database pribadi, database pelanggan, akuntansi database.
3. Distributed database merupakan kelompok kerja lokal database dan departemen di kantor-kantor dan lokasi kerja yang lainnya. Dalam database ini terdapat dua segmen yaitu user database dan operasional yang datanya digunakan dan duhasilkan hanya pada pengguna situs itu sendiri.
4. Data *warehouse* yaitu sebuah data *warehouse* yang menyimpan data dari tahun-tahun sebelumnya hingga saat ini. Data warehouse merupakan sumber utama data yang telah terintegrasi sehingga bisa digunakan dan dimanfaatkan oleh para pengguna seluruh organisasi yang profesional. Perkembangan yang terjadi akhir ini dari data *warehouse* ialah dipergunakan sebagai *Shared nothing architecture* untuk mendukung dan memfasilitasi *ekstrem scalling.*
5. *Entity Relationship Diagram* (ERD)

ERD merupakan suatu model jaringan yang menggunakan susunan data yang disimpan pada sistem secara abstrak. ERD juga menggambarkan hubungan antara satu entitas yang lain dalam suatu sistem yang terintegrasi. ERD digunakan oleh perancangan sistem untuk memodelkan data yang nantinya akan dikembangkan menjadi basis data (*database*). ERD juga merupakan model konseptual yang dapat mendiskripsikan hubungan antara file yang digunakan untuk memodelkan struktur data serta hubungan antar data (Yakub, 2008).

**Tabel 2.5. Simbol-simbol *Entity Relationship Diagram* (ERD)**

|  |  |  |
| --- | --- | --- |
| **No.** | **Notasi** | **Keterangan** |
|  | Description: entity | Entitas adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai. |
|  | Description: relationship | Relasi yaitu menunjukkan adanya hubungan sejumlah entitas yang berbeda. |
|  | Description: attribute | Atribut berfungsi untuk mendeskripsikan karakter entitas (atribut yang berfungsi sebagai *key* diberi garis bawah). |
|  |  | Garis yaitu sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut. |

Berikut adalah penggambaran relasi antar himpunan entitas lengkap dengan kardinalitas relasi :

1. Relasi satu-ke-satu (*one-to-one*)

Hubungan satu ke satu entitas atau atribut adalah jenis hubungan yang hanya dapat dilakukan satu entitas atau satu atribut dengan satu entitas atau atribut yang lainnya.



**Gambar 2.3. Relasi *One-to-One***

1. Relasi satu-ke-banyak (*one-to-many*)

Hubungan satu ke banyak entitas atau atribut adalah jenis hubungan yang hanya dapat dilakukan satu entitas atau satu atribut dengan beberapa entitas atau atribut yang lainnya.



**Gambar 2.4. Relasi *One-to-Many***

1. Relasi banyak-ke-banyak (*many-to-many*)

Hubungan banyak ke banyak entitas atau atribut adalah jenis hubngan yang hanya dapat dilakukan satu entitas atau satu atribut dengan beberapa entitas atau atribut yang lainnya dan satu entitas atau satu atribut dengan beberapa entitas atau atribut yang lainnya.

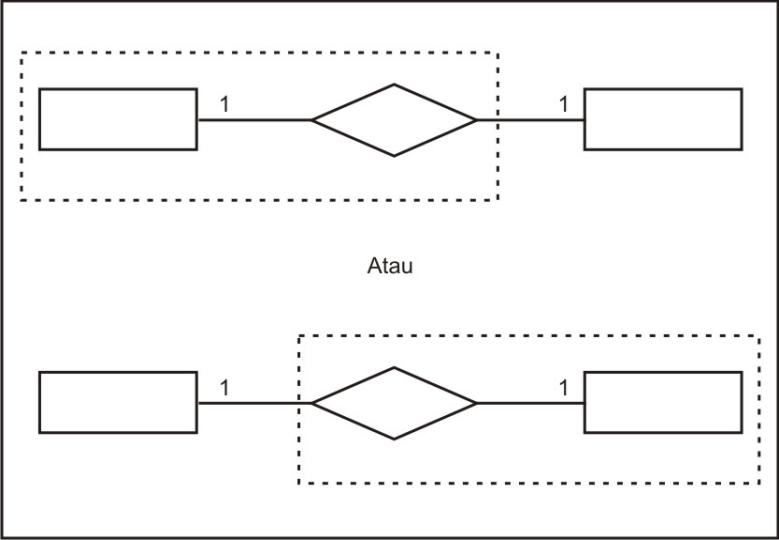


**Gambar 2.5. Relasi *Many-to-Many***

1. Transformasi ERD ke LRS

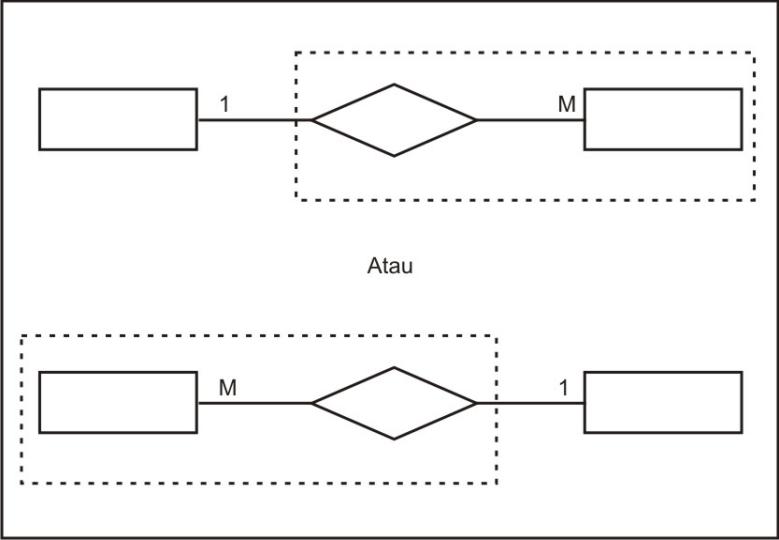
Sebuah model sistem yang digambarkan dengan sebuah ERD akan mengikuti pola aturan pemodelan tertentu. Dalam kaitannya dengan konversi ke LRS, maka perubahan yang terjadi adalah mengikuti aturan-aturan berikut ini.

1. Perhatikan kardinalitas karena sangat berpengaruh pada transformasi.
2. Transformasi *one-to-one* (1 : 1). Pedoman relasinya adalah ke arah *weak entity*, dan ke entitas yang membutuhkan referensi atau ke entitas dengan jumlah atribut yang lebih sedikit.



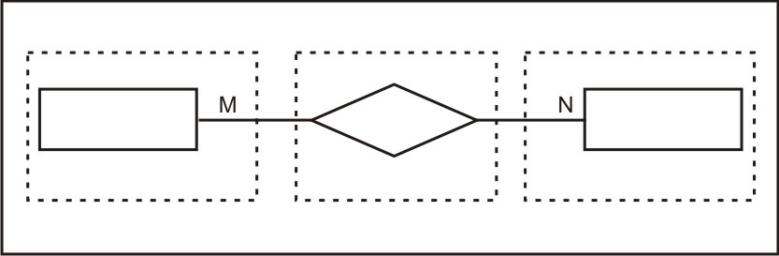
**Gambar 2.6. Transformasi *one-to-one (1 : 1)***

1. Transformasi *one-to-many* (1 : M). Pedoman relasinya adalah tidak perlu melihat jumlah atribut yang lebih sedikit, dan selalu digabung ke arah *many.*



**Gambar 2.7. Transformasi *one-to-many (1 : M)***

1. Transformasi *many-to-many* (M : N). Relasinya berdiri sendiri atau membentuk tabel sendiri.

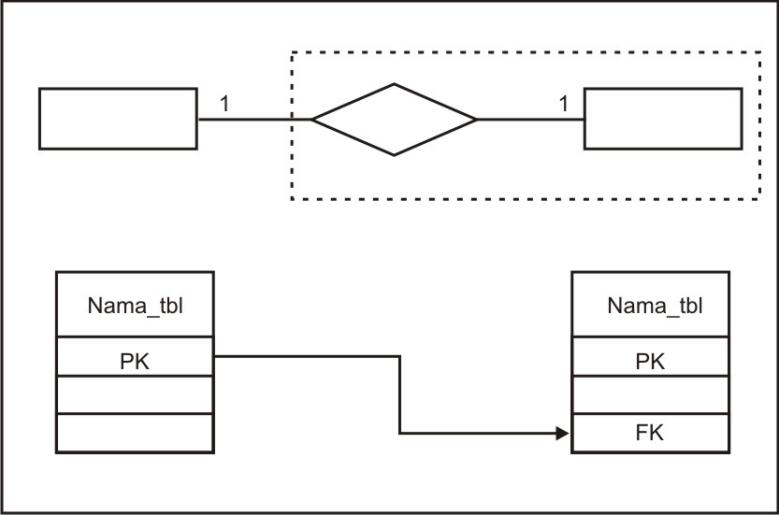


**Gambar 2.8. Transformasi *many-to-many (M : N)***

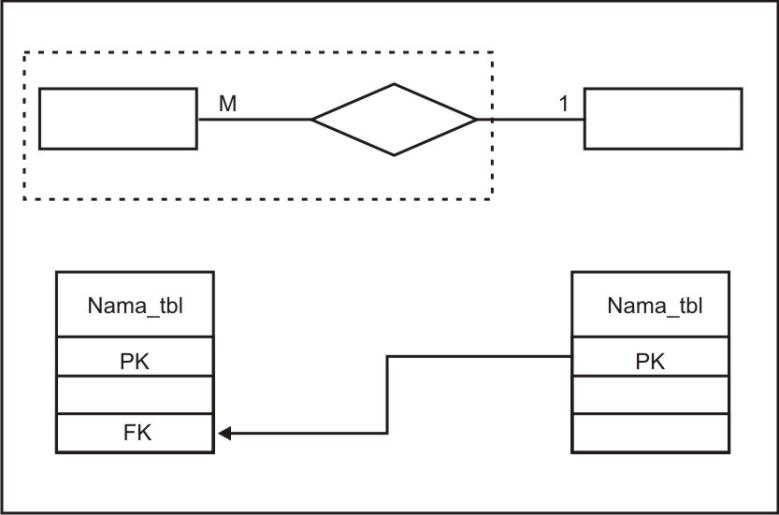
1. *Logical Relationship Structure* (LRS)

LRS adalah representasi dari struktur *record-record* pada *table-table* yang terbentuk dari hasil antar himpunan entitas. Setiap *table* harus memiliki paling sedikit satu *primary key*, dimana sebuah *primary key* merupakan bagian dari kelompok atribut yang memberikan nilai yang unik di dalam sebuah *table*. Pedoman membuat LRS, yaitu:

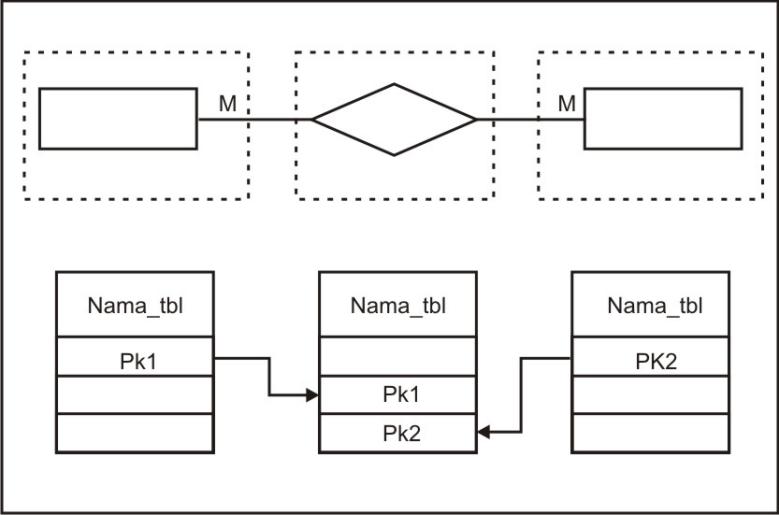
1. Tiap entitas dan relasi (Jika kardinalitas setiap entitas *many-to-many*) juga menjadi sebuah *Logical Record Struktur* (LRS).
2. Nama LRS menjadi nama tabel.
3. Tiap satu atribut menjadi satu kolom.
4. Nama atribut akan menjadi nama kolom.
5. LRS ditentukan oleh relasi antar entitas, seperti berikut:



**Gambar 2.9. LRS dengan transformasi 1: 1**

******

**Gambar 2.10. LRS dengan transformasi 1:M / M:1**

******

**Gambar 2.11. LRS dengan transformasi M:N**

### Software Pendukung

#### Java

Java diciptakan oleh suatu tim yang dipimpin oleh Patrick Naughton dan James Gosling dalam suatu proyek dari Sun Microsystem yang memiliki kode Green dengan tujuan untuk menghasilkan bahasa komputer sederhana yang dapat dijalankan di peralatan sederhana dengan tidak terikat pada arsitekture tertentu. Mulanya disebut OAK, tetapi karena OAK sendiri merupakan nama dari bahasa pemrograman komputer yang sudah ada. Maka Sun mengubahnya menjadi Java.

Sun kemudian meluncurkan browser dari Java yang disebut Hot Java yang mampu menjalankan applet. Setelah itu teknologi Java diadopsi oleh Netscape yang memungkinkan program Java dijalankan di browser Netscape yang kemudian diikuti Internet Explorer. Karena keunikanya dan kelebihanya, teknologi Java mulai menarik banyak vendor seperti IBM,Symantec, Inprise, dll.

Sun merilis versi awal Java secara resmi pada awal tahun 1996 yang kemudian terus berkembang hingga muncul JDK 1.1, kemudian JDK 1.2 yang mulai disebut sebagai versi Java2 karena banyak mengandung peningkatan dan perbaikan. Perubahan utama adalah adanyaSwing yang merupakan teknologi GUI (*Graphical User Interface*) yang mampu menghasilkan window yang portabel.Dan pada tahun 1998 – 1999 lahirlah teknologi J2EE ( *Java 2 Enterprise* Edition ) yang berbasis J2SE yang diawali dengan servlet dan EJB kemudian diikuti JSP. Java juga menjadi lebih cepat populer di lingkungan server side dikarenakan kelebihanya di lingkungan network dan terdistribusi serta kemampuan multithreading. Sedangkan J2ME (*Java 2 Micro Edition*) dapat menghasilkan aplikasi mobile baik games maupun software yang dapat dijalankan di peralatan mobile seperti ponsel.

Java adalah suatu teknologi di dunia software komputer, yang merupakan

suatu bahasa pemrograman, dan sekaligus suatu platform. Sebagai bahasa pemrograman, Java dikenal sebagai bahasa pemrograman tingkat tinggi. Java mudah dipelajari, terutama bagi programmer yang telah mengenal C/C++. Java merupakan bahasa pemrograman berorientasi objek yang merupakan paradigma pemrograman masa depan. Sebagai bahasa pemrograman Java dirancang menjadi handal dan aman.Java juga dirancang agar dapat dijalankan di semua platform. Dan juga dirancang untuk menghasilkan aplikasi – aplikasi dengan performansi yang terbaik, seperti aplikasi database Oracle 8i/9i yang core-nya dibangun menggunakan bahasa pemrograman Java. Sedangkan Java bersifat neutral architecture, karena Java Compiler yang digunakan untuk mengkompilasi kode program Java dirancang untuk menghasilkan kode yang netral terhadap semua arsitekture perangkat keras yang disebut sebagai Java Bytecode.

Sebagai sebuah platform, Java terdiri atas dua bagian utama, yaitu:

1. *Java Virtual Machine* (JVM).
2. *Java Application Programming Interface* (Java API).

Sun membagi arsitektur Java membagi tiga bagian, yaitu:

1. Enterprise Java (J2EE) untuk aplikasi berbasis web, aplikasi sistem tersebar dengan beraneka ragam klien dengan kompleksitas yang tinggi. Merupakan superset dari Standar Java
2. Standar Java (J2SE), ini adalah yang biasa dikenal sebagai bahasa Java.
3. Micro Java (J2ME) merupakan subset dari J2SE dan salah satu aplikasinya yang banyak dipakai adalah untuk wireless device / mobile device.(Hermawan, Asep;, 2015)

#### NetBeans

Netbeans adalah sebuah aplikasi *Integrated Development Environment* (IDE) yang berbasis Java dari Sun Microsystems yang berjalan di atas *swing*. *Swing* merupakan sebuah teknologi Java untuk mengembangkan aplikasi *dekstop* yang dapat berjalan pada berbagai macam *platform* seperti Windows, Linux, Mac OS X dan Solaris. Sebuah IDE merupakan lingkup pemrograman yang diintegrasikan ke dalam suatu aplikasi perangkat lunak yang menyediakan *Graphic User Interface* (GUI), suatu kode editor atau teks, suatu *compiler* dan suatu *debugger.*

Netbeans juga dapat digunakan programmer untuk menulis, meng-*compile*, mencari kesalahan dan menyebarkan program Netbeans yang ditulis dalam bahasa pemrograman Java namun selain itu dapat juga mendukung bahasa pemrograman lainnya dan program ini pun bebas untuk digunakan dan untuk membuat profesional *dekstop*, *enterprise, web,* dan *mobile applications* dengan bahasa Java, C/C++, dan bahkan *dynamic language* seperti PHP, Javascript, Groovy dan Ruby.

Netbeans merupakan sebuah proyek kode terbuka yang sukses dengan pengguna yang sangat luat, komunitas yang terus tumbuh, dan memiliki hampir 100 mitra (dan terus bertambah). Sun Microsystems mendirikan proyek kode terbuka Netbeans pada bulan Juni 2000 dan terus menjadi sponsor utama. Dan saat ini pun Netbeans memiliki 2 produk yaitu Platform Netbeans dan Netbeans IDE.

Platform Netbeans merupakan *framework* yang dapat digunakan kembali (*reusable*) untuk menyederhanakan pengembangan aplikasi dekstop dan Platform Netbeans juga menawarkan layanan-layanan yang umum bagi aplikasi dekstop, mengijinkan pengembangan untuk fokus ke logika yang spesifik terhadap aplikasi. Fitur-fitur yang terdapat dalam Netbeans diantaranya:

1. *Smart Code Completion*; untuk mengusulkan nama variabel dari suatu tipe, melengkapi *keyword* dan mengusulkan tipe parameter dari sebuah *method*.
2. *Bookmarking*; fitur yang digunakan untuk menandai baris yang suatu saat hendak kita modifikasi.
3. *Go to Commands*; fitur yang digunakan untuk *jump* ke deklarasi variabel.
4. *Code Generator*; jika kita menggunakan fitur ini kita dapat meng-*generateconstructor setter and getter method* dan yang lainnya.

#### Mysql

MySQL merupakan bahasa standar untuk mengakses *database*, yaitu menggunakan bahasa *Structured Query Language* (SQL). Sejak tahun 70-an bahasa ini telah dikembangkan oleh IBM, yang kemudian diikuti dengan adanya Oracle, Informix, dan Sybase. Dengan menggunakan SQL, proses akses *database* menjadi lebih *user-friendly* dibandingkan dengan misalnya dbase ataupun Clipper yang masih menggunakan sintaks pemrograman murni. MySQL adalah sebuah *server database* SQL *multiuser* dan *multithreaded*. SQL sendiri adalah salah satu bahasa *database* yang paling populer di dunia. Implementasi program *server database* ini adalah program *daemon* MySQL dan beberapa program lain serta beberapa referensi. Keunggulan MySQL dibandingkan dengan perangkat lunak *database server* lainnya yaitu:

1. *Freeware*, *high performance*, biaya rendah, dan *portable*.
2. Mudah untuk dikonfigurasi dan dipelajari.
3. MySQL merupakan program yang *multi-threaded*, sehingga dipasang pada *server* yang memiliki multi-CPU.
4. Didukung program-program umum seperti C/C++, Java, Perl, PHP, Python, TCL APIs.
5. Bersifat *open source*.
6. Memiliki sistem keamanan yang baik dengan verifikasi host.
7. Memiliki kemampuan mengelola *database* dengan jumlah *record* yang banyak.
8. Mendukung *record* yang memiliki kolom dengan panjang tetap atau panjang bervariasi.

MySQL merupakan program *database server* dengan SQL sebagai bahasa yang digunakan didalamnya. SQL dibagi menjadi dua bentuk *query* (Nugroho, 2008)*,* yaitu:

1. DDL (*Data Definition Language*)

DDL merupakan sebuah metode *query* SQL yang berguna untuk mendefinisikan data pada sebuah *database*, adapun *query* yang dimiliki adalah:

1. CREATE: digunakan untuk melakukan pembuatan tabel dan *database*.
2. DROP: digunakan untuk melakukan penghapusan tabel maupun *database.*
3. ALTER: digunakan untuk melakukan pengubahan struktur tabel yang telah dibuat, baik menambah *field* (add), mengganti nama *field*(change), atapun menamakannya kembali (rename), serta menghapus (drop).
4. DML (*Data Manipulation Language*)

DML merupakan sebuah metode *query* yang dapat digunakan apabila DDL telah terjadi, sehingga fungsi dari *query* ini adalah untuk melakukan pemanipulasian *database* yang telah ada atau yang telah dibuat sebelumnya. Berikut ini *query* yang termasuk didalamnya adalah:

1. SELECT: digunakan untuk menampilkan data pada tabel.
2. INSERT: digunakan untuk melakukan penginputan/ pemasukan data pada tabel dalam database.
3. UPDATE: digunakan untuk melakukan pengubahan atau peremajaan terhadap data yang ada pada tabel.
4. DELETE: digunakan untuk melakukan penghapusan data pada tabel.

#### Xampp

XAMPP merupakan sebuah tool yang menyediakan beberapa paket perangkat lunak ke dalam satu buah paket. Dengan menginstal XAMPP, tidak perlu lagi melakukan instalasi dan konfigurasi *web server* Apache, PHP, dan MySQL secara manual. XAMPP akan menginstalasi dan mengonfigurasinya secara otomatis. (<http://www.apachefriends.org/>).

### Metode Pengujian Sistem

#### Whitebox Testing

*White box testing* yaitu menguji perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi, masukan dan keluaran yang sesuai desain spesifikasi kebutuhan. Pengujian kotak putih dilakukan dengan memeriksa logis dari kode program (Shalahuddin & Rosa, 2011). Pengujian White Box merupakan pengujian untuk memperlihatkan cara kerja dari produk secara rinci sesuai dengan spesifikasinya. Metode pengujian dengan menggunakan struktur kontrol program untuk memperoleh kasus uji.

Dengan menggunakan white box akan didapatkan kasus uji yang :

1. Menjamin seluruh jalur independen di dalam modul yang dieksekusi sekurang-kurangnya sekali.
2. Menguji semua keputusan logikal.
3. Menguji seluruh Loop yang sesuai dengan batasannya.
4. Menguji seluruh struktur data internal yang menjamin validitas.

#### Blackbox Testing

Pengujian *black box* menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program (Shalahuddin & Rosa, 2011). Pengujian Black Box merupakan pengujian untuk mengetahui apakan semua fungsi perangkat lunak telah berjalan semestinya sesuai dengan kebutuhan fungsional yang telah didefinisikan. Metode Black Box memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.(Shalahudin, Muhammad; Rosa, Ariani;, 2011)

Black Box dapat menemukan kesalahan dalam kategori berikut:

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan interface.
3. Kesalahan dalam struktur data atau akses basis data eksternal.
4. Inisialisasi dan kesalahan terminasi.
5. Validitas fungsional.
6. Kesensitifan sistem terhadap nilai input tertentu.
7. Batasan dari suatu data.

**BAB III**

**ANALISA DAN PERANCANGAN**

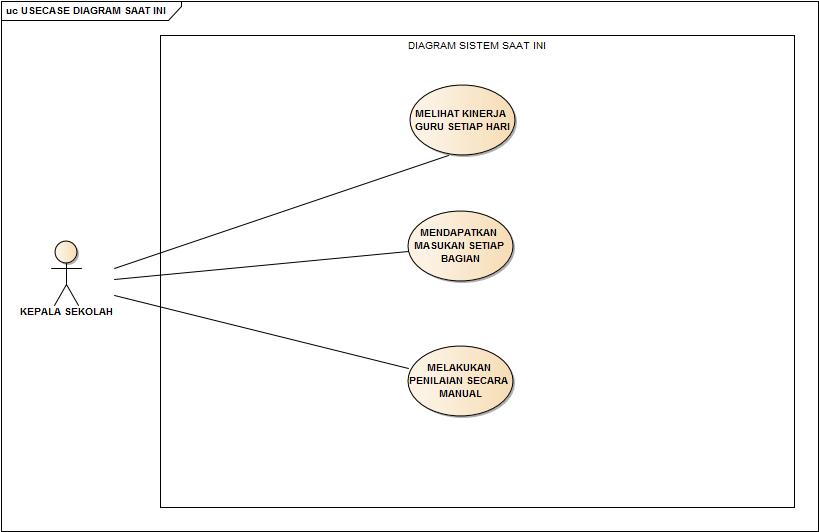
## Analisa

Analisa sistem merupakan proses penguraian suatu sistem informasi yang utuh dan nyata ke dalam komponen-komponen yang bertujuan untuk mengidentifikasikan serta mengevaluasi masalah-masalah yang muncul, hambatan-hambatan yang sering terjadi, serta kebutuhan yang diharapkan, sehingga dapat memberikan solusi untuk pengembangan ke arah yang lebih baik dan sesuai dengan kebutuhan perkembangan teknologi.

### Analisa Sistem Saat ini

Prosedur dalam memberikan bonus menurut kinerja guru adalah dengan menilai kinerja guru dengan kriteria yaitu, Absensi, Kedisipilinan, Tanggung Jawab, Kemampuan Motivasi, Prestasi. Pada saat proses penilaian kinerja guru selalu mendapat kendala dalam setiap memutuskan guru yang akan mendapatkan bonus, kendalanya penilaian dilakukan oleh Kepala Sekolah secara langsung dengan melihat kinerja guru setiap harinya. Kemudian Kepala Sekolah melakukan penilaian dengan mengisi sebuah *file* berupa *form* untuk menentukan guru mana yang mempunyai nilai tertinggi dari beberapa kriteria penilaian tersebut. Sehingga proses penilaian terkadang tidak akurat dan guru dengan penilaian tertinggi mendapatkan bonus.

Saat ini semua proses tersebut belum dilakukan karena belum adanya Sistem Pendukung Keputusan pada SMK Putra Satria, Jakarta Selatan. Analisa sistem manual dilakukan berdasarkan urutan kejadian yang ada dan dari urutan kejadian tersebut dapat dibuat *Diagram* Kepala Sekolah melakukan penilaian guru secara langsung:



**Gambar 3.1 Use Case Diagram Sistem Saat ini**

Dibawah ini merupakan *use case* dari sistem manual pada pemberian bonus guru sebagai berikut :

1. Kepala Sekolah melihat kinerja guru setiap hari
2. Kepala Sekolah mendapatkan masukan dari setiap bagian
3. Melakukan penilaian secara manual

### Evaluasi Sistem Saat Ini

Pada sistem manual, Kepala Sekolah melakukan pemilihan guru terbaik dalam pemberian bonus belum memiliki penilaian yang sesuai, belum juga memiliki kriteria dan bobot yang cukup sehingga pada hasil pemilihan guru terbaik dalam pemberian bonus masih bersifat subjektif. Selain itu membutuhkan waktu yang lumayan untuk mengumpulkan data. Setelah dilakukan penelitian, ditemukan beberapa hal yang terjadi diantaranya :

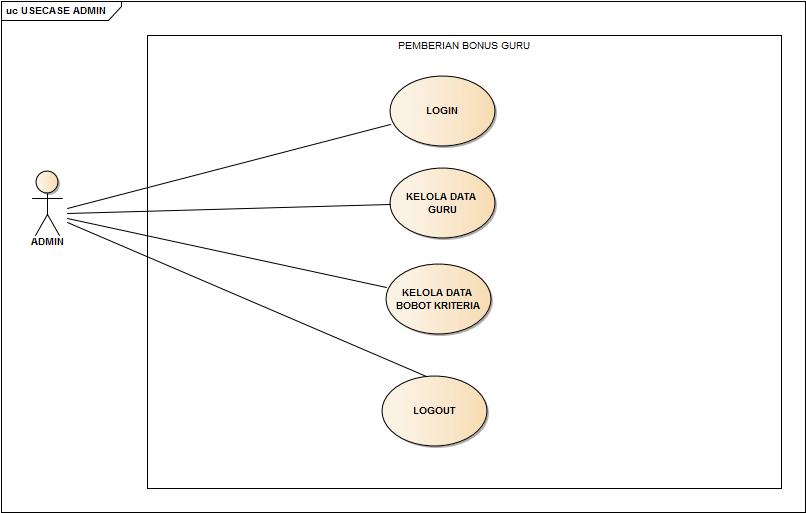
1. Belum adanya sistem yang digunakan sebagai rekomendasi kepada Kepala Sekolah dalam hal melakukan pemilihan guru terbaik dalam pemberian bonus.
2. Kepala Sekolah mendapat kendala menentukan seseorang guru terbaik dalam pemberian bonus karena belum adanya metode penilaian yang dapat memproses penilaian guru terbaik dalam pemberian bonus.
3. Proses penilaian perhitungan berdasarkan kriteria tanpa menggunakan suatu metode tertentu, sehingga dibutuhkan sistem pendukung keputusan untuk penilaian guru terbaik dalam pemberian bonus.

### Analisa Sistem Yang Diusulkan

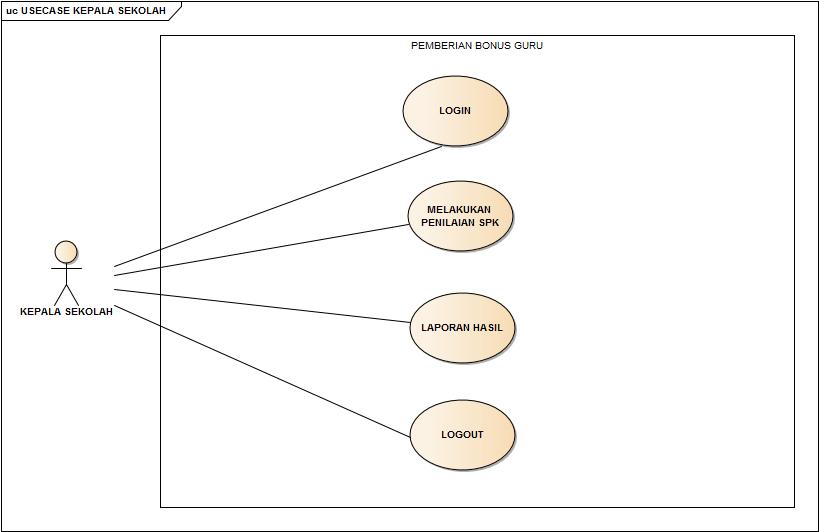
Setelah mengetahui sistem manual yang berjalan dan telah dievaluasi maka usulan pemecahan masalah pada sistem manual tersebut, akan dibuat aplikasi sistem pendukung keputusan dengan metode SAW yang dapat membantu Kepala Sekolah untuk menentukan guru yang paling layak direkomendasikan mendapat bonus.

1. Dibuatnya sistem pengolahan data untuk mempermudah kepala sekolah dalam menentukan penilaian guru terbaik dalam pemberian bonus.
2. Proses seleksi dilakukan menggunakan sistem pendukung keputusan dengan menggunakan metode *SIMPLE ADDITIVE WEIGHTING*

Sistem yang diusulkan untuk menentukan pemilihan guru terbaik dalam pemberian bonus pada SMK Putra Satria Jakarta Selatan dibagi menjadi 2 pengguna, yaitu admin sistem yang mengelola data dan Kepala sekolah sebagai *user* yang melakukan penilaian dan mendapatkan laporan, yang tergambar sebagai berikut:

1.  Admin

**Gambar 3.2. Sistem diusulkan untuk admin**

1.  Kepala Sekolah

**Gambar 3.3. Sistem Diusulkan Untuk Kepala Sekolah**

### Analisa Sistem Pendukung Keputusan dengan Metode SAW

Dalam menentukan siswa terbaik dengan menggunakan metode *Simple Additive Weighting*(SAW) di perlukan kriteria-kriteria dan bobot untuk melakukan perhitungan sehingga akan didapat alternatif terbaik untuk pemilihan siswa terbaik. Kriteria dan bobot pada penilaian siswa pada SMK PUTRA SATRIA, Jakarta Selatan adalah sebagai berikut :

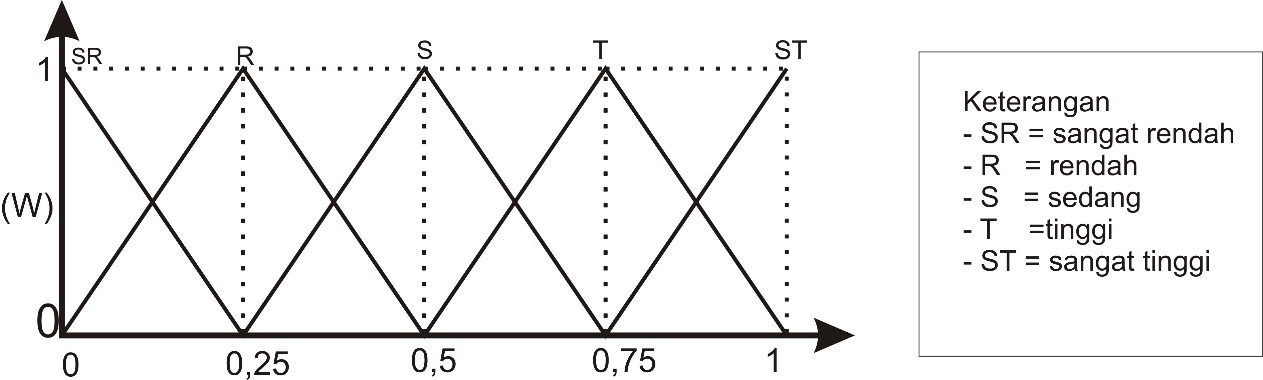
1. Kriteria dan bobot

**Tabel 3.1. Kriteria**

|  |  |
| --- | --- |
| Kriteria | Keterangan |
| C1 | Absensi |
| C2 | Kedisiplinan |
| C3 | Tanggung Jawab |
| C4 | Kemampuan Motivasi |
| C5 | Prestasi |

Dari masing­-masing kriteria tersebut akan ditentukan bobot­bobotnya. Pada bobot terdiri dari enam bilangan fuzzy, yaitu sangat rendah (SR), rendah (R), sedang (S), tengah (T1), tinggi (T2), dan sangat tinggi (ST)

Dari masing­masing bobot tersebut, maka dibuat suatu variabel yang akan dirubah kedalam bilangan fuzzy dengan rumus yaitu variabel ke­n/n­1



**Tabel 3.2 Variabel dan Bobot ( Nilai )**

|  |  |
| --- | --- |
| Variable | Bobot (Nilai) |
| Sangat Rendah (SR) | Variabel ke-0 / (6-1) = 0 |
| Rendah (R) | Variabel ke-1 / (6-1) = 1/5 = 0,25 |
| Sedang (S) | Variabel ke-2 / (6-1) = 2/5 = 0,5 |
| Tinggi (T) | Variabel ke-4/ (6-1) = 4/5 = 0,75 |
| Sangat Tinggi (ST) | Variabel ke-5/ (6-1) = 5/5 = 1 |

Berikut ini adalah analisa terhadap kriteria-kriteria untuk penentuan pemberian bonus.

**Tabel 3.3 Nilai Rata – Rata Absensi**

|  |  |  |
| --- | --- | --- |
| Nilai Rata – rata Absensi (C1) | Variabel | Nilai |
| C1<= 60 | Sangat Rendah (SR) | 0 |
| C1> 61 – 70 | Rendah (R) | 0,25 |
| C1 > 71 – 80 | Cukup (C) | 0,5 |
| C1 > 81 – 90 | Tinggi (T) | 0,75 |
| C1 > 91 | Sangat Tinggi (ST) | 1 |

**Tabel 3.4 Nilai Rata – Rata Kedisiplinan**

|  |  |  |
| --- | --- | --- |
| Nilai Rata – rata kedisiplinan (C2) | Variabel | Nilai |
| C2 <= 65 | Sangat Rendah (SR) | 0 |
| C2 > 66 – 75 | Rendah (R) | 0,25 |
| C2 > 76 – 85 | Cukup (C) | 0,5 |
| C2 > 86 – 95 | Tinggi (T) | 0,75 |
| C2 > 96 | Sangat Tinggi (ST) | 1 |

**Tabel 3.5 Nilai Rata – Rata Tanggung Jawab**

|  |  |  |
| --- | --- | --- |
| Nilai Rata – rata Tanggung Jawab (C3) | Variabel | Nilai |
| C3 <= 60 | Sangat Rendah (SR) | 0 |
| C3 > 61 – 70 | Rendah (R) | 0,25 |
| C3 > 71 – 80 | Cukup (C) | 0,5 |
| C3 > 81 – 90 | Tinggi (T) | 0,75 |
| C3 > 91 | Sangat Tinggi (ST) | 1 |

**Tabel 3.6 Nilai Rata – Rata Kemampuan Motivasi**

|  |  |  |
| --- | --- | --- |
| Nilai Rata – rata kemampuan motivasi (C4) | Variabel | Nilai |
| C4 <= 60 | Sangat Rendah (SR) | 0 |
| C4 > 61 – 70 | Rendah (R) | 0,25 |
| C4 > 71 – 80 | Cukup (C) | 0,5 |
| C4 > 81 – 90 | Tinggi (T) | 0,75 |
| C4 > 91 | Sangat Tinggi (ST) | 1 |

**Tabel 3.7 Nilai Rata – Rata Prestasi**

|  |  |  |
| --- | --- | --- |
| Nilai Rata – rata Prestasi (C5) | Variabel | Nilai |
| C5 <= 65 | Sangat Rendah (SR) | 0 |
| C5 > 66 – 75 | Rendah (R) | 0,25 |
| C5 > 76 – 85 | Cukup (C) | 0,5 |
| C5 > 86 – 95 | Tinggi (T) | 0,75 |
| C5 > 96 | Sangat Tinggi (ST) | 1 |

Berikut ini adalah table Data Guru dalam pemberian Bonus.

**Tabel 3.8 Data Guru pemberian bonus**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Nama | Absensi | Kedisiplinan | Tanggung jawab | Kemempuan Motivasi | Prestasi |
| 1 | Guru A | 75 | 70 | 78 | 85 | 85 |
| 2 | Guru B | 85 | 75 | 98 | 74 | 73 |
| 3 | Guru C | 68 | 85 | 70 | 94 | 85 |
| 4 | Guru D | 70 | 80 | 75 | 89 | 96 |
| 5 | Guru E | 90 | 90 | 91 | 90 | 84 |

Supaya lebih jelas dimisalkan untuk dari Tabel .3.8 ( Data Guru pemberian bonus ) diatas adalah A1 = Guru A, A2 = Guru B, A3 = Guru C, A4 = Guru D Dan A5 = Guru E.

**Tabel 3.9 menunjukkan rating kecocokan dari setiap alternatif pada setiap kriteria**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Alternatif | Kriteria | | | | |
| C1 | C2 | C3 | C4 | C5 |
| A1 | 0,5 | 0,25 | 0,5 | 0,75 | 0,5 |
| A2 | 0,75 | 0,25 | 1 | 0,5 | 0,25 |
| A3 | 0,25 | 0,5 | 0,25 | 1 | 0,5 |
| A4 | 0,25 | 0,5 | 0,5 | 0,75 | 1 |
| A5 | 0,75 | 0,75 | 1 | 0,75 | 0,5 |

Dari Tabel 3.9 diubah kedalam matriks keputusan X dengan data :

Untuk menentukan bobot dari pemberian bonus kurang mampu dibentuk dalam tabel 3.10

**Tabel 3.10 Bobot untuk pemberian bonus**

|  |  |  |
| --- | --- | --- |
| Kriteria | Bobot | Nilai |
| C1 (Absensi) | Tinggi (T) | 0,25 |
| C2 (Kedisiplinan) | Cukup (C) | 0,3 |
| C3 (Tanggung Jawab) | Sedang (S) | 0,15 |
| C4 (Kemampuan Motivasi) | Sedang (S) | 0,15 |
| C5 (Prestasi) | Cukup (C) | 0,15 |

Dari Tabel 3.10 diperoleh nilai bobot (W) dengan data :

W = [0,25 0,3 0,15 0,15 0,15]

Menormalisasi matriks X menjadi matrik R berdsarkan persamaan 1

Keterangan :

*r*ij = nilai rating kinerja ternormalisasi .  
xij = nilai atribut yang dimiliki dari setiap kriteria.  
Maxi xij = nilai terbesar dari setiap kriteri  
Benefit = jika nilai terbesar adalah terbaik

**Tabel 3.11 kriteria ( benefit )**

|  |  |  |
| --- | --- | --- |
| Kriteria | Cost | Benefit |
| Absensi |  |  |
| Kedisiplinan |  |  |
| Tanggung Jawab |  |  |
| Kemampuan Motivasi |  |  |
| Jumlah Saudara |  |  |

* 1. Untuk absensi termasuk ke dalam atribut biaya (benefit), karena semakin besar maka dianggap semakin baik,

Jadi :

=

=

=

=

=

* 1. Untuk kedisiplinan termasuk ke dalam atribut biaya (benefit), karena semakin besar maka dianggap semakin baik,

Jadi :

=

=

=

=

=

* 1. Untuk tanggung jawab termasuk ke dalam atribut biaya (benefit), karena semakin besar maka dianggap semakin baik,

Jadi :

=

=

=

=

=

* 1. Untuk kemampuan motivasi termasuk ke dalam atribut biaya (benefit), karena semakin besar maka dianggap semakin baik,

Jadi :

=

=

=

=

=

* 1. Untuk kemampuan motivasi termasuk ke dalam atribut biaya (benefit), karena semakin besar maka dianggap semakin baik,

Jadi :

=

=

=

=

=

Melakukan proses perangkingan dengan menggunakan persamaan 2 :

Keterangan :  
Vi = integral untuk setiap alternatif  
Wj = nilai bobot dari setiap kriteria  
rij = nilai rating kinerja ternormalisasi  
Jadi :

* + - * 1. Nilai total integral A1

=(0,667)(0,25)+(0,333)(0,3)+(0,5)(0,15)+(0,75)(0,15)+(0,25)(0,15)

=0,1668+0,0999+0,075+0,1125+0,0375=0,4917

* + - * 1. Nilai total integral A2

=(1)(0,25)+(0,333)(0,3)+(1)(0,15)+(0,5)(0,15)+(0,25)

(0,15)

=0,25+0,0999+0,15+0,075+0,0375

=0,6124

* + - * 1. Nilai total integral A3

=(0,333)(0,25)+(0,667)(0,3)+(0,25)(0,15)+(1)(0,15)+

(0,5)(0,15)

=0,0832+0,2001+0,0375+0,15+0,075

=0,5458

* + - * 1. Nilai total integral A4

=(0,333)(0,25)+(0,667)(0,3)+(0,5)(0,15)+(0,75)(0,15)

+(1)(0,15)

=0,0832+0,2001+0,075+0,1125+0,15

=0,6208

* + - * 1. Nilai total integral A5

=(1)(0,25)+(1)(0,3)+(1)(0,15)+(0,75)(0,15)+(0,5)(0,15)

=0,25+0,3+0,15+0,1125+0,075

=0,8875

**Tabel 3.12 Nilai total integer**

|  |  |  |
| --- | --- | --- |
| No | Alternatif | Nilai Total Integral |
| 1 | A1 = siswa A | 0,4917 |
| 2 | A1 = siswa B | 0,6124 |
| 3 | A1 = siswa C | 0,5458 |
| 4 | A1 = siswa D | 0,6208 |
| 5 | A1 = siswa E | 0,8875 |

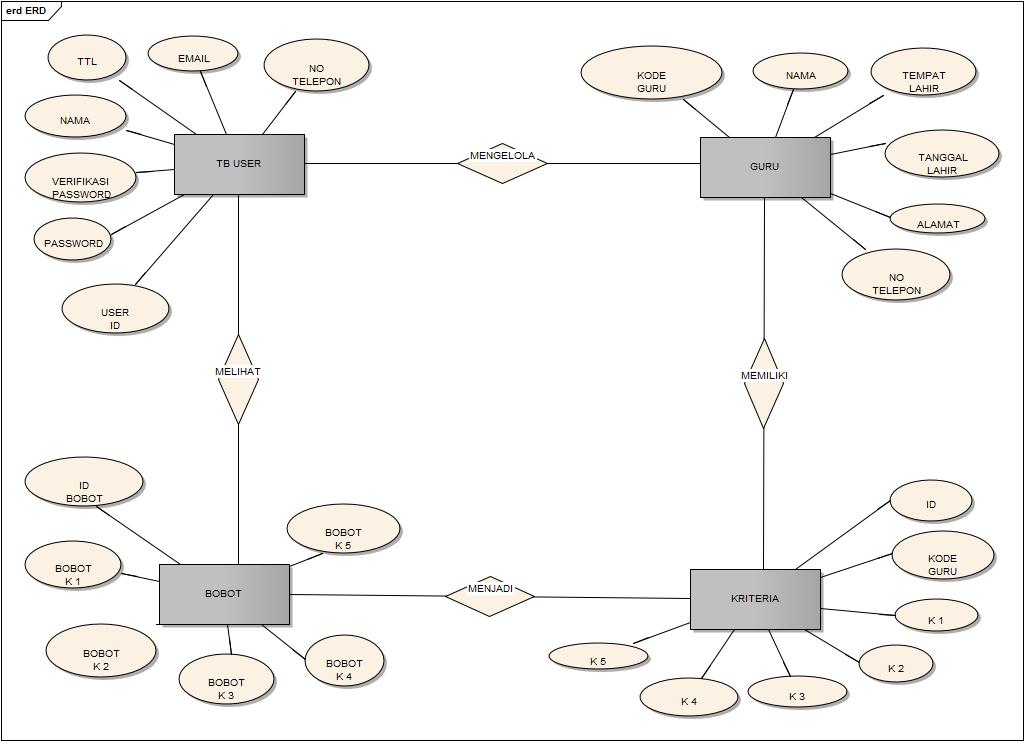
Jadi, yang berhak untuk menerima bonus yang direkomendasikan berdasarkan pencarian menggunakan Simple Addtive Weighting adalah guru E.

## Perancangan Sistem

### Perancangan Basis Data

Perancangan Basis Data meliputi perancangan *Entity Relationship Diagram* (ERD), Transformasi ERD ke LRS, *Logical Record Structure*, dan spesifikasi Basis Data yang digunakan.

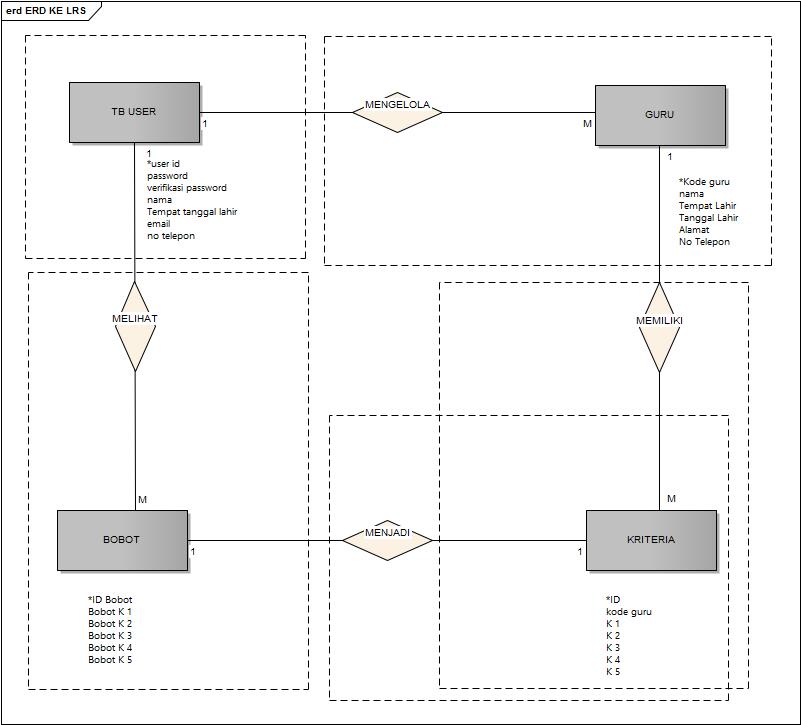
#### Entity Relationship Diagram ( ERD )

*Diagram* ERD diatas menggambarkan struktur *database* yang akan dibangun dalam sistem perhitungan pemberian bonus.

**Gambar 3.4*Entity Relationship Diagram* ( ERD )**

#### Transformasi ERD ke LRS

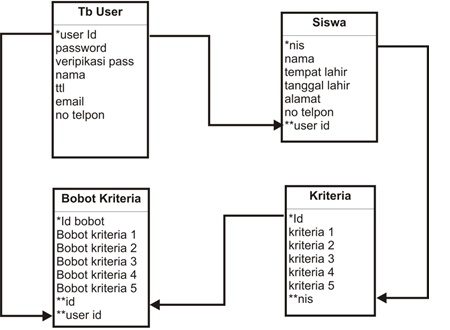
Berikut adalah gambar transformasi ERD (*Entity Relationship Diagram)* diubah kebentuk LRS (*Logical Record Structure*).



**Gambar 3.5 Transformasi ERD ke LRS**

#### Logical Record Structure (LRS)

Setelah ditransformasikan ERD ke LRS, maka bentuk *Logical Record Structure* (LRS) yang sudah terbentuk seperti pada gambar dibawah ini :



**Gambar 3.6*Logical Record Structure* (LRS)**

#### Spesifikasi Basis Data

Spesifikasi basis data mengelompokkan uraian rinci setiap tabel. Tabel adalah komponen utama dan pertama dari sebuah *database.* Struktur tabel untuk setiap tabel yang terdapat dalam *database* sistem ini adalah sebagai berikut :

1. Tabel User

**Tabel 3.13 Tabel user**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Indeks** |
| id\_user | Varchar | 50 | Primary Key |
| Passwaord | Varchar | 50 | - |
| Veripikasi password | Varchar | 50 | - |
| Nama | Varchar | 50 | - |
| Tempat tangal lahir | Varchar | 50 | - |
| Email | Varchar | 50 | - |
| No telpon | Varchar | 15 | - |

1. Tabel Guru

**Tabel 3.14 Tabel guru**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Indeks** |
| Kode Guru | Varchar | 11 | Primary Key |
| Nama | Varchar | 50 | - |
| Tempat lahir | Varchar | 35 | - |
| Alamat | Varchar | 255 | - |
| No telpon | Varchar | 15 | - |

1. Tabel Kriteria

**Tabel 3.15 Tabel kriteria**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Indeks** |
| Id | Int | 10 | Primary Key |
| Kode Guru | Varchar | 11 | - |
| Kriteria 1 | Decimal | - | - |
| Kriteria 2 | Decimal | - | - |
| Kriteria 3 | Decimal | - | - |
| Kriteria 4 | Decimal | - | - |
| Kriteria 5 | Decimal | - | - |

1. Tabel Bobot Kriteria

**Tabel 3.16 Tabel Bobot Kriteria**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Indeks** |
| Id bobot | Int | 10 | Primary Key |
| Bobot Kriteria 1 | Decimal | - | - |
| Bobot Kriteria 2 | Decimal | - | - |
| Bobot Kriteria 3 | Decimal | - | - |
| Bobot Kriteria 4 | Decimal | - | - |
| Bobot Kriteria 5 | Decimal | - | - |

### Perancangan Aplikasi

#### Use Case Diagram



**Gambar 3.7 Use Case Diagram**

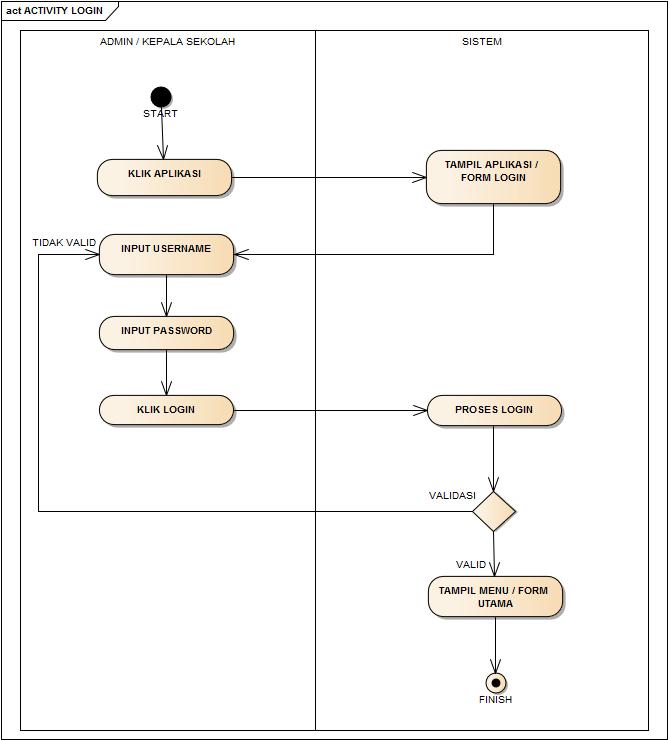
Deskripsi *Use Case* ( Admin ) :

1. Admin melakukan *login* pada sistem penunjang keputusan pemberian bonus berdasarkan kinerja guru.
2. Admin mengelola data Guru sebagai alternatif.
3. Admin mengelola bobot kriteria.
4. Admin melakukan logout

Deskripsi *Use Case*( Kepala Sekolah ) :

1. Kepala Sekolah melakukan *login* pada sistem penunjang keputusan pemberian bonus berdasarkan kinerja guru.
2. Kepala Sekolahmelakukan penilaian SPK.
3. Kepala SekolahMelihat Laporan Hasil.
4. Kepala Sekolah melakukan logout.

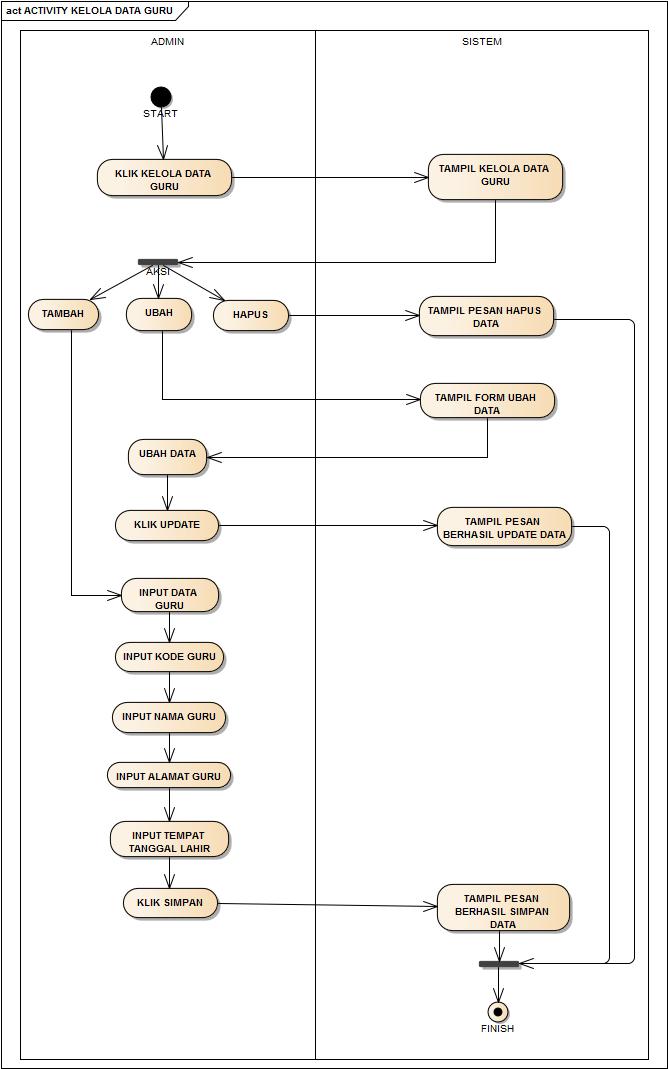
#### Activity Diagram

1. *Acitivity Diagram* Login

**Gambar 3.8*Activity Diagram* Login**

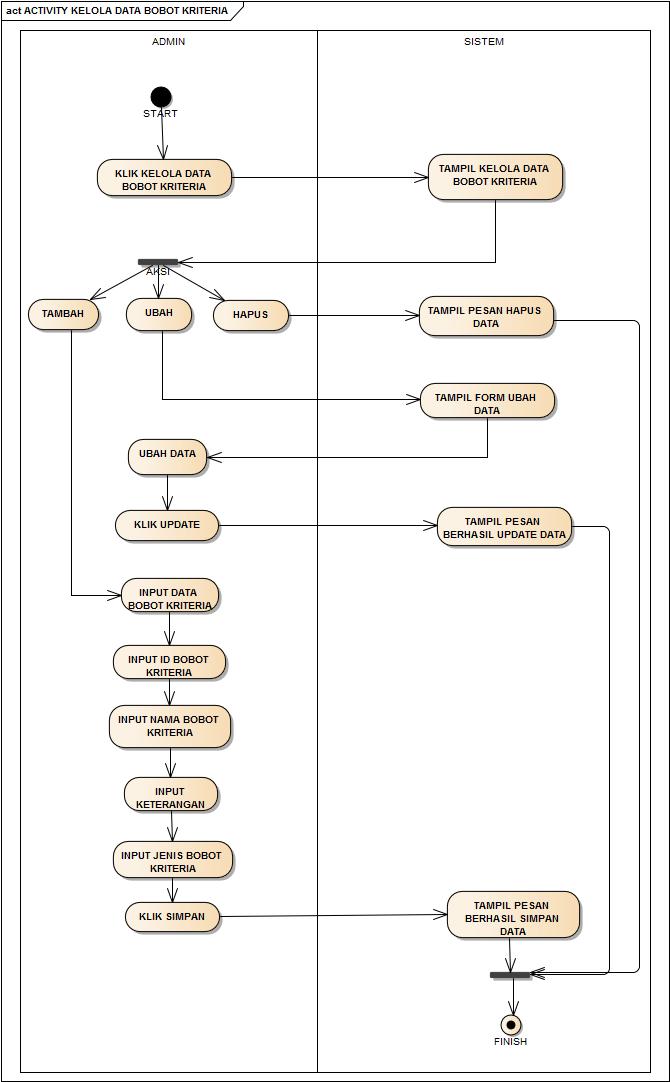
Pada gambar diatas menjelaskan kegiatan *Login* yang dilakukan oleh Admin dan Kepala Sekolah pada sistem penunjang keputusan pemberian bonus berdasarkan kinerja guru.

1. *Activity Diagram* Kelola Data Guru



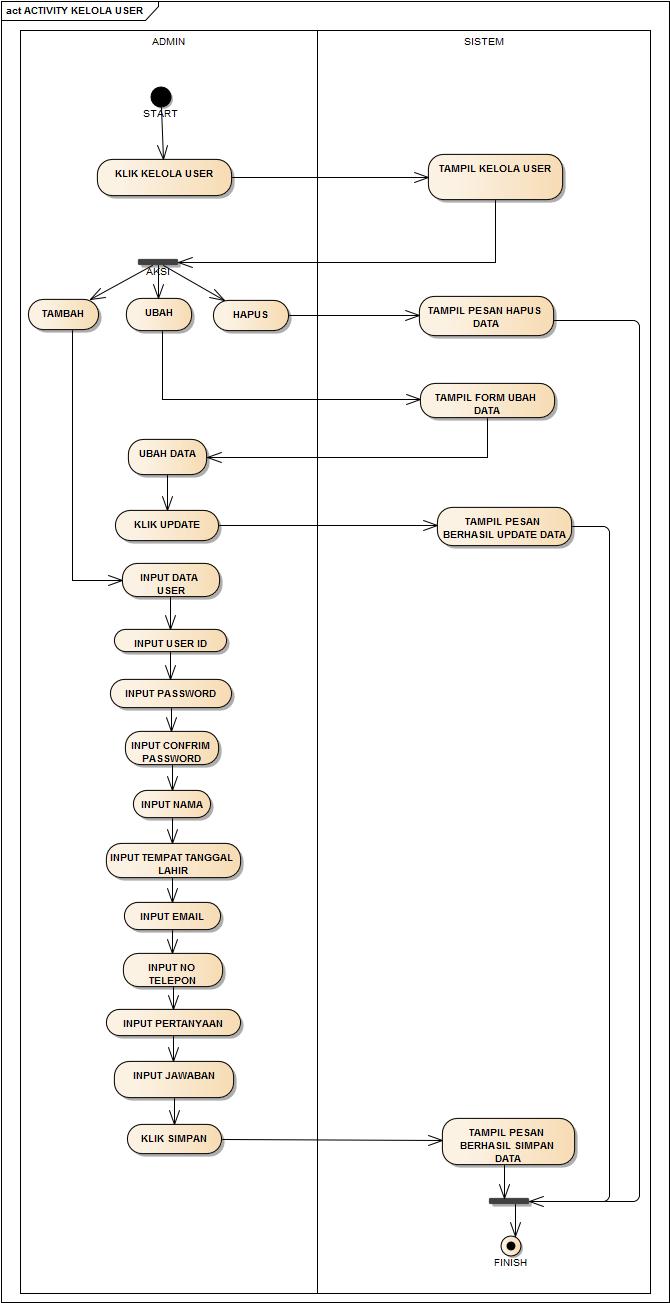
**Gambar 3.9*Activity Diagram* Kelola Data Guru**

Pada gambar diatas menjelaskan kegiatan Kelola Data Guruyang dilakukan oleh Admin pada sistem penunjang keputusan pemberian bonus berdasarkan kinerja guru.

1. *Activity Diagram* Kelola Bobot Kriteria

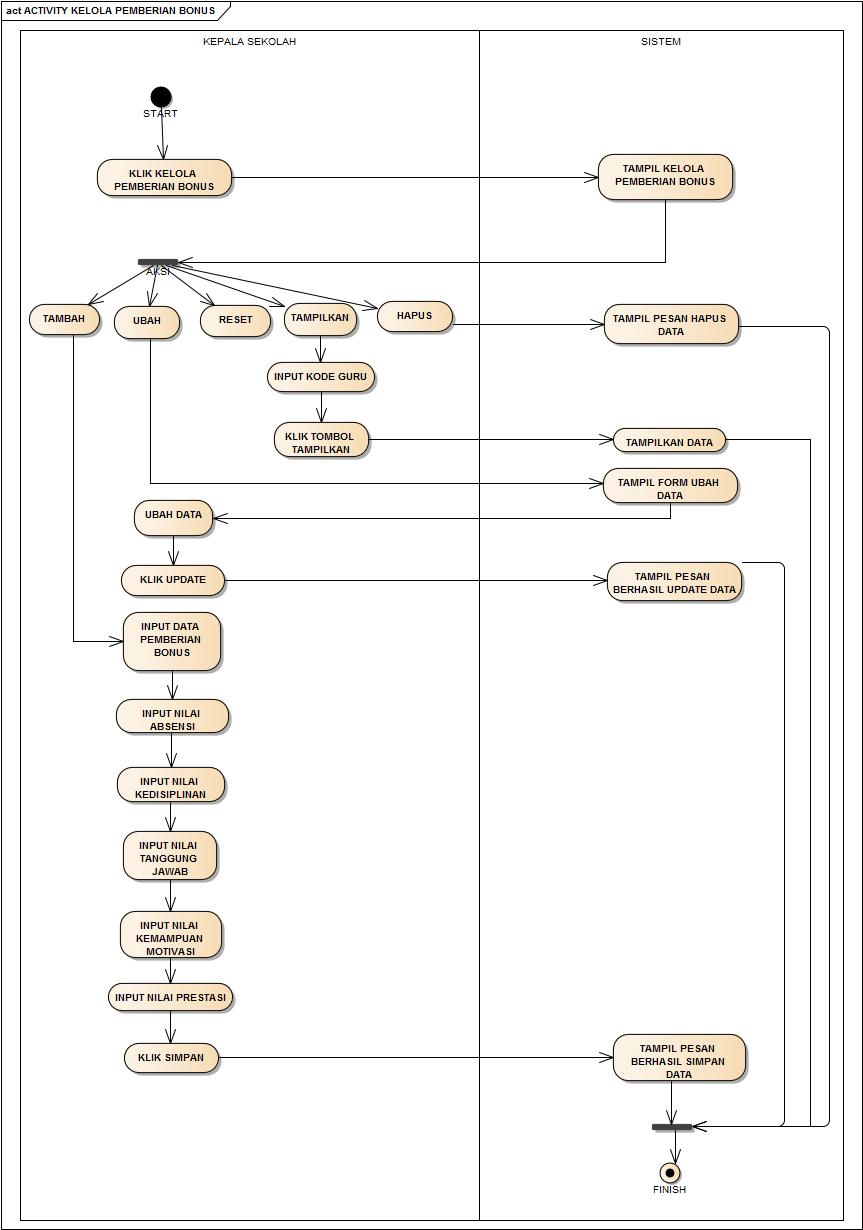
**Gambar 3.10*Activity Diagram* Kelola Bobot Kriteria**

Pada gambar diatas menjelaskan kegiatan Kelola Bobot Kriteriayang dilakukan oleh Admin pada sistem penunjang keputusan pemberian bonus berdasarkan kinerja guru.

1. *Activity Diagran* Kelola User

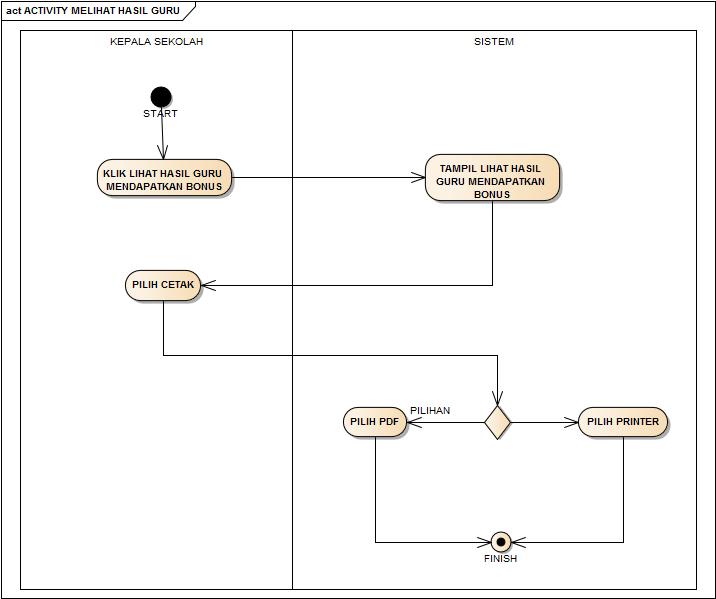
**Gambar 3.11*Activity Diagram* Kelola User**

Pada gambar diatas menjelaskan kegiatan Input Kelola Useryang dilakukan oleh Admin pada sistem penunjang keputusan pemberian bonus berdasarkan kinerja guru.

1. *Activity Diagram* Kelola Pemberian Bonus

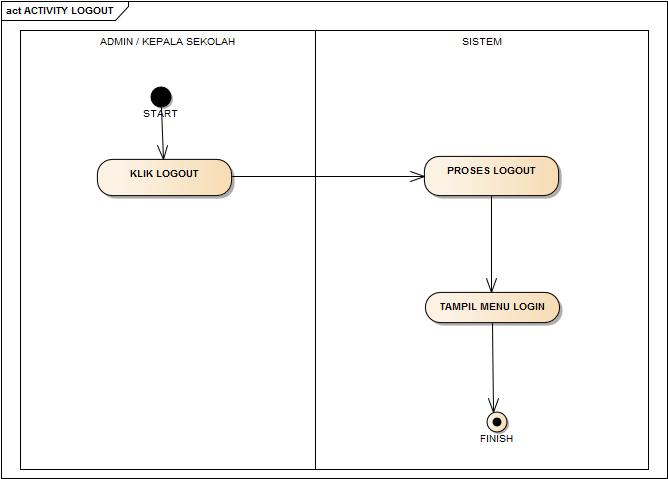
**Gambar 3.12*Activity Diagram* Kelola Pemberian Bonus**

Pada gambar diatas menjelaskan kegiatan melakukan kelola pemberian bonus dengan metode SAW yang dilakukan oleh kepala sekolah pada sistem pemilihan siswa terbaik.

1. *Activity Diagram* Melihat Hasil Guru

**Gambar 3.13 Melihat Hasil Guru**

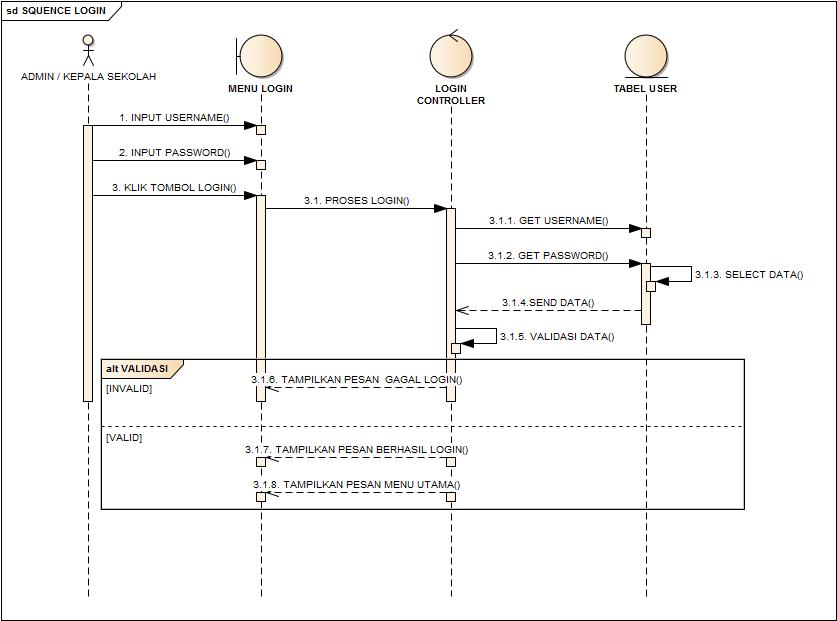
Pada gambar diatas menjelaskan kegiatan menampilkan laporan hasil penilaian yang dilakukan oleh Kepala Sekolah pada sistem pemberian bonus.

1. *Activity Diagram Logout*

**Gambar 3.14*Activity Diagram Logout***

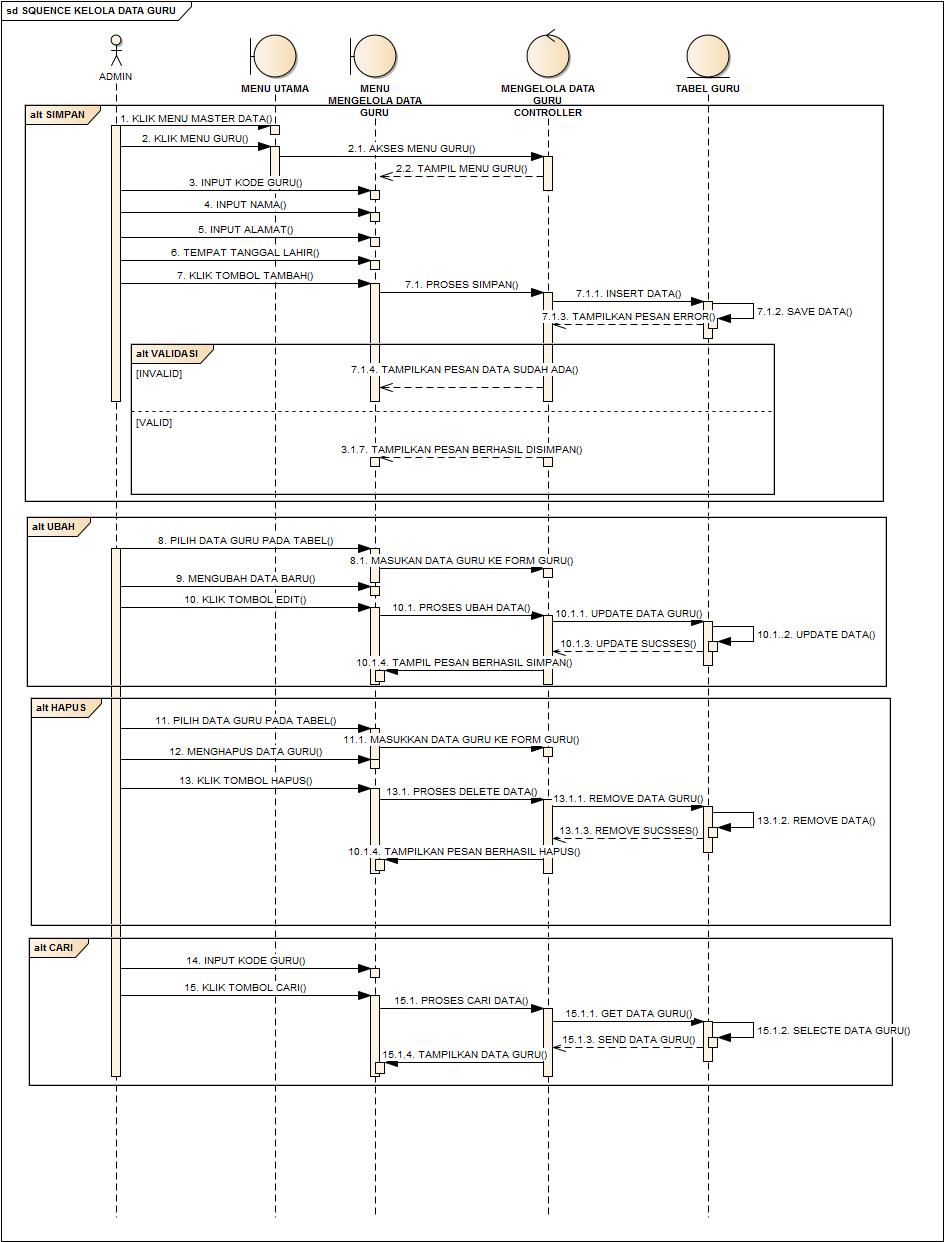
Pada gambar diatas menjelaskan kegiatan tentang Admin dan Kepala Sekolah melakukan Logout.

#### Squence Diagram

1. *Squence Diagram Login*

**Gambar 3.15*Squence Diagram Login***

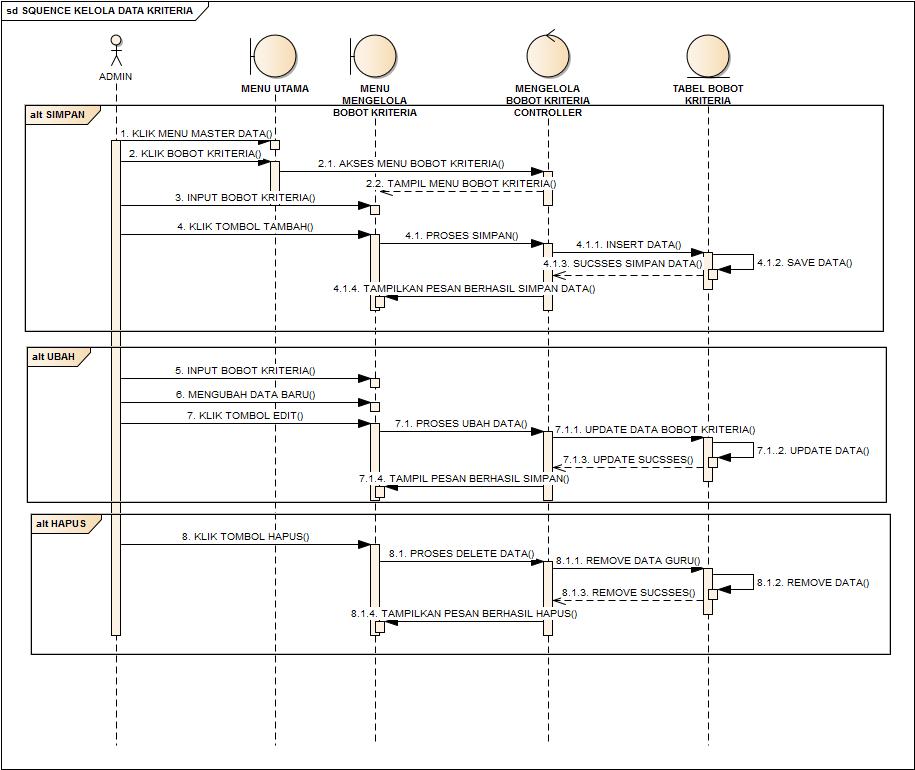
*Sequence diagram* Login diatas merupakan kegiatan yang dilakukan pada *form* Login dimana Admin / Kepala Sekolah memasukan *username* dan *password*. *Username* dan *password* harus sesuai dengan data yang ada pada *database*.

1. *Squence Diagram* Kelola Data Guru

**Gambar 3.16*Squence Diagram* Kelola Data Guru**

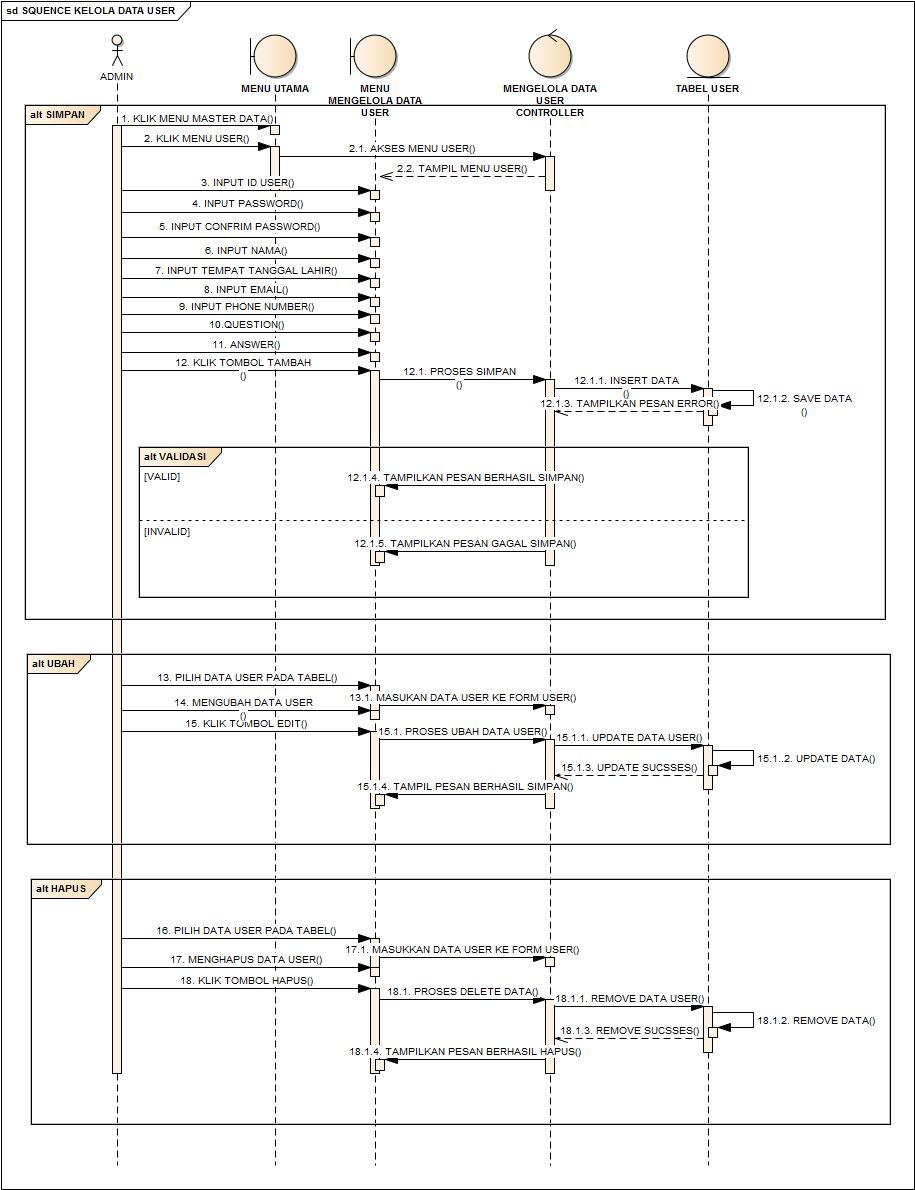
*Sequence diagram* Kelola Data Guru diatas merupakan kegiatan yang dilakukan pada Halaman Kelola Data Guru dimana Admin dapat menambah, mengubah, dan menghapus data Guru.

1. *Squence Diagram* Kelola Bobot Kriteria



**Gambar 3.17*Squence Diagram* Kelola Bobot Kriteria**

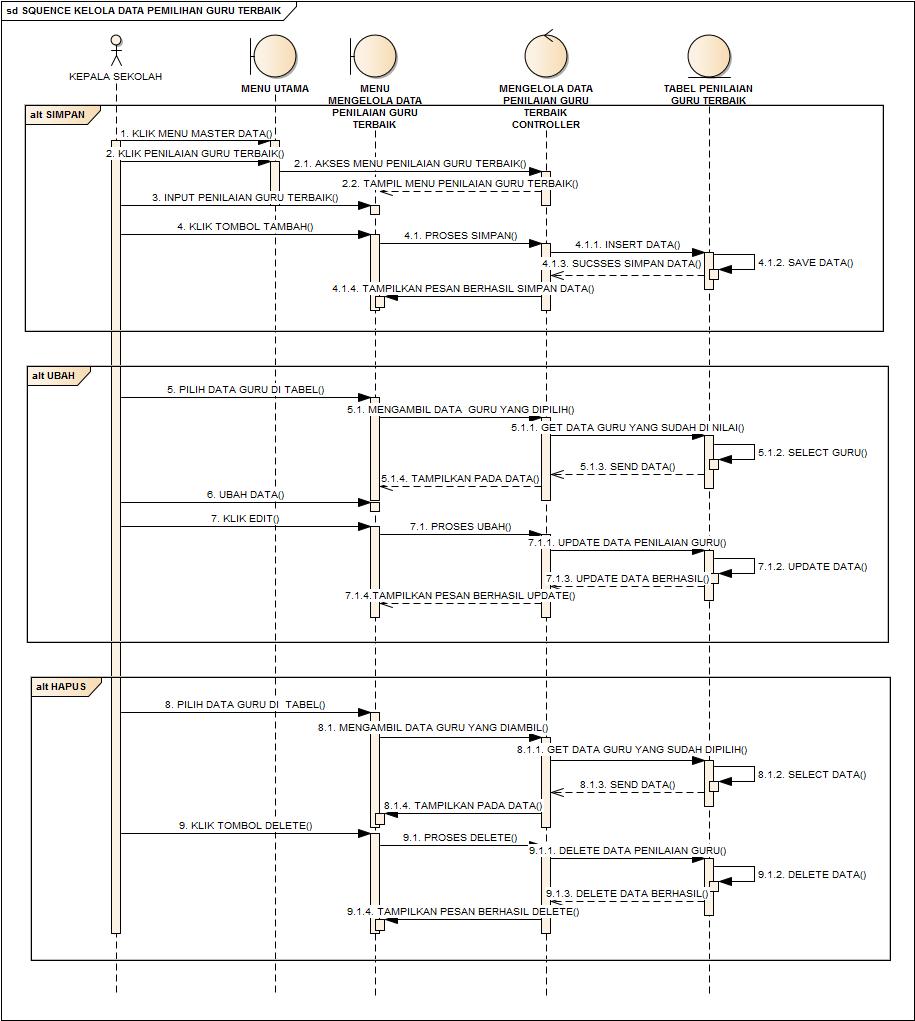
*Sequence diagram* Kelola Bobot Kriteria diatas merupakan kegiatan yang dilakukan pada Halaman Kelola Bobot Kriteria dimana Admin dapat menambah, mengubah, dan menghapus data kriteria.

1. *Squence Diagram* Kelola Data *User*

**Gambar 3.18*Squence Diagram* Kelola Data *User***

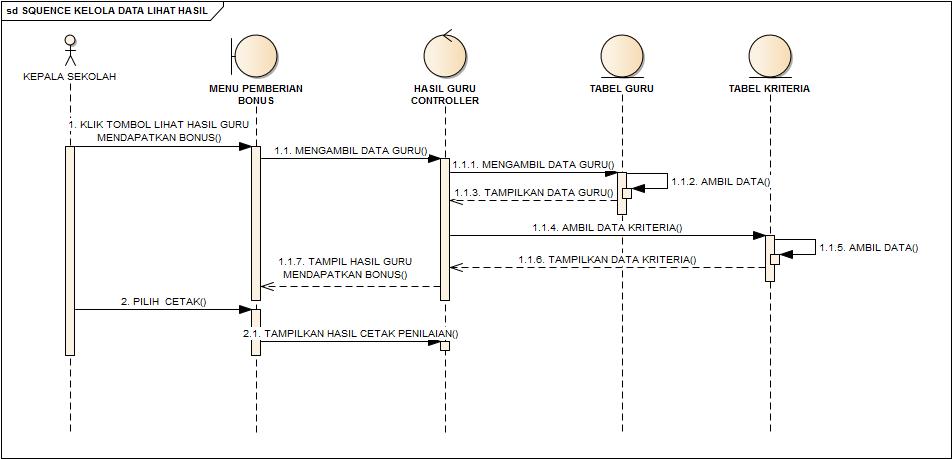
*Sequence diagram* Kelola Data User diatas merupakan kegiatan yang dilakukan pada Halaman Kelola Data User dimana Admin menginput data-data sebagai user.

1. *Squence Diagram* Kelola Data Pemilihan Guru Terbaik



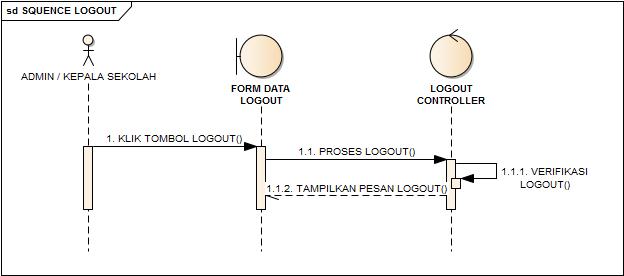
**Gambar 3.19*Squence Diagram* Kelola Data Pemilihan Guru Terbaik**

*Sequence diagram* Kelola data penilaian guru terbaik SPK diatas merupakan kegiatan yang dilakukan pada Halaman Pemberian bonus SPK dimana Kepala Sekolah dapat melakukan penilaian untuk menentukan guru terbaik.

1. *Squence Diagram* Melihat Hasil Guru

**Gambar 3.20*Squence Diagram* Melihat Hasil Guru**

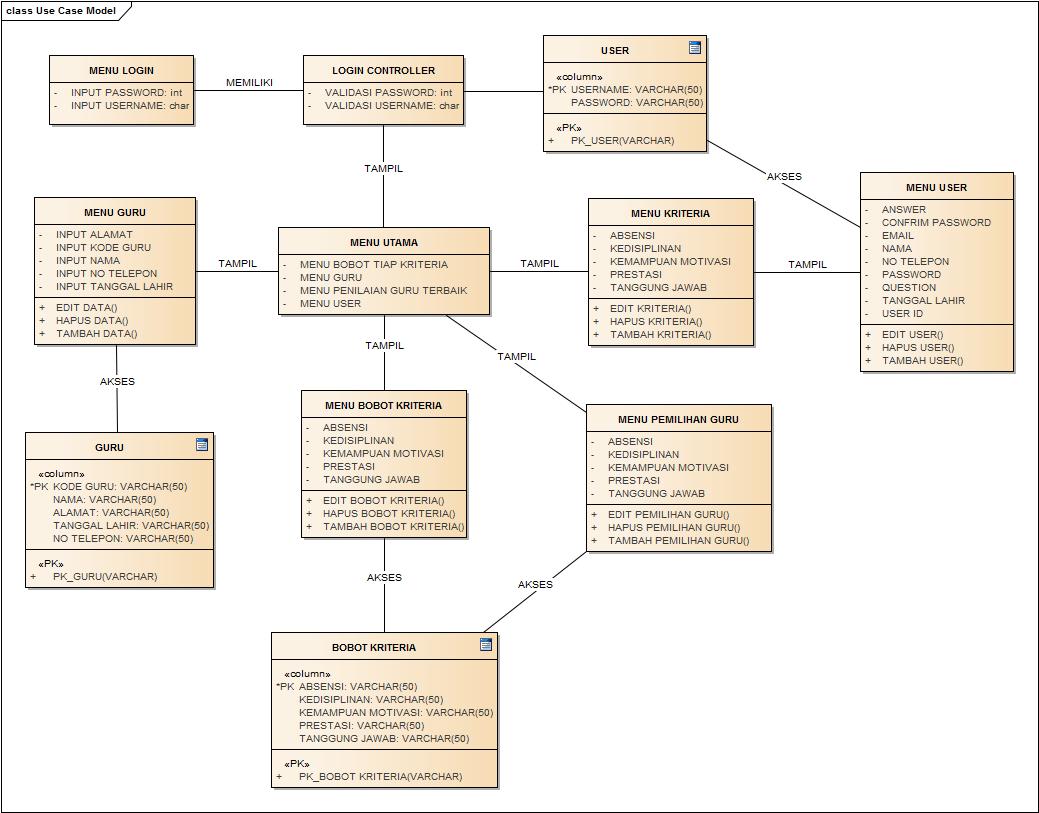
*Sequence diagram* Melihat Hasil Guru diatas merupakan kegiatan yang dilakukan pada Halaman Pemberian Bonus dimana Kepala Sekolah dapat melihat hasil penilaian yang dilakukan dengan metode SAW untuk menentukan Guru terbaik.

1. *Squence Diagram Logout*

**Gambar 3.21*Squence Diagram Logout***

*Sequence diagram* Logout diatas merupakan kegiatan Logout yang di lakukan oleh Admin / Kepala sekolah.

#### Class Diagram



**Gambar 3.22*Class Diagram***

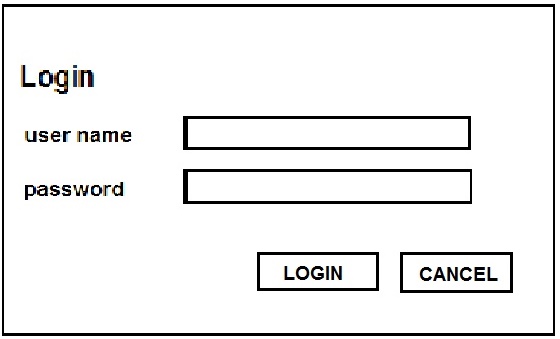
*Class diagram* Sistem Pemberian Bonus menurutkinerja guru diatas merupakan rancangan struktur *class,userinterface*, dan tabel yang akan digunakan.

#### Rancangan UserInterface

1. Lampiran Awal

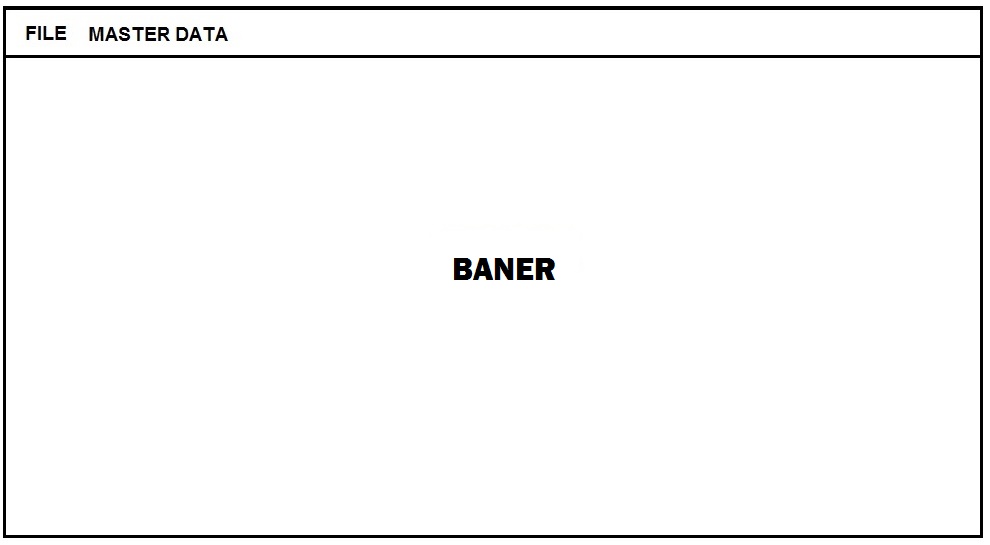
**Gambar 3.23. Rancangan *User Interface* Lampiran Awal**

Pada gambar 3.23 rancangan *User Interface* halaman beranda, komponennya adalah Lampiran Awal jendela aplikasi SAW yang akan di gunakan.

1. Halaman *Login*

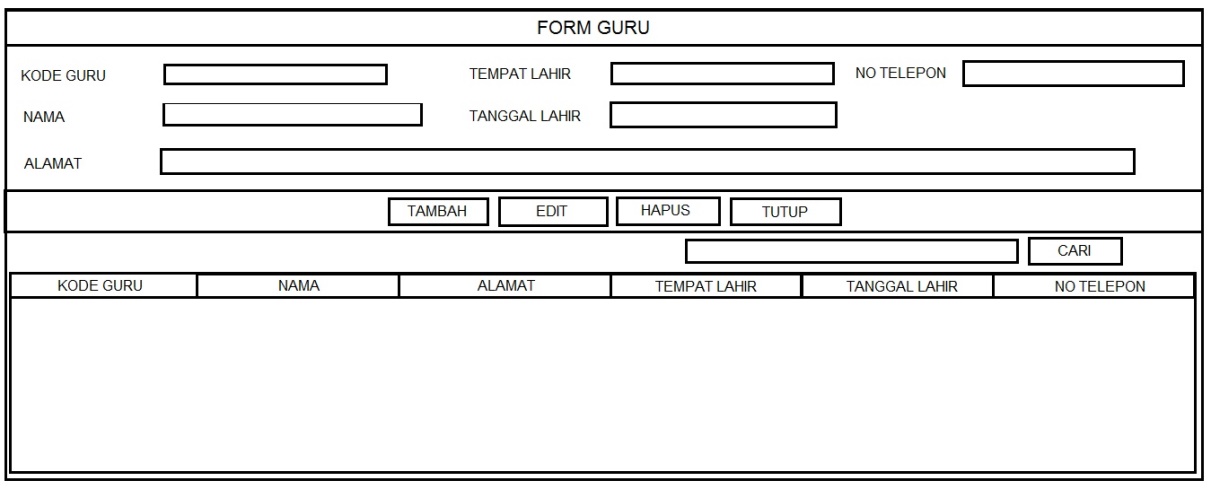
**Gambar 3.24 Rancangan *User Interface* Halaman *Login***

Pada gambar 3.24 rancangan *User Interface* halaman Login, komponennya adalah :

1. *Banner* untuk menampilkan tampilan dari judul halaman.
2. Input terdiri dari :
3. Input *username*.
4. Input *password*.
5. Tombol Login dan Cancel.
6. Halaman Utama

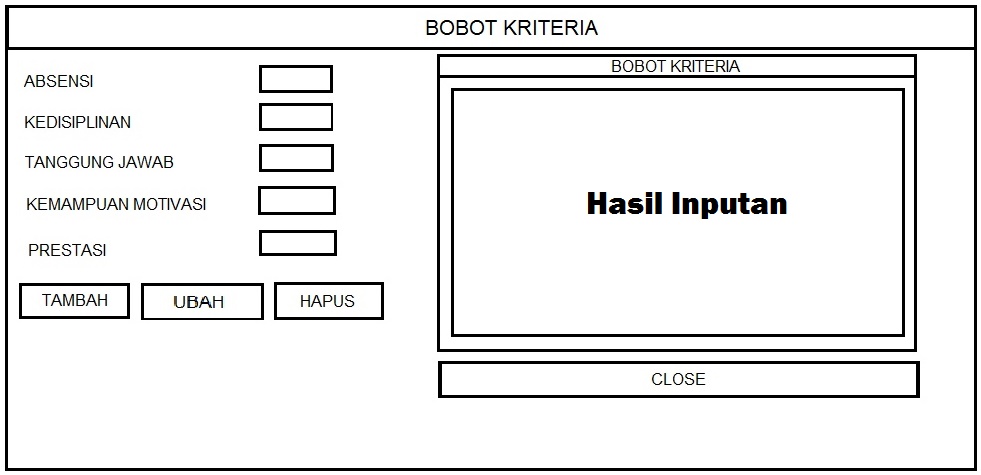
**Gambar 3.25. Rancangan *User Interface* Halaman Utama**

Pada gambar 3.25 rancangan *User Interface* halaman utama, komponennya adalah :

1. *File*untuk menampilkan tampilan *Logout* dan *Exit*.
2. Master Data terdiri dari :
3. Menu Kelola Data Guru.
4. Menu Kelola Bobot Kriteria.
5. Menu User
6. Menu Pemberian Bonus
7. Berupa tampilan Tulisan dari Judul Aplikasi dan Nama Sekolah.
8. Halaman Kelola Data Guru

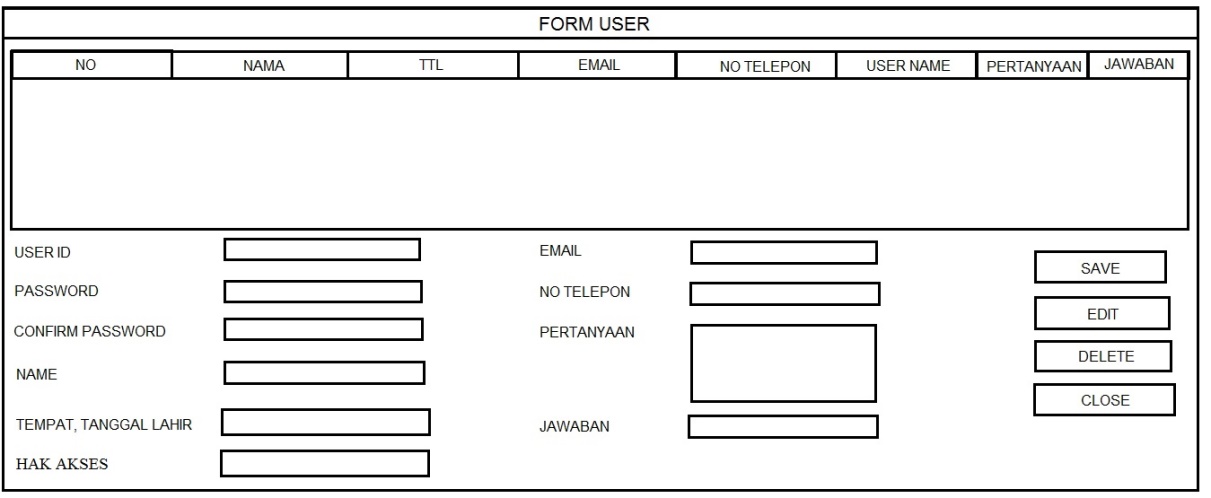
**Gambar 3.26 Halaman Rancangan *User Interface* Kelola data Guru**

Pada gambar 3.26 Rancangan *User Interface* halaman Kelola Data Guru, komponennya adalah :

1. Input terdiri dari :
2. Input Kode Guru.
3. Input Nama.
4. Input Alamat.
5. Input Tempat Lahir.
6. Input Tanggal Lahir.
7. Input No Telepon.
8. Input Cari
9. Tombol Terdiri :
10. Tombol Tambah.
11. Tombol Ubah.
12. Tombol Hapus.
13. Tombol Tutup.
14. Tombol Cari
15. Tabel data guru.
16. Halaman Kelola Bobot Kriteria

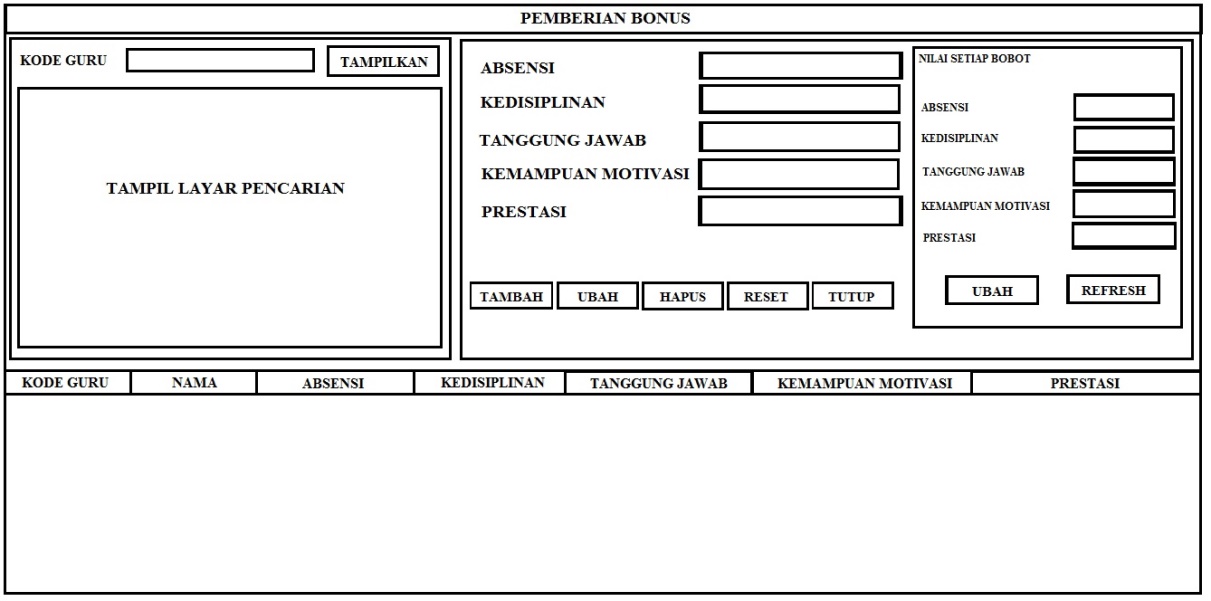
**Gambar 3.27. Rancangan Halaman *User Interface* Kelola Bobot Kriteria**

Pada gambar 3.27 Rancangan *User Interface* halaman Kelola Bobot Kriteria, komponennya adalah :

1. Input Nilai Kriteria terdiri dari :
2. Input Absensi.
3. Input Kedisiplinan.
4. Input Tanggung Jawab.
5. Input Kemampuan Motivasi.
6. Input Prestasi.
7. Tombol Terdiri :
8. Tombol Tambah.
9. Tombol Ubah.
10. Tombol Hapus.
11. Tombol Tutup.
12. Layar Hasil Inputan Nilai Rata-rata Bobot Kriteria.
13. Halaman Kelola Data *User*

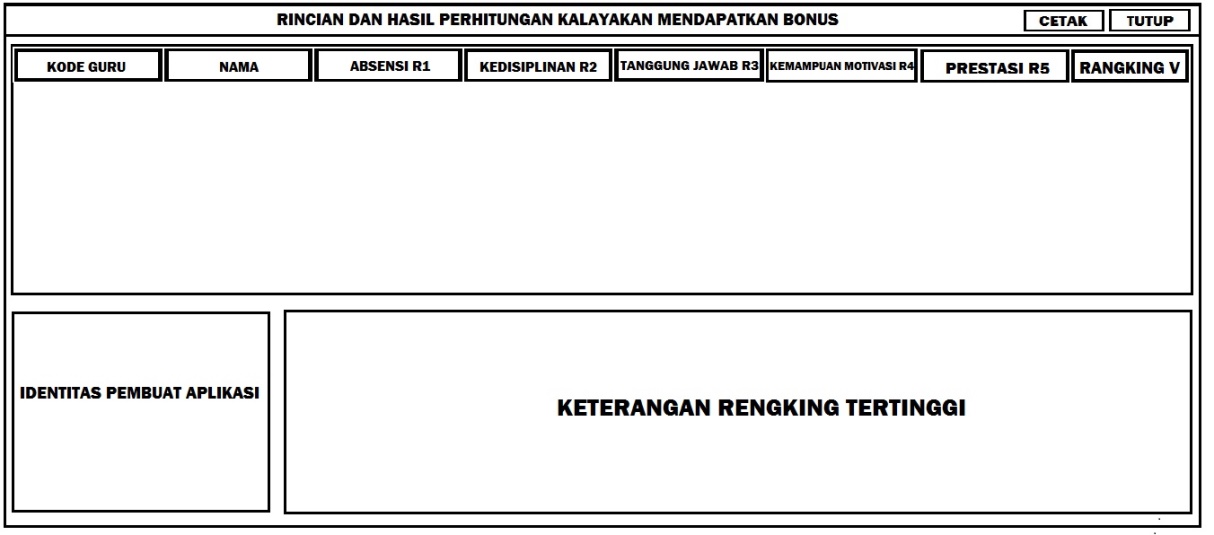
**Gambar 3.28. Rancangan Halaman *User Interface* Kelola Data *User***

Pada gambar 3.28 Rancangan *User Interface* halaman Kelola*User*, komponennya adalah :

1. Tabel Data User.
2. Input terdiri dari :
3. Input User Id.
4. Input Password.
5. Input Konfrim Password.
6. Input Nama.
7. Input Tempat Tanggal Lahir.
8. Input Hak Akses
9. Input Email
10. Input No Telepon
11. Input Pertanyaan
12. Input Jawaban
13. Tombol Terdiri :
14. Tombol Tambah.
15. Tombol Ubah.
16. Tombol Hapus.
17. Tombol Tutup.
18. Halaman Pemberian Bonus

**Gambar 3.29 Rancangan Halaman *User Interface* Penilaian Pemberian Bonus**

Pada gambar 3.29 Rancangan *User Interface* halaman Penilaian Pemberian Bonus, komponennya adalah :

1. Input terdiri dari :
2. Input Kode Guru.
3. Input Absensi.
4. Input Kedisiplinan.
5. Input Tanggung Jawab.
6. Input Kemampuan Motivasi.
7. Input Prestasi.
8. Tombol Terdiri :
9. Tombol Tambah.
10. Tombol Ubah.
11. Tombol Hapus.
12. Tombol Refresh
13. Tombol Tutup.
14. Tombol Tampilkan
15. Tabel data penilaian guru.
16. Halaman Lihat Hasil Mendapatkan Bonus

**Gambar 3.30 Rancangan Halaman *User Interface* Lihat Hasil Penilaian Mendapatkan Bonus**

Pada gambar 3.30 Rancangan *User Interface* halaman Kelola*User*, komponennya adalah :

1. Tabel Data Rincian Dan Hasil Perhitungan.
2. Identitas Pembuat
3. Keterangan Rengking Tertinggi
4. Tombol Terdiri :
5. Tombol Cetak.
6. Tombol Tutup.

# BAB IV

**IMPLEMENTASI DAN PEMBAHASAN**

## Implementasi

Implementasi adalah proses untuk memastikan terletaknya suatu kebijakan dan tercapainya kebijakan tersebut, dibawah ini merupakan beberapa bagian implementasi yang dibutuhkan sebagai berikut.

### Implementasi Aplikasi

Tahap implementasi merupakan tahap lanjutan dan perancangan sistem yang akan dilakukan jika sistem disetujui, termasuk program yang telah dibuat agar siap untuk dioperasikan secara optimal sesuai dengan kebutuhan.

### Spesifikasi Perangkat Lunak Yang Digunakan

Sistem operasi yang digunakan pada pembuatan aplikasi ini adalah *Windows.* Sedangkan bahasa pemrograman yang digunakan untuk membangun perangkat lunak ini adalah *Java* kemudahan *Java* dalam membangun suatu aplikasi menjadi salah satu faktor penulis menggunakannya. *Database* dibangun menggunakan *MYSQL*. Untuk implementasi dan pengujian dilakukan sepenuhnya pada perangkat keras PC(*Personal Computer*) / Leptop.

#### Spesifikasi Perangkat Keras Yang Digunakan

Spesifikasi perangkat keras terpenting untuk membuat Sistem Pendukung Keputusan Pemberian Bonus Berdasarkan Penilaian Kinerja Guru Dengan Metode S.A.W ini akan dijelaskan secara rinci seperti yang terlihat pada table dibawah ini.

**Tabel 4.1. Kebutuhan Perangkat Keras Komputer Yang Digunakan**

|  |  |  |
| --- | --- | --- |
| No | Perangkat Keras | Jenis Ukuran |
| 1 | Processor | Intel Pentium IV 2,4 Mhz |
| 2 | Memory | 512 GB |
| 3 | Harddisk | 80GB |
| 4 | Monitor | Standar |
| 5 | Keyboard + Mouse | Standar |

## Implementasi Antar Muka ( *Interface* )

Pengertian implementasi antar muka adalah salah satu layanan yang disediakan sistem operasi sebagai sarana interaksi anata pengguna dengan sistem operasi. Antar muka adalah komponen sistem operasi yang bersentuhan langsung dengan pengguna terhadap 2 (dua) jenis antarmuka, yaitu *Command Line Interface* ( CLI ) dan *Graphical User Interface* (GUI) berikut ini adalah implementasi setiap antarmuka.

### Implementasi Menu Utama

Form halaman utama merupakan sentral dari semua halaman, halaman ini berisi menu – menu yang dapat diakses oleh pengguna. Berikut ini implementasi struktur menu pada halaman utama sistem pendukung keputusan pemberian bonus menurut kinerja guru:

**Tabel 4.2. Implementasi Menu utama**

|  |  |
| --- | --- |
| Menu | Deskripsi |
| File | Merupakan Submenu untuk menampilkan logout atau exit. |
| Master Data | Merupakan submenu untu menampilkan semua data dan perhitungan. |

### Implementasi Menu Master Data

Implementasi halaman berdasarkan menu yang dibuat dalam sistem pendukung keputusan pemberian bonus menurut kinerja guru sebagai berikut.

**Tabel 4.3. Implementasi Menu Master Data**

|  |  |
| --- | --- |
| Menu | Deskripsi |
| Data Guru | Menu ini berisi tentang data para guru dan admin dapat menambahkan atau menghapus data. |
| Bobot Tiap Kriteria | Menu ini berisi tentang menentukan bobot sebuah kriteria dalam sebuah perhitungan Sistem Pendukung Keputusan. |
| User | Menu ini berisi tentang sebuat data user dan admin dapat menambahkan atau menghapus data. |
| Pemberian Bonus | Menu ini Berisi Hasil dan Perhitungan Sistem Pendukung Keputusan. |

### Implementasi Menu Data Guru

Implemetasi Halaman Berdasarkan Menu yang dibuat pada menu data guru hanya admin saja yang dapat menambahkan atau menghapus data guru.

**Tabel 4.4. Implementasi Menu Data guru**

|  |  |
| --- | --- |
| Menu | Deskripsi |
| Kode Guru | Menu ini dibuat untuk membuat atau menampilkan kode guru yang dilakukan oleh admin. |
| Nama | Menu ini dibuat untuk membuat atau menampilkan nama guru yang dilakukan oleh admin. |
| Alamat | Menu ini dibuat untuk membuat atau menampilkan alamat yang dilakukan oleh admin. |
| Tempat Lahir | Menu ini dibuat untuk membuat atau menampilkan Tempat Lahir yang dilakukan oleh admin. |
| Tanggal Lahir | Menu ini dibuat untuk membuat atau menampilkan Tanggal Lahir yang dilakukan oleh admin. |
| No Telepon | Menu ini dibuat untuk membuat atau menampilkan No telepon yang dilakukan oleh admin. |
| Tambah | Menu ini untuk menambahkan data guru yang dibuat oleh admin. |
| Ubah | Menu ini dibuat untuk mengubah data guru yang telah dibuat oleh admin. |
| Hapus | Menu ini dibuat untuk menghapus data guru yang telah dibuat oleh admin. |
| Tutup | Menu ini dibuat untuk menutup tampilan menu data guru. |
| Cari | Menu ini untuk mencari data guru yang Telah dibuat oleh admin. |

### Implementasi Menu Bobot Tiap Kriteria

Implemetasi Halaman Berdasarkan Menu yang dibuat pada menu Bobot tiap kriteria hanya admin saja yang dapat menentukan nilai bobot kriteria.

**Tabel 4.5 Implementasi Menu Bobot tiap Kriteria**

|  |  |
| --- | --- |
| Menu | Deskripsi |
| Absensi | Menu ini dibuat untuk menentukan nilai bobot dari kriteria absensi guru yang dilakukan oleh admin. |
| Kedisiplinan | Menu ini dibuat untuk menentukan nilai bobot dari kriteria kedisiplinan guru yang dilakukan oleh admin. |
| Tanggung Jawab | Menu ini dibuat untuk menentukan nilai bobot dari kriteria Tanggung jawab yang dilakukan oleh admin. |
| Kemampuan Motivasi | Menu ini dibuat untuk menentukan nilai bobot dari kriteria Kemampuan Motivasi yang dilakukan oleh admin. |
| Prestasi | Menu ini dibuat untuk menentukan nilai bobot dari kriteria Prestasi guru yang dilakukan oleh admin. |
| Tambah | Menu ini untuk menambahkan nilai bobot tiap kriteria yang dibuat oleh admin. |
| Ubah | Menu ini dibuat untuk mengubah nilai bobot tiap kriteria yang telah dibuat oleh admin. |
| Hapus | Menu ini dibuat untuk menghapus Penilaian Bobot tiap kriteria yang telah dibuat oleh admin. |
| Tutup | Menu ini dibuat untuk menutup tampilan menu Bobot tiap kriteria. |

### Implementasi Menu user

Implemetasi Halaman Berdasarkan Menu yang dibuat pada menu user hanya admin saja yang dapat menambah atau menghapus user.

**Tabel 4.6 Implementasi User**

|  |  |
| --- | --- |
| Menu | Deskripsi |
| User Id | Menu User id ini membuat atau menampilakn identitas user yang dilakukan oleh admin |
| Password | Menu ini dibuat untuk memberi kode keamanan kepada user yang dibuat yang di buat oleh admin |
| Konfrim Password | Menu ini dibuat untuk memastikan kode keamanan yang dibuat oleh admin benar. |
| Nama | Menu ini dibuat untuk mengetahui informasi siapa user yang sudah dibuat oleh admin. |
| Tempat, Tanggal Lahir | Menu ini dibuat untuk mengetahui informasi siapa user yang sudah dibuat oleh admin. |
| Hak ases | Menu ini dibuat untuk memberikan hak ases antara admin dengan user. |
| Email | Menu ini dibuat untuk mengetahui informasi siapa user yang sudah dibuat oleh admin. |
| No telepon | Menu ini dibuat untuk mengetahui informasi siapa user yang sudah dibuat oleh admin. |
| Pertanyaan | Menu ini dibuat untuk menambahkan perlindungan keamanan pada user dan admin. |
| Jawaban | Menu ini dibuat untuk menambahkan perlindungan keamanan pada user dan admin. |
| Simpan | Menu ini dibuat untuk menyimpan user atau admin baru yang telah dibuat. |
| Ubah | Menu ini dibuat untuk mengubah user dan admin yang telah dibuat. |
| Hapus | Menu ini dibuat untuk menghapus user yang telah dibuat oleh admin. |
| Keluar | Menu ini untuk keluar dari tampilan user. |

### Implementasi Menu Pemberian Bonus

Implemetasi Halaman Berdasarkan Menu yang dibuat pada menu pemberian bonus hanya user saja yang dapat menambah , menghapus dan memberikan penilaian.

**Tabel 4.7 Implementasi Menu Pemberian Bonus**

|  |  |
| --- | --- |
| Menu | Deskripsi |
| Kode guru | Menu kode guru dibuat untuk mencari data guru yang akan di beri penilaian oleh user |
| Absensi | Menu absensi pada implementasi pemberian bonus di buat untuk memberikan penilaian absensi guru. |
| Kedisiplinan | Menu absensi pada implementasi pemberian bonus di buat untuk memberikan penilaian kedisiplinan guru. |
| Tanggung jawab | Menu absensi pada implementasi pemberian bonus di buat untuk memberikan penilaian tanggung jawab guru dalam melakukan pengajaran. |
| Kemampuan motivasi | Menu absensi pada implementasi pemberian bonus di buat untuk memberikan penilaian kemampuan motivasi guru dalam memberikan motivasi ke siswa. |
| Prestasi | Menu absensi pada implementasi pemberian bonus di buat untuk memberikan penilaian prestasi guru dalam pengajaran. |
| Tambah | Menu ini dibuat untuk menambah kan penilaian yang telah dibuat oleh user. |
| Ubah | Menu ini dibuat untuk mengubah penilaian yang telah dibuat oleh user. |
| Hapus | Menu ini dibuat untuk menghapus data penilaian yang telah dibuat oleh user. |
| Refresh / reset | Menu ini dibuat untuk mereset kembali penilaian yang telah diketik oleh user. |
| Tutup | Menu ini dibuat untuk menutup tampilan pemberian bonus. |
| Tampilkan | Menu ini dibuat untuk memproses pencarian data guru yang ingin di cari. |
| Lihat hasil guru mendapatkan bonus | Menu ini dibuat untuk melihat hasil dari perhitungan dalam pemberian bonus. |

**Tabel 4.8 Implementasi Lihat hasil guru mendapatkan bonus**

|  |  |
| --- | --- |
| Menu | Deskripsi |
| Tabel | Menu table ini digunakan untuk melihat semua data penilaian. |
| Keterangan Hasil | Menu keterangan hasil di buat untuk mengetahui siapa yang akan mendapatkan bonus. |
| Cetak | Menu cetak ini dibuat untuk mencetak hasil penilaian. |
| Tutup | Menu ini dibuat untuk menutup tampilan Lihat hasil Guru mendapatkan bonus. |

## Implementasi Antar Muka

Implementasi rancangan antarmuka dengan menggunakan bahasa pemrograman Netbeans IDE 8.0, tahapan yang harus dilakukan untuk implementasi sistem pada computer, mulai dari tahap persiapan aplikasi dikomputer sampai dengan pengujian aplikasi sehingga siap digunakan beserta petunjuk umum penggunaan aplikasi yang digambarkan pada layar utama komputer. Berikut ini adalah implementasi antar muka dari aplikasi yang dibuat:

### Implementasi Tampilan Awal

Implementasi Tampilan Awal saat run berjalan dan hasil perancangan dapat di lihat seperti dibawah ini :

**Gambar 4.1 Implementasi Tampilan Awal**

Pada implementasi tampilan awal dimana proses pertama kali dalam membuka aplikasi ini.

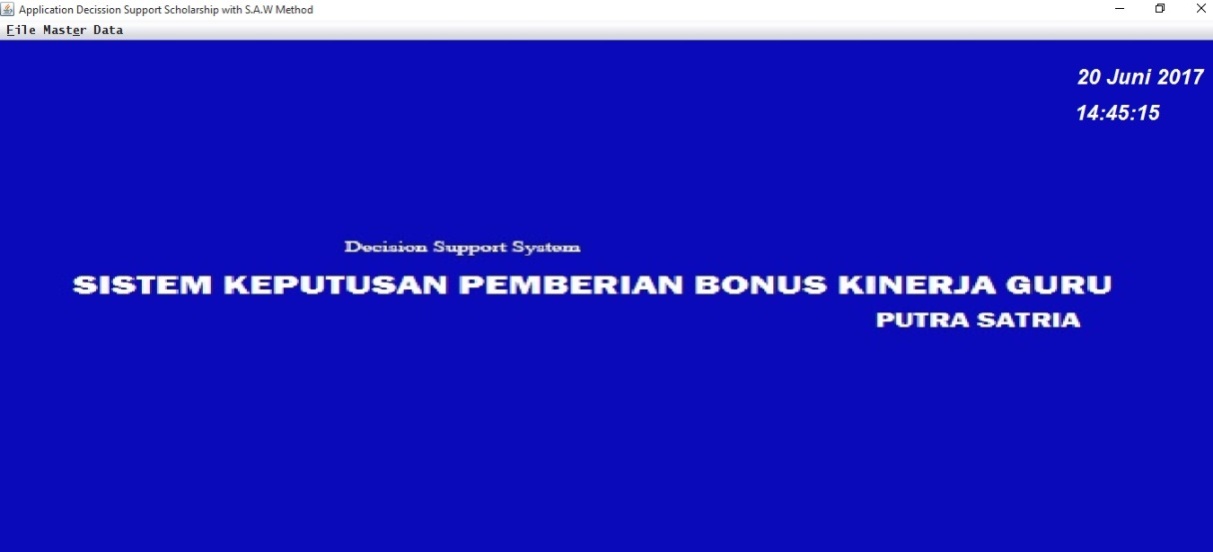
### Implementasi Tampilan Login

****Implementasi Tampilan Login dan hasil perancangan dapat di lihat seperti dibawah ini :

**Gambar 4.2 Implementasi Tampilan Login**

Pada Implementasi Tampilan form login ini menggunakan form login multi dimana admin dan user dapat melakukan login dalam satu form.

### Implementasi Tampilan Halaman Utama

****Implementasi Tampilan Halaman Utama dan hasil perancangan dapat di lihat seperti dibawah ini :

**Gambar 4.4 Implementasi Halaman Utama**

Pada gambar tersebut adalah menu utama pada halaman utama. Terdapat dua tombol yaitu tombol file dan master data.

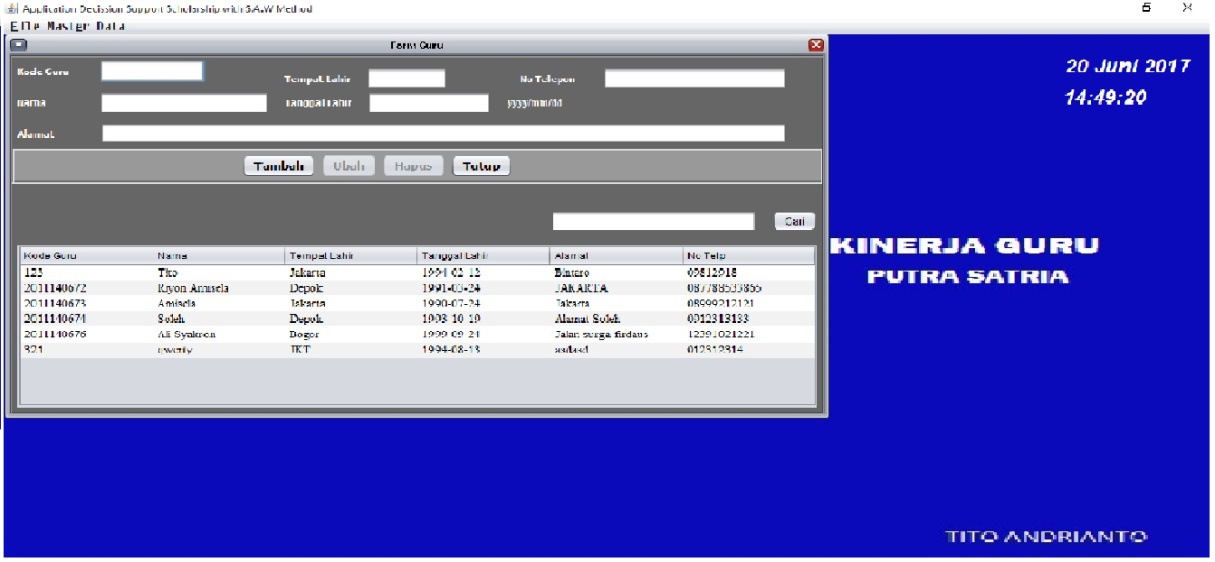
Tombol File :

* Deskripsi : Menjelaskan Tentang Aplikasi ini
* Logout : Logout keluar menu utama
* Exit : Exit Menutup aplikasi

Tombol Master Data :

* Data Guru : data guru untuk menyimpan ,ubah, hapus dan tambahkan guru yang akan dinilai
* Bobot Tiap Kriteria : pemberian nilai bobot dari kriteria yang telah ditentukan
* User : untuk menambahkan admin dan user baru
* Pemberian Bonus : penilaian atau perhitungan dengan menggunakan metode SAW untuk menentukan siapa yang akan mendapatkan bonus.

### Implementasi Tampilan Data Guru

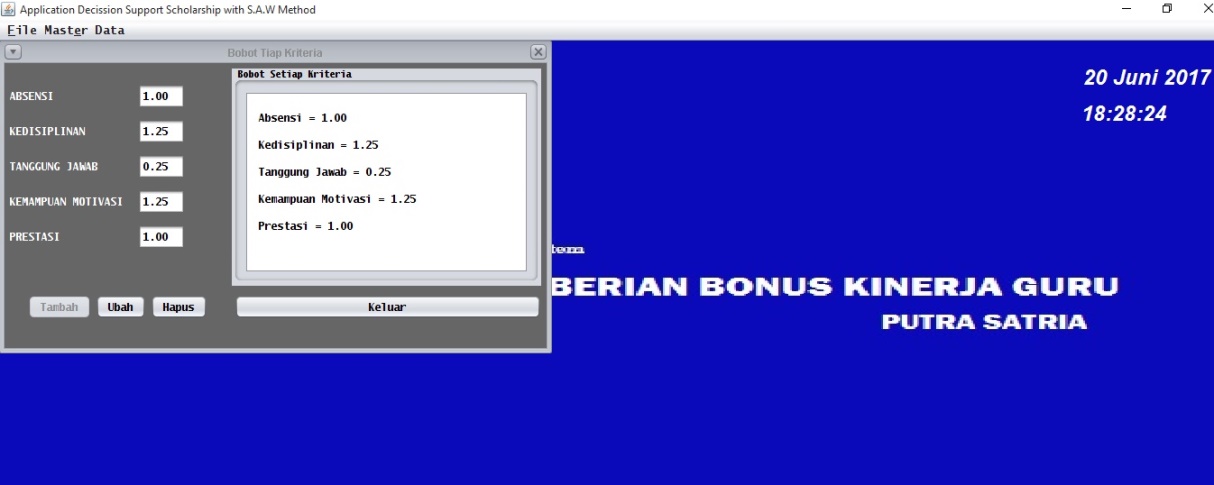
Implementasi Tampilan data guru dan hasil perancangan dapat di lihat seperti dibawah ini :

**Gambar 4.5 Implementasi Tampilan Data Guru**

Pada Implementasi Tampilan Data guru ini digunakan untuk menambahkan, mengubah, dan menghapus data guru yang akan dinilai.

### Implementasi Tampilan Bobot Tiap Kriteria

Implementasi Tampilan Bobot Tiap Kriteria dan hasil perancangan dapat di lihat seperti dibawah ini :

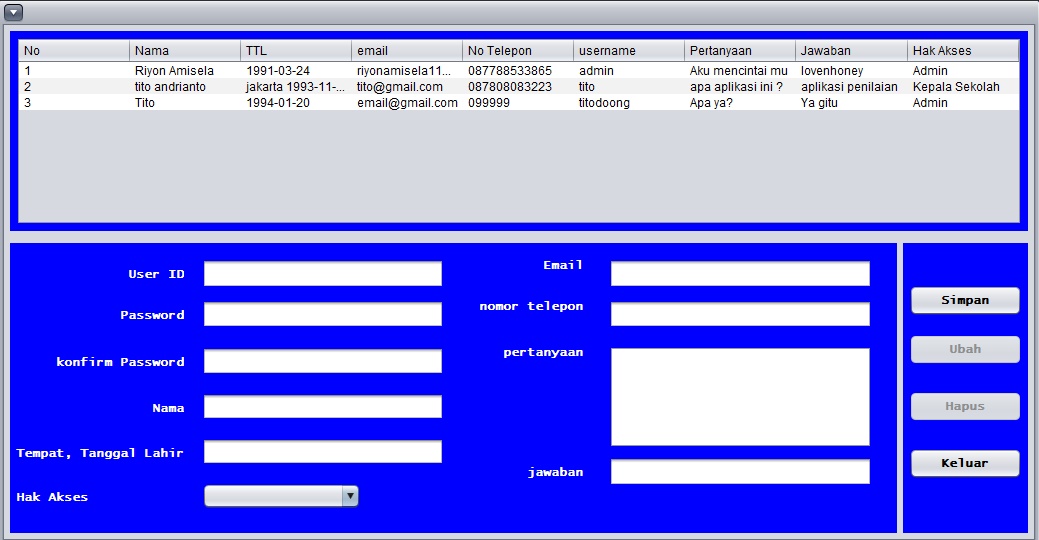
****

**Gambar 4.6 Implementasi Tampilan Bobot Tiap Kriteria**

Pada Implementasi Tampilan Bobot Tiap Kriteria ini digunakan untuk memberikan nilai bobot rata-rata pada setiap kriteria yang telah ditentukan.

### Implementasi Tampilan Data User

Implementasi Tampilan Data User dan hasil perancangan dapat di lihat seperti dibawah ini :

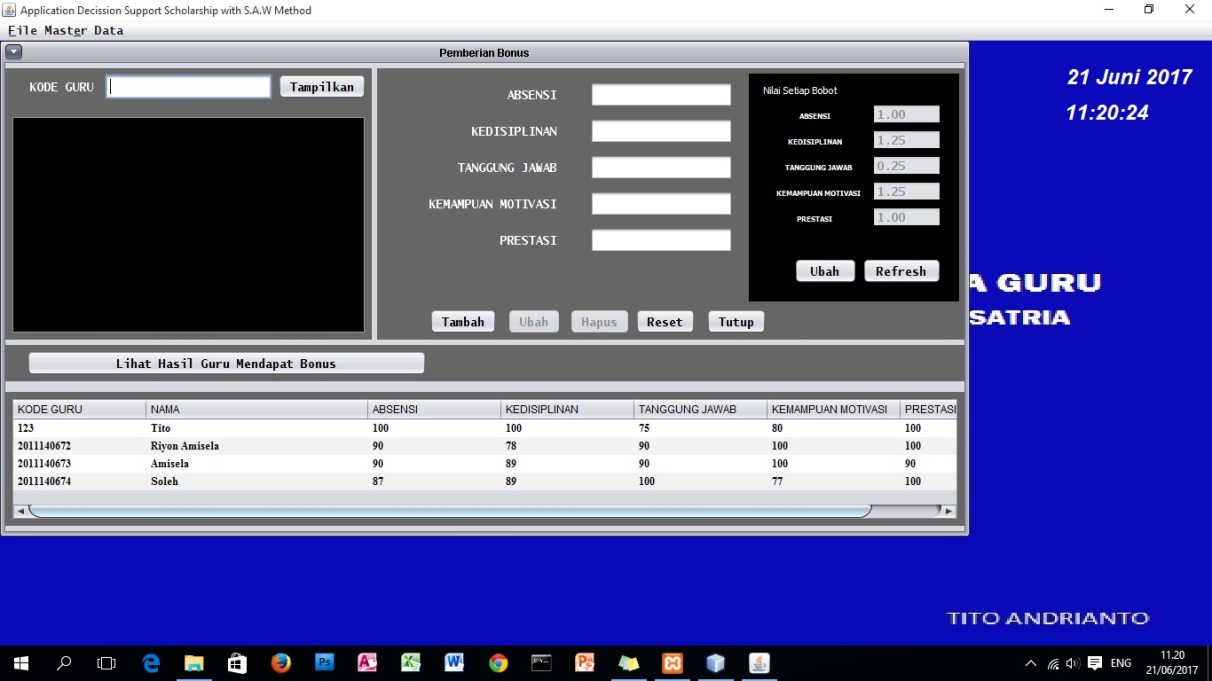


**Gambar 4.7 Implementasi Tampilan Data User**

Pada Implementasi Tampilan Data User ini digunakan untuk membuat, mengubah dan menghapus admin dan user baru.

### Implementasi Tampilan Pemberian Bonus

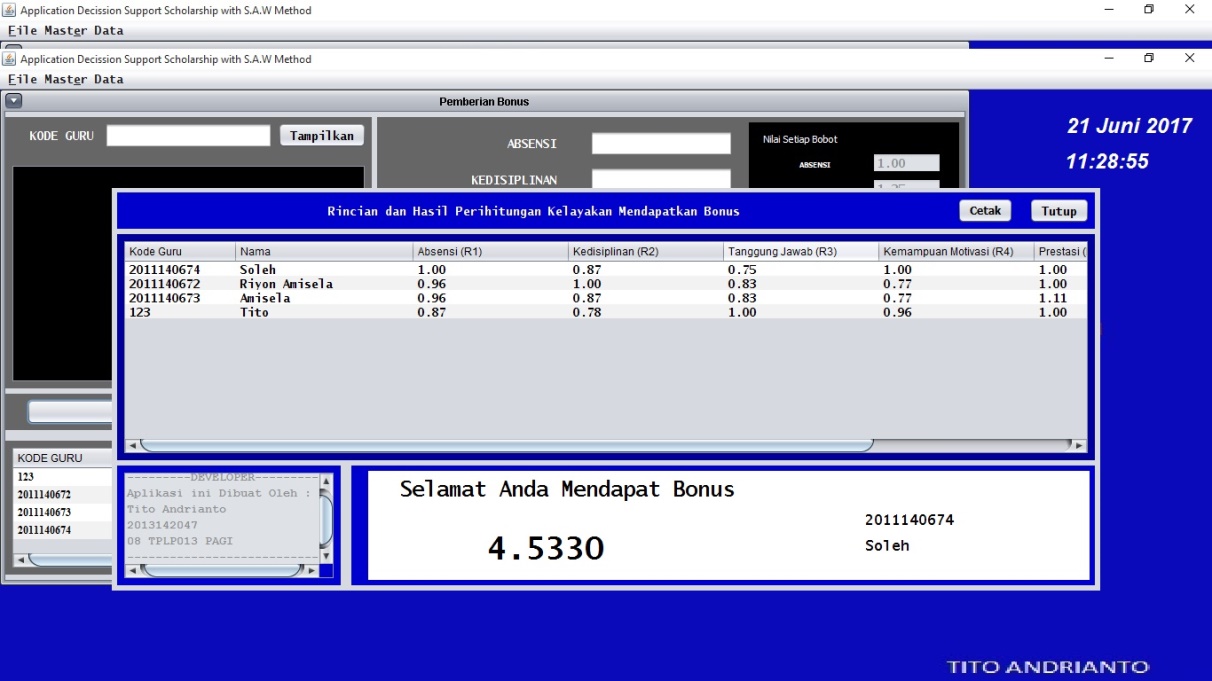
Implementasi Tampilan Pemberian Bonus dan hasil perancangan dapat di lihat seperti dibawah ini :

****

**Gambar 4.8 Implementasi Tampilan Pemberian Bonus**

Pada Implementasi Tampilan pemberian bonus ini untuk melakukan penilaian menggunakan metode saw siapa guru yang akan mendapatkan bonus.

### Implementasi Tampilan Lihat Guru Mendapatkan Bonus

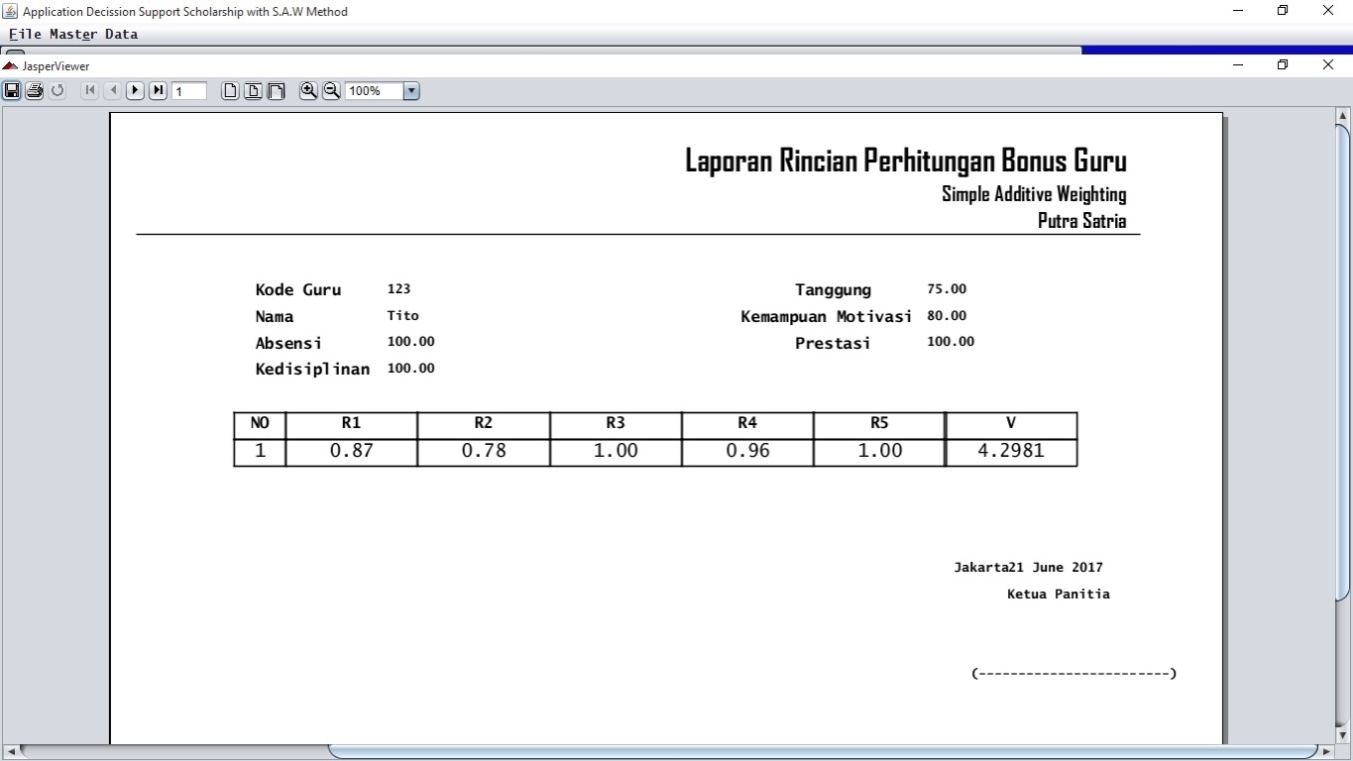
****Implementasi Tampilan Lihat Guru Mendapatkan Bonus dan hasil perancangan dapat di lihat seperti dibawah ini :

**Gambar 4.9 Implementasi Tampilan Lihat Guru Mendapatkan Bonus**

Pada Implementasi Tampilan Lihat Hasil Guru Mendapatkan Bonus ini adalah melihat hasil dari sebuah perhitung dengan metode saw yang dilakukan di menu pemberian bonus.

### Implementasi Tampilan Cetak Hasil

Implementasi Tampilan Lihat Cetak Hasil dan hasil perancangan dapat di lihat seperti dibawah ini :

****

**Gambar 4.10 Implementasi Tampilan Cetak Hasil**

Pada Implementasi Tampilan Cetak Hasil ini adalah sebuah output hasil akhir dari perhitung dengan metode saw yang dilakukan di menu pemberian bonus.

## Pengujian Black Box Dan White Box

### Black Box

Pengujian *black box* berfokus persyaratan fungsional perangkat lunak. Pengujian sistem pendukung keputusan pemberian bonus menurut kinerja guru ini menggunakan data uji berupa data *input* dari *administrator* pada sistem yang telah dibuat.

**Tabel 4.9 Rencana Pengujian**

|  |  |
| --- | --- |
| Item Uji | Jenis Penguji |
| Login | Black Box |
| Data Guru | Black Box |
| Bobot Tiap Kriteria | Black Box |
| Data User | Black Box |
| Pemberian Bonus | Black Box |

Berdasarkan rencana pengujian yang telah disusun, maka dapat dilakukan pengujian sebagai berikut :

1. **Pengujian *Login***

**Tabel 4.10 Pengujian *Login***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Khasus Dan Hasil Uji ( Data Benar )** | | | | | | |
| **Data Yang Dimasukkan** | **Yang Diharapkan** | | **Pengamatam** | | **Kesimpulan** | |
| *Input Username* dan *Password* sesuai pengguna katagori *Administrator* | Dapat *Login* dan masuk kehalaman utama *Admin* | | Menampilkan *Form* Menu Utama, seperti yang diharapkan | | * Diterima   **( )** Ditolak | |
| **Khasus Dan Hasil Uji ( Data Salah )** | | | | | | |
| *Input username* dan *Password* tidak sesuai pengguna katagori *Administrator* | | Tidak dapat *login* dan masuk kehalaman utama | | Menampilkan pesan *Error login* | | * Diterima   ( ) Ditolak |
| *Username* dan *Password* tidak dibiarkan kosong lalu tekan tombol *login* | | Tidak dapat *login* | | Menampilkan pesan *username* dan *Password* harus diisi | | ( ) Ditolak |

1. **Pengujian Data Guru**

**Tabel 4.11 Pengujian Data Guru**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kasus Dan Hasil Uji ( Data Benar )** | | | |
| **Data Yang Dimasukkan** | **Yang diharapkan** | **Pengamatan** | **Kesimpulan** |
| *Input* Kode guru, nama guru, alamat, tempat lahir, tanggal lahir, no telepon | Berhasil Menambahkan data guru | Menampilkan data sesuai yang di*Input.* | * Diterima   ( ) Ditolak |
| Klik *Action edit* pada Menu data guru | Menampilkan data pengguna sesuai dengan baris data pemain yang akan diubah | Sesuai dengan harpan | * Diterima   ( ) Ditolak |
| Klik *Action Delete* manajemen data guru | Jika data pengguna berhasil dihapus maka data guru akan hilang | Sesuai dengan harapan | * Diterima   ( ) Ditolak |
| **Kasus dan Hasil Uji ( Data Salah )** | | | |
| Kotak *Input* tidak diisi seperti yang telah ditentukan | Menampilkan Pesan Kesalahan | Sesuai dengan harapan | * Diterima   ( )Ditolak |

1. **Pengujian Pemberian Bobot Tiap Kriteria**

**Tabel 4.12 Pengujian Pemberian Bobot Tiap Kriteria**

|  |  |  |  |
| --- | --- | --- | --- |
| **Khasus Dan Hasil Uji ( Data Benar )** | | | |
| Data Yang DiMasukkan | Yang Diharapkan | Pengamatan | Kesimpulan |
| *Input* Absensi, Kedisiplinan, Tanggung Jawab, Kemampuan Motivasi, Prestasi | Berhasil Menambahkan Bobot Tiap Kriteria | Menampilkan Nilai Bobot sesuai yang telah ditentukan | * Diterima   ( ) Ditolak |
| Klik *Action edit* pada Menu Bobot Tiap Kriteria | Menampilkan Nilai Bobot yang telah diubah | Sesuai dengan harapan | * Diterima   ( ) Ditolak |
| Klik *Action Delete* Nilai Bobot Tiap Kriteria | Jika nilai rata - rata berhasil dihapus maka nilai bobot tiap kriteria akan hilang | Sesuai dengan harapan | * Diterima   ( ) Ditolak |
| **Khasus dan Hasil Uji ( Data Salah )** | | | |
| Kotak *Input* tidak diisi seperti yang telah ditentukan | Menampilkan pesan Kesalahan | Sesuai dengan harapan | * Diterima   ( ) Ditolak |

1. **Pengujian Data *User***

**Tabel 4.13 Pengujian Data User**

|  |  |  |  |
| --- | --- | --- | --- |
| **Khasus Dan Hasil Uji ( Data Benar )** | | | |
| **Data yang di masukkan** | **Data yang di harapkan** | **Pengamatan** | **Kesimpulan** |
| *Input User ID, Password, Confrim Password, Name,* Tempat tanggal lahir, Hak ases, Email, No telepen, Pertanyaan, Jawaban | Berhasil menambahkan data user | Menampilkan data sesuai yang di *input* | * Diterima   ( ) Ditolak |
| Klik *Action edit* pada Menu *User* | Menampilkan Data *user* sesuai dengan baris data *user* yang akan diubah | Sesuai dengan Harapan | * Diterima   ( ) Ditolak |
| Klik *Action Delete* manajemen data *user* | Jika data *User* berhasil dihapus maka data *user* akan Hilang | Sesuai dengan Harapan | * Diterima   ( ) Ditolak |
| **Khasus Dan Hasil Uji ( Data Salah )** | | | |
| Kotak *Input* tidak diisi seperti yang telah ditentukan | Menampilkan pesan Kesalahan | Sesuai dengan harapan | * Diterima   ( ) Ditolak |

1. **Pengujian Pemberian Bonus**

**Tabel 4.14 Pengujian Pemberian Bonus**

|  |  |  |  |
| --- | --- | --- | --- |
| **Khasus Dan Hasil Uji ( Data Benar )** | | | |
| **Data yang di masukkan** | **Data yang di harapkan** | **Pengamatan** | **Kesimpulan** |
| *Input* Kode Guru, Absensi, Kedisiplinan, Tanggung Jawab, Kemampuan Motivasi, Prestasi | Berhasil Menambahkan Hasil Penilaian | Menampilkan data guru yang sudah di nilai | * Diterima   ( ) Ditolak |
| Klik *Action* Tampilkanpada Menu pemberian bonus | Menampilkan data guru sesuai kode guru yang dicari | Sesuai dengan harapan | * Diterima   ( ) Ditolak |
| Klik *Action edit* pada Menu pemberian bonus | Menampilkan data guru yang sudah diberi penilaian yang telah diubah | Sesuai dengan harapan | * Diterima   ( )Ditolak |
| Klik *Action Delete* manajemen data pemberian bonus | Jika data Guru yang sudah dinilai berhasil dihapus maka dataGuru yang sudah di beri penilaianakan Hilang | Sesuai dengan Harapan | * Diterima   ( ) Ditolak |
| **Khasus Dan Hasil Uji ( Data Salah )** | | | |
| Kotak *Input* Penilaiantidak diisi seperti yang telah ditentukan | Menampilkan pesan Kesalahan | Sesuai dengan harapan | * Diterima   ( ) Ditolak |

### White Box

Uji coba *White Box* merupakan metode desain uji khasus yang menggunakan struktur control dari desain procedural untuk menghasilkan kasus – kasus uji. Pengujian *white box* di desain untuk mengungkapkan kesalahan pada persyaratan fungsional tanpa mengabaikan kerja internal dari suatu program. Seperti yang terlihat pada table *white boxtesting* dibawah ini :

1. Uji *White box* Sistem pemberian bonus

**Tabel 5.15 Pengujian *White Box* Sistem pemberian Bonus**

|  |  |  |
| --- | --- | --- |
| Pengujian | Test Case Benar | Test Case Salah |
| public void login()  {if(usernameTxt.getText().isEmpty() ||passTxt.getPassword().equals("")){  JOptionPane.showMessageDialog(null, "User dan Pass Tidak Boleh Kosong");  }  else{  Connection c = Koneksi.getKoneksi();  String sql = "select \* from tbuser where user\_id = ? and pass = ?;";  try{  PreparedStatement p = c.prepareStatement(sql); p.setString(1, usernameTxt.getText());  p.setString(2, String.valueOf(passTxt.getPassword()));  ResultSet r = p.executeQuery();  if(r.next()){  String hakAkses = r.getString("level");  MainForm mf = new MainForm();  mf.setVisible(true);  mf.setHakAkses(hakAkses);  this.dispose();  }  else{  JOptionPane.showMessageDialog(null, "User dan Password Tidak Sesuai");  }  }catch(SQLException ex){  JOptionPane.showMessageDialog(null, "Gagal Login Keterangan "+ex);  }  }  } | Sistem Dapat Memproses Login | Sistem Tidak Ada Proses Login |
| private void btnTambahActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  String kodeGuru = kodeGuruTxt.getText();  String nama = namaTxt.getText();  String telp = telpTxt.getText();  String tgllahir = tglTxt.getText();  String tempat = tempatTxt.getText();  String alamat = alamatTxt.getText();  String notelp = telpTxt.getText();  if((kodeGuruTxt.getText().isEmpty())||(namaTxt.getText().isEmpty())  ||(telpTxt.getText().isEmpty())||(tglTxt.getText().isEmpty())||  (tempatTxt.getText().isEmpty())||(alamatTxt.getText()  .isEmpty())||(telpTxt.getText().isEmpty())){  JOptionPane.showMessageDialog(null, "Data Tidak Dapat Disimpan, Ada data yang kosong");  }  else{  try{  Connection c = Koneksi.getKoneksi();  String sql = "insert into guru values (?, ?, ?, ?, ?, ?);";  PreparedStatement p = c.prepareStatement(sql);  p.setString(1, kodeGuru);  p.setString(2, nama);  p.setString(3, tempat);  p.setString(4, tgllahir);  p.setString(5, alamat);  p.setString(6, notelp);  p.executeUpdate();  p.close();  JOptionPane.showMessageDialog(null, "Data Tersimpan ");  reset();  }catch(SQLException e){  JOptionPane.showMessageDialog(null, "Data Tidak Dapat Disimpan "+e);  }  finally{  loaddata();  }  }  kodeGuruTxt.setEnabled(true);  btnEdit.setEnabled(false);  btnHapus.setEnabled(false);  btnTambah.setEnabled(true);  } | Sistem Berhasil Menambah Data Guru | Sistem Tidak Berhasil Menambahkan Data Guru |
| private void btnEditActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  int i = tableGuru.getSelectedRow();  if(i == -1){  return;  }  String kodeGuru = (String) tableModel.getValueAt(i, 0);  String nama = namaTxt.getText();  String telp = telpTxt.getText();  String tgllahir = tglTxt.getText();  String tempat = tempatTxt.getText();  String alamat = alamatTxt.getText();  String notelp = telpTxt.getText();  try{  Connection c = Koneksi.getKoneksi();  String sql = "update guru set nama = ?, tempat\_lahir = ?, tanggal\_lahir = ?, alamat = ?, no\_telp = ? where kode\_guru = ?;";  PreparedStatement p = c.prepareStatement(sql);  p.setString(1, nama);  p.setString(2, tempat);  p.setString(3, tgllahir);  p.setString(4, alamat);  p.setString(5, notelp);  p.setString(6, kodeGuru);  p.executeUpdate();  p.close();  JOptionPane.showMessageDialog(null, "Data Berhasil Di Ubah ");  reset();  }catch(SQLException e){  JOptionPane.showMessageDialog(null, "Data Tidak Dapat Di Ubah "+e);  }  finally{  loaddata();  }  kodeGuruTxt.setEnabled(true);  btnEdit.setEnabled(false);  btnHapus.setEnabled(false);  btnTambah.setEnabled(true);  } | Sistem Berhasi Mengubah Data Guru | Sistem Gagal Mengubah Data Guru |
| private void btnHapusActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  int i = tableGuru.getSelectedRow();  if(i == -1){  return;  }  String kodeGuru = (String) tableModel.getValueAt(i, 0);  try{  Connection c = Koneksi.getKoneksi();  String sql = "delete from guru where kode\_guru = ?";  PreparedStatement p = c.prepareStatement(sql);  p.setString (1, kodeGuru);  p.executeUpdate();  JOptionPane.showMessageDialog(null, "Data Berhasil di Hapus");  p.close();  reset();  }catch(SQLException ex){  JOptionPane.showMessageDialog(null, "Data Gagal di Hapus \n"+ex);  }  finally{  loaddata();  }  kodeGuruTxt.setEnabled(true);  btnEdit.setEnabled(false);  btnHapus.setEnabled(false);  btnTambah.setEnabled(true);  } | Sistem Berhasil Menghapus Data Guru | Sistem Gagal Menghapus Data Guru |
| public void caridata(){  //hapusseluruhdata  tableModel.getDataVector().removeAllElements();  //memeberi tahu tablekososng  tableModel.fireTableDataChanged();  if (cariTxt.getText().isEmpty()){  loaddata();  }  else{  try{  String cari = cariTxt.getText();  Connection c = KoneksiDB.Koneksi.getKoneksi();  Statement s = c.createStatement();  String sql = "select \* from guru where kode\_guru like '"+cari+"'";  ResultSet r = s.executeQuery(sql);  while(r.next()){  Object[] o = new Object[6];  o[0] = r.getString("kode\_guru");  o[1] = r.getString("nama");  o[2] = r.getString("tempat\_lahir");  o[3] = r.getString("tanggal\_lahir");  o[4] = r.getString("alamat");  o[5] = r.getString("no\_telp");  tableModel.addRow(o);  }  r.close();  s.close();  }catch(SQLException e){  JOptionPane.showMessageDialog(null, "Tidak Dapat Menampilkan Database"+e);  }  }  reset();  } | Sistem Berhasil Mencari Data Guru Yang Telah Dibuat | Sistem Gagal Mencari Data Guru |
| private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  // TODO add your handling code here:  try{  String NamaFile = "src/Report/RincianTiapGuru.jasper";  Connection c = Koneksi.getKoneksi();  HashMap hash = new HashMap();  File file = new File(NamaFile);  JasperReport jasperReport = (JasperReport) JRLoader.loadObject(file.getPath());  JasperPrint jasperPrint = JasperFillManager.fillReport(jasperReport, hash,c);  JasperViewer.viewReport(jasperPrint);  }catch(Exception e){  JOptionPane.showMessageDialog(null,"Kesalahan "+e);  }  } | Sistem Berhasil Mencetak Hasil | Sistem Gagal Mencetak Hasil Penilaian |

# BAB V

**KESIMPULAN DAN SARAN**

## KESIMPULAN

Berdasarkan hasil perancangan, implementasi dan pengujian sistem pendukung keputusan pemberian bonus menurut kinerja guru menggunakan metode *Simple additive wighting* ( S.A.W ), maka diambil kesimpulan sebagai berikut :

1. Penerapan metode S.A.W pada sistem pendukung keputusan pemberian bonus menurut kinerja guru terbukti lebih efektif.
2. Sistem pendukung keputusan terbukti mampu mengatasi masalah secara objektif dalam pemberian bonus menurut kinerja guru di SMK PUTRA SATRIA.

## SARAN

Dari hasil analisa dan perancangan sistem ini peneliti manyadari bahwa masih banyak kekurangan dan kesalahan dalam pengembangan sistem ini, dan untuk meningkatkan kemampuan dan fungsi dari sistem ini ada beberapa saran yang diberikan untuk pengembangan lebih lanjut :

1. Program yang dibuat ini masih dapat dikembangkan agar menjadi sistem yang lebih lengkap berdasarkan dengan kepentingan yang lebih luas. Hal yang dapat dikembangkan antara lain adalah penyeleksian jumlah guru tidak ada batasnya.
2. Diharapkan sistem ini dapat dikembangkan lagi pada aplikasi berbasis *mobile* atau pun lainnya.
3. Diharapkan sistem ini dikembangkan dengan pengembangan dengan metode seperti WG atau lainnya.

# DAFTAR PUSTAKA

, Mulawarman;. (2011). Pengertian UML. *Memahami Penggunaan UML*, 2.

Fauziah, Wakhidatul;. (2015). Pengertian SAW ( Simple Additive weighting ). *PENERAPAN METODE SIMPLE ADDITIVE WEIGHTING DALAM*, 37.

Hermawan, Asep;. (2015). Mengenal Java. *Pengenalan Java*, 1.

Kusnayadi;. (2011). Definisi Basis data. *Definisi Basis data*, 1.

Shalahudin, Muhammad; Rosa, Ariani;. (2011). Pengertian WhiteBox dan BlackBox. In Modula, *Rekayasa Perangkat Lunak* (p. 133). Bandung: Modula.

Taufik, Rohmat. (2012). Teori Sistem Pendukung Keputusan. *RANCANG BANGUN SISTEM PENDUKUNG KEPUTUSAN*, 2.

Zulkilfi, Sariffudin;. (2016). Decision Suport system pemberian bonus. *Jurnal TAM(Technologi Acceptance Model )*, 2.