



**SISTEM PENILAIAN PRESTASI KERJA GURU SEBAGAI LUARAN  
PEMILIHAN CALON KEPALA SEKOLAH MENGGUNAKAN METODE  
*PROFILE MATCHING - AHP*  
(STUDI KASUS : SMA NEGERI GRUJUGAN BONDOWOSO)**

**SKRIPSI**

Oleh

**Tania Rosifrianti**

**NIM 122410101050**

**PROGRAM STUDI SISTEM INFORMASI**

**UNIVERSITAS JEMBER**

**2017**



**SISTEM PENILAIAN PRESTASI KERJA GURU SEBAGAI LUARAN  
PEMILIHAN CALON KEPALA SEKOLAH MENGGUNAKAN METODE  
*PROFILE MATCHING - AHP*  
(STUDI KASUS : SMA NEGERI GRUJUGAN BONDOWOSO)**

**SKRIPSI**

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat  
untuk menyelesaikan Program Studi Sistem Informasi (S1)  
dan mencapai gelar Sarjana Komputer

Oleh

**Tania Rosifrianti**  
**NIM 122410101050**

**PROGRAM STUDI SISTEM INFORMASI  
UNIVERSITAS JEMBER**

**2017**

## PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT yang senantiasa memberikan rahmat dan hidayah-Nya untuk mempermudah dan melancarkan dalam mengerjakan skripsi;
2. Ibunda Yayuk Sulistyowati dan Ayahanda Harun Rosyidi tersayang yang selalu memberikan do'a, kasih sayang, motivasi, dan motivasi yang luar biasa;
3. Adik Kandung Agung Muliawan dan Tedi Setiawan Cahyadi yang selalu memberi dukungan;
4. Seluruh keluarga besar yang selalu menyayangiku;
5. Bapak Kepala Sekolah SMA Negeri Grugugan Drs. Darsono yang mau dan dengan sabar memberikan sumber ilmu dalam proses pengumpulan data skripsi
6. Almamater Program Studi Sistem Informasi Universitas Jember
7. Sahabatku bersama doa dan dukungannya;
8. Guru-guruku sejak taman kanak-kanak hingga perguruan tinggi yang telah memberikan ilmu dan bimbingan;

## MOTTO

“Tidak semua yang kita miliki harus diumumkan, tidak semua yang kita lakukan diberitahukan. Jadilah seperti gunung es di dalam lautan, yang terlihat hanya pucuk kecilnya saja, sedangkan di bawah, di dalam laut, tersimpan erat bagian raksasanya.<sup>1</sup>”



---

<sup>1</sup> Darwis Tere Liye. 2015. *#AboutLove*. Jakarta : Gramedia Pustaka Utama.

**PERNYATAAN**

Saya yang bertanda tangan di bawah ini:

Nama : Tania Rosifrianti

NIM : 122410101050

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Penilaian Prestasi Kerja Guru Sebagai Luaran Pemilihan Calon Kepala Sekolah Menggunakan Metode *Profile Matching* - AHP (Studi Kasus : SMA Negeri Grujugan Bondowoso)”, adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 14 Juli 2017

Yang menyatakan,

Tania Rosifrianti

NIM 122410101050

**SKRIPSI**

**SISTEM PENILAIAN PRESTASI KERJA GURU SEBAGAI LUARAN  
PEMILIHAN CALON KEPALA SEKOLAH MENGGUNAKAN METODE  
*PROFILE MATCHING* - AHP  
(STUDI KASUS : SMA NEGERI GRUJUGAN BONDOWOSO)**

Oleh

**Tania Rosifrianti**

**NIM 122410101050**

Pembimbing :

Dosen Pembimbing Utama : Anang Andrianto., S.T., M.T.  
Dosen Pembimbing Pendamping : Yanuar Nurdiansyah.,S.T.,M.Cs.

**PENGESAHAN PEMBIMBING**

Skripsi berjudul “Sistem Penilaian Prestasi Kerja Guru Sebagai Luaran Pemilihan Calon Kepala Sekolah Menggunakan Metode *Profile Matching* - AHP (Studi Kasus : SMA Negeri Grujugan Bondowoso)” telah diuji dan disahkan pada:

Hari, tanggal : Jumat, 14 Juli 2017

Tempat : Program Studi Sistem Informasi Universitas Jember.

Disetujui oleh :

Pembimbing I,

Pembimbing II,

Anang Andrianto.,S.T.,M.T  
NIP 196909281993021001

Yanuar Nurdiansyah.,S.T.,M.Cs.  
NIP 198201012010121004

**PENGESAHAN PENGUJI**

Skripsi berjudul “Sistem Penilaian Prestasi Kerja Guru Sebagai Luaran Pemilihan Calon Kepala Sekolah Menggunakan Metode Profile Matching - AHP (Studi Kasus : SMA Negeri Grujugan Bondowoso)” telah diuji dan disahkan pada:

hari, tanggal : Jumat, 14 Juli 2017

tempat : Program Studi Sistem Informasi Universitas Jember.

Tim penguji:

Penguji I,

Penguji II,

Dr. Saiful Bukhori, ST., M.Kom  
NIP 196811131994121001

Fajrin Nurman Arifin, ST., M.Eng  
NIP 198411012015042001

Mengesahkan  
Ketua Program Studi,

Prof. Drs. Slamim, M.Comp.Sc., Ph.D

NIP 196704201992011001



## RINGKASAN

**Sistem Penilaian Prestasi Kerja Guru Sebagai Luaran Pemilihan Calon Kepala Sekolah Menggunakan Metode *Profile Matching* - AHP (Studi Kasus : SMA Negeri Grujugan Bondowoso);** Tania Rosifrianti, 122410101050; 2017: 301 halaman; Progam Studi Sistem Informasi Universitas Jember.

Guru merupakan jenis pekerjaan ini tidak bisa di lakukan oleh sembarang orang diluar bidang kependidikan. Kemampuan guru perlu ditingkatkan dan dikembangkan secara terus menerus menurut jabatan fungsional guru. Diperlukan penilaian prestasi kerja guru untuk menilai hasil kerja pada masing-masing guru sehingga mewujudkan tenaga pendidik yang profesional. Hasil penilaian prestasi kerja guru dapat dimanfaatkan untuk menyusun profil kinerja guru sebagai input dalam penyusunan program Pengembangan Keprofesian Berkelanjutan.

Kepala sekolah merupakan pimpinan tertinggi didalam suatu sekolah. Jabatan kepala sekolah merupakan pengembangan keprofesian kelanjutan karier bagi seorang guru. Penilaian prestasi kerja guru merupakan cara untuk mengetahui seberapa pantas seorang guru agar memiliki hak untuk melanjutkan ke jenjang karier lebih tinggi yaitu sebagai kepala sekolah.

Tugas seleksi administrasi dalam pencalonan diri sebagai kepala sekolah dikerjakan pada tingkat sekolah. Kesulitan dan kesalahan lebih rentang terjadi karena banyaknya data kriteria dengan bobot berbeda-beda yang dihasilkan dari pelaporan penilaian selama menjadi guru. Hal itulah yang sama terjadi pada SMA Negeri Grujugan Bondowoso.

Sistem Penilaian Prestasi Kerja Guru Sebagai Luaran Pemilihan Calon Kepala Sekolah ini menggunakan metode *Profile Matching* – AHP. Pemilihan metode tersebut dilakukan karena dalam merekap dan menghitung hasil penilaian calon peserta nama guru yang boleh mencalonkan diri sebagai calon kepala sekolah terdiri dari berbagai

macam kriteria penilaian dengan masing-masing bobot perbandingan kriteria yang berbeda, dari situ dibutuhkan metode AHP, sedangkan untuk membandingkan jarak selisih nilai ideal capaian prestasi kinerja guru dengan masing-masing data nilai yang akan di hitung yaitu hasil kerja prestasi guru menggunakan metode *Profile Matching*.



## PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul Sistem Penilaian Prestasi Kerja Guru Sebagai Luaran Pemilihan Calon Kepala Sekolah Menggunakan Metode *Profile Matching* - AHP (Studi Kasus : SMA Negeri Grujugan Bondowoso) Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penyusunan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Oleh karena itu penulis menyampaikan terima kasih kepada:

1. Allah SWT, yang telah memberikan kesempatan untuk hidup menyelesaikan skripsi ini, dan kekuatan yang diberikan sehingga skripsi ini selesai;
2. Prof. Drs. Slamir, M.Comp.Sc.,Ph.D, selaku Ketua Program Studi Sistem Informasi Universitas Jember;
3. Anang Andrianto, S.T., M.T., selaku Dosen Pembimbing Utama dan Yanuar Nurdiansyah, S.T.,M.Cs., selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
4. Drs. Antonius Cahya P, M.App., Sc., Ph.D, selaku Dosen Pembimbing Akademik yang telah memberikan bimbingan selama penulis menjadi mahasiswa;
5. Seluruh Bapak dan Ibu dosen beserta staf karyawan di program studi sistem informasi;
6. Ibunda Yayuk Sulistyowati dan Ayahanda Harun Rosyidi tersayang yang selalu memberikan dukungan do'a, kasih sayang, motivasi yang luar biasa;
7. Adik Kandung Agung Muliawan dan Tedi Setiawan Cahyadi tercinta yang selalu memberi dukungan;
8. Bapak Kepala Sekolah SMA Negeri Grujugan Drs. Darsono yang mau dan dengan sabar memberikan sumber ilmu dalam proses pengumpulan data skripsi
9. Seluruh keluarga besar tersayang;

10. Teman-teman Formation (SI-2012) tercinta dan seluruh warga Program Studi Sistem Informasi yang telah menjadi keluarga bagi penulis selama menempuh pendidikan di Program Studi Sistem Informasi;
11. Vela Vidya Wati, Roni Kurniawan, Suci Nur Ramadhani, Fitriani, Ghausilia, Seftya Candra, Destya Akhirta Putri, Umi Lutfiana, Lina Khamsal selalu teman seperjuangan dan sahabat terbaik yang selalu memberi dukungan dan semangat.
12. Serta semua pihak yang tidak dapat disebutkan satu per satu yang telah membantu baik tenaga maupun pikiran dalam pelaksanaan kegiatan penelitian dan penyusunan skripsi ini.

DAFTAR ISI

|  |      |
|--|------|
| PERSEMBAHAN .....                                    | iii  |
| MOTTO .....  | iv   |
| PERNYATAAN.....                                      | v    |
| SKRIPSI.....   | vi   |
| PENGESAHAN PEMBIMBING.....                           | vii  |
| PENGESAHAN PENGUJI.....                              | viii |
| RINGKASAN .....                                      | ix   |
| PRAKATA .....  | xi   |
| DAFTAR ISI.....                                      | xiii |
| DAFTAR TABEL.....                                    | xx   |
| DAFTAR GAMBAR .....                                  | xxii |
| BAB 1. PENDAHULUAN .....                             | 1    |
| 1.1 Latar Belakang .....                             | 1    |
| 1.2 Rumusan Masalah .....                            | 7    |
| 1.3 Tujuan Dan Manfaat.....                          | 7    |
| <b>1.3.1 Tujuan</b> .....                            | 7    |
| <b>1.3.2 Manfaat</b> .....                           | 7    |
| 1.4 Batasan Masalah.....                             | 8    |
| 1.5 Sistematika Penulisan.....                       | 10   |
| BAB 2. TINJAUAN PUSTAKA .....                        | 12   |
| 2.1 Alur Pemilihan Kepala Sekolah .....              | 12   |
| 2.2 Penelitian Terdahulu.....                        | 12   |
| 2.3 <i>Profile Matching</i> (Pencocokan Profil)..... | 14   |
| 2.4 Metode Analytic Hierarchy Process (AHP) .....    | 15   |
| 2.5 Sistem Penunjang Keputusan .....                 | 21   |
| BAB 3 METODE PENELITIAN.....                         | 22   |
| 3.1 Jenis Penelitian .....                           | 22   |

|   |     |
|---|-----|
| 3.2 Tempat dan Waktu Penelitian .....   | 22  |
| 3.3 Teknik Penyajian dan Analisis Data .....  | 22  |
| <b>3.3.1 Analisis Kebutuhan (<i>Requirments Definition</i>)</b> .....                     | 23  |
| <b>3.3.2 Desain Sistem (<i>System and Software Design</i>)</b> .....                      | 24  |
| <b>3.3.3 Implementasi dan Pengujian Unit (<i>Implementation and Unit Testing</i>)</b> 25  |     |
| <b>3.3.4 Integritas dan Pengujian Sistem (<i>Integration and System Testing</i>)</b> . 25 |     |
| <b>3.3.5 Pemeliharaan (<i>Operational Maintance</i>)</b> .....                            | 26  |
| BAB 4 PENGEBANGAN SISTEM .....  | 27  |
| 4.1 Analisis Kebutuhan .....  | 27  |
| <b>4.1.1 SOP (<i>Statement of purpose</i>)</b> .....                                      | 27  |
| <b>4.1.2 Kebutuhan Fungsional</b> .....   | 28  |
| <b>4.1.3 Kebutuhan Non-Fungsional</b> .....   | 29  |
| <b>4.1.4 Fungsi Sistem</b> .....  | 29  |
| 4.2 Desain Sistem .....   | 31  |
| <b>4.2.1 Business Process</b> .....   | 31  |
| <b>4.2.2 Usecase Diagram</b> .....  | 32  |
| <b>3.2.3 Usecase Skenario</b> .....   | 39  |
| <b>4.2.4 Activity Diagram</b> .....   | 65  |
| <b>4.2.5 Sequence Diagram</b> .....   | 71  |
| <b>4.2.6 Class Diagram</b> .....  | 77  |
| <b>4.2.7 Entity Relationship Diagram (ERD)</b> .....                                      | 78  |
| 4.3 Penulisan Kode Program .....  | 79  |
| 4.4 Pengujian .....   | 84  |
| <b>4.4.1 White Box</b> .....  | 84  |
| <b>4.4.2 Black Box</b> .....  | 95  |
| <b>4.4.3 Uji Validitas</b> .....  | 96  |
| BAB 5. HASIL DAN PEMBAHASAN.....  | 101 |

|   |     |
|---|-----|
| 5.1 Hasil Pembuatan Sistem Penilaian Prestasi Kerja Guru Sebagai Luaran Pemilihan Calon Kepala Sekolah Menggunakan Metode <i>Profile Matching</i> – AHP             | 101 |
| 5.1.1 Tampilan Awal Sistem.....   | 101 |
| 5.1.2 Tampilan Home .....   | 102 |
| 5.1.3 Fitur Manajemen Data Guru (Admin) .....   | 104 |
| 5.1.4 Fitur Manajemen Data User (Admin).....  | 108 |
| 5.1.5 Fitur Manajemen Data Pribadi Guru .....   | 110 |
| 5.1.6 Fitur Update Password Pribadi .....   | 112 |
| 5.1.7 Fitur Melihat Kriteria Penilaian Kinerja.....   | 113 |
| 5.1.8 Fitur Manajemen Nilai Ideal <i>Profile Matching</i> .....   | 114 |
| 5.1.9 Fitur Melihat Nilai Kinerja <i>Profile Matching</i> .....   | 115 |
| 5.1.10 Fitur Manajemen Bobot Kriteria AHP .....   | 116 |
| 5.1.11 Fitur Melihat Bobot Kriteria AHP .....   | 117 |
| 5.1.12 Fitur Manajemen Verifikasi Peringkat Nilai Kinerja .....   | 119 |
| 5.1.13 Fitur Melihat Peringkat Nilai Kinerja Terverifikasi .....  | 120 |
| 5.1.14 Fitur Manajemen Update Rekap Hasil Peringkat .....   | 121 |
| 5.1.15 Fitur Melihat Rekap Peringkat.....   | 122 |
| 5.2 Hasil Penerapan Perhitungan <i>Profile Matching</i> .....   | 124 |
| 5.3 Hasil Penerapan Perhitungan Metode <i>Analytic Hierarchy Process</i> (AHP)  | 128 |
| 5.4 Hasil Penerapan Penggabungan Metode <i>Profile Matching</i> - AHP .....   | 133 |
| 5.6 Pembahasan Implementasi Metode <i>Profile Matching</i> dan AHP pada istem Penilaian Prestasi Kerja Guru Sebagai Luaran Pemilihan Calon Kepala Sekolah           | 134 |
| 5.6.1 Pembahasan Pembuatan Sistem Penilaian Prestasi Kerja Guru Sebagai Luaran Pemilihan Calon Kepala Sekolah Menggunakan Metode <i>Profile Matching</i> – AHP..... | 134 |
| 5.6.2 Pembahasan Implementasi Metode <i>Profile Matching</i> .....  | 136 |
| 5.6.3 Pembahasan Implementasi Metode AHP .....  | 137 |
| 5.6.4 Pembahasan Penerapan Penggabungan Metode <i>Profile Matching</i> – AHP  | 140 |
| BAB 6. PENUTUP .....  | 142 |

|   |     |
|---|-----|
| 6.1 Kesimpulan.....   | 142 |
| 6.2 Saran .....   | 143 |
| DAFTAR PUSTAKA .....  | 145 |
| LAMPIRAN.....   | 146 |
| Lampiran A. <i>Scenario</i> .....   | 146 |
| A1. Usecase Manajemen Data User (Update) .....                            | 146 |
| A2. Usecase Manajemen Data Pribadi Guru .....                             | 147 |
| A3. Usecase Update Password Pribadi .....                                 | 150 |
| A4. Usecase Melihat Kriteria Penilaian Kinerja Guru .....                 | 152 |
| A5. Usecase Melihat Nilai Kinerja Guru.....                               | 153 |
| A6. Usecase Manajemen Nilai Ideal Profile Matching .....                  | 153 |
| A7. Usecase Manajemen Bobot Kriteria AHP .....                            | 156 |
| A8. Usecase Melihat Nilai Kinerja Profile Matching (View) .....           | 157 |
| A9. Usecase Melihat Bobot Kriteria AHP.....                               | 158 |
| A10. Usecase Melihat Peringkat Nilai Kinerja.....                         | 159 |
| A11. Usecase Manajemen Verifikasi Peringkat Nilai Kinerja .....           | 159 |
| A12. Usecase Melihat Peringkat Nilai Kinerja Terverifikasi .....          | 161 |
| A13. Usecase Manajemen Update Rekap Hasil Peringkat .....                 | 162 |
| A14. Usecase Melihat Rekap Penilaian.....                                 | 163 |
| A15. Logout .....   | 164 |
| Lampiran B. Activity Diagram.....   | 166 |
| B1. Activity Diagram Manajemen Data User (Update) Status Aktif User.....  | 166 |
| B2. Activity Diagram Manajemen Data User (Update) Status Nonaktif User .. | 167 |
| B3. Activity Diagram Manajemen Data User (Update) Reset Password User.    | 167 |
| B4. Activity Diagram Manajemen Data Pribadi (View) .....                  | 168 |
| B5. Activity Diagram Manajemen Data Pribadi (Edit) .....                  | 169 |
| B6. Activity Diagram Update Password Pribadi .....                        | 171 |
| B7. Activity Diagram Melihat Kriteria Penilaian Kinerja Guru (View) ..... | 172 |
| B8. Activity Diagram Melihat Nilai Kinerja Guru (View).....               | 172 |
| B9. Activity Diagram Manajemen Nilai Ideal Profile Matching (View) .....  | 173 |



|  |     |
|--|-----|
| B10. Activity Diagram Manajemen Nilai Ideal Profile Matching (Edit) .....            | 174 |
| B11. Activity Diagram Melihat Nilai Kinerja Profile Matching (View).....             | 175 |
| B12. Activity Diagram Manajemen Bobot Kriteria AHP (View) .....                      | 175 |
| B13. Activity Diagram Manajemen Bobot Kriteria AHP (Edit) .....                      | 176 |
| B14. Activity Diagram Manajemen Bobot Kriteria AHP (View) .....                      | 177 |
| B15. Activity Diagram Manajemen Verifikasi Peringkat Nilai Kinerja (View)            | 177 |
| B16. Activity Diagram Manajemen Verifikasi Peringkat Nilai Kinerja (Update)<br>..... | 178 |
| B17. Activity Diagram Melihat Peringkat Nilai Kinerja Terverifikasi (View).          | 178 |
| B18. Activity Diagram Manajemen Update Rekap Hasil Peringkat (View).....             | 179 |
| B19. Activity Diagram Manajemen Update Rekap Hasil Peringkat (Update)..              | 180 |
| B20. Activity Diagram Melihat Rekap Penilaian (View).....                            | 181 |
| B21. Activity Diagram Logout .....   | 182 |
| Lampiran C. <i>Sequence Diagram</i> .....  | 182 |
| C1. Sequence Diagram Manajemen Data User (Update) Status Aktif User .....            | 182 |
| C2. Sequence Diagram Manajemen Data User (Update) Nonaktif Aktif User.               | 183 |
| C3. Sequence Diagram Manajemen Data User (Update) Reset Password User<br>.....       | 183 |
| C4. Sequence Diagram Manajemen Data Pribadi (View) .....                             | 183 |
| C5. Sequence Diagram Manajemen Data Pribadi (Edit) .....                             | 184 |
| C6. Sequence Diagram Update Password Pribadi .....                                   | 185 |
| C7. Sequence Diagram Melihat Kriteria Penilaian Kinerja Guru (View) .....            | 185 |
| C8. Sequence Diagram Melihat Nilai Kinerja Guru (View) .....                         | 186 |
| C9. Sequence Diagram Manajemen Nilai Ideal Profile Matching (View) .....             | 186 |
| C10. Sequence Diagram Manajemen Nilai Ideal Profile Matching (Edit) .....            | 187 |
| C11. Sequence Diagram Melihat Nilai Kinerja Profile Matching (View) .....            | 188 |
| C12. Sequence Diagram Manajemen Bobot Kriteria AHP (View) .....                      | 188 |
| C13. Sequence Diagram Manajemen Bobot Kriteria AHP (Edit) .....                      | 189 |
| C14. Sequence Diagram Manajemen Bobot Kriteria AHP (View) .....                      | 189 |

|   |     |
|---|-----|
| C15. Sequence Diagram Manajemen Verifikasi Peringkat Nilai Kinerja (View)   | 190 |
| C16. Sequence Diagram Manajemen Verifikasi Peringkat Nilai Kinerja (Update) | 190 |
| C17. Sequence Diagram Melihat Peringkat Nilai Kinerja Terverifikasi (View)  | 191 |
| C18. Sequence Diagram Manajemen Update Rekap Hasil Peringkat (View)...      | 191 |
| C19. Sequence Diagram Manajemen Update Rekap Hasil Peringkat (Update)       | 192 |
| C20. Sequence Diagram Melihat Rekap Penilaian (View).....                   | 193 |
| C21. Sequence Diagram Logout.....   | 193 |
| Lampiran D. Penulisan Kode Program.....                                     | 194 |
| D1. Penulisan Kode Program class controllers/C_ahp .....                    | 194 |
| D2. Penulisan Kode Program class controllers/C_AHPguru.....                 | 198 |
| D3. Penulisan Kode Program class controllers/C_datapegawai .....            | 202 |
| D4. Penulisan Kode Program class controllers/C_datauser.....                | 215 |
| D5. Penulisan Kode Program class controllers/C_hasil .....                  | 217 |
| D6. Penulisan Kode Program class controllers/c_manager .....                | 221 |
| D7. Penulisan Kode Program class controllers/C_pm.....                      | 223 |
| D8. Penulisan Kode Program class controllers/C_PMguru.....                  | 226 |
| D9. Penulisan Kode Program class controllers/C_profile .....                | 229 |
| D10. Penulisan Kode Program class controllers/C_rekap.....                  | 233 |
| D11. Penulisan Kode Program class controllers/dashboard .....               | 235 |
| D12. Penulisan Kode Program class controllers/Login .....                   | 236 |
| D13. Penulisan Kode Program class models/M_ahp.....                         | 238 |
| D14. Penulisan Kode Program class models/M_datapegawai.....                 | 240 |
| D15. Penulisan Kode Program class models/M_hasil .....                      | 246 |
| D16. Penulisan Kode Program class models/M_manager.....                     | 249 |
| D17. Penulisan Kode Program class models/M_pm .....                         | 251 |
| D18. Penulisan Kode Program class models/M_rekap.....                       | 254 |
| D19. Penulisan Kode Program class models/Mdatauser .....                    | 255 |

|   |     |
|---|-----|
| D20. Penulisan Kode Program class models/Mlogin .....   | 257 |
| D21. Penulisan Kode Program class models/Mprofile ..... | 258 |
| Lampiran E. Pengujian Black Box .....                   | 263 |



## DAFTAR TABEL

|  |     |
|--|-----|
| Tabel 2. 1 Contoh Bobot Nilai Gap .....  | 15  |
| Tabel 2. 2 Matrik Perbandingan Berpasangan .....                                       | 16  |
| Tabel 2. 3 Skala Prioritas Kriteria.....   | 17  |
| Tabel 2. 4 Nilai Rata-Rata Konsistensi.....  | 19  |
| Tabel 4. 1 Deskripsi Aktor .....   | 35  |
| Tabel 4. 2 Deskripsi Usecase .....   | 36  |
| Tabel 4. 3 Tabel Skenario Usecase Login.....   | 39  |
| Tabel 4. 4 Tabel Usecase Skenario Manajemen Data Guru (View) .....                     | 43  |
| Tabel 4. 5 Tabel Usecase Skenario Manajemen Data Guru (Input) .....                    | 45  |
| Tabel 4. 6 Tabel Usecase Skenario Manajemen Data Guru (Edit) .....                     | 51  |
| Tabel 4. 7 Tabel Usecase Skenario Manajemen Tambah Nilai Guru (View).....              | 56  |
| Tabel 4. 8 Tabel Usecase Skenario Manajemen Tambah Nilai Guru (Input).....             | 58  |
| Tabel 4. 9 Kode program function bobot_kriteria().....                                 | 79  |
| Tabel 4. 10 <i>Kode program</i> function <i>edit_bobot_kriteria()</i> .....            | 82  |
| Tabel 4. 11 <i>Kode program</i> function <i>md_relasi_kriteria()</i> .....             | 83  |
| Tabel 4. 12 Test case function <i>metode</i> Profile Matching <i>index()</i> .....     | 90  |
| Tabel 4. 13 Test case function <i>metode</i> AHP <i>bobot_kriteria_manager()</i> ..... | 92  |
| Tabel 4. 14 Tabel r Product Moment .....   | 100 |
| Tabel 5. 1 Data Nilai Kinerja Prestasi Kinerja Guru .....                              | 125 |
| Tabel 5. 2 Data Bobot Kriteria Penilaian Prestasi Kinerja Guru .....                   | 126 |
| Tabel 5. 3 Detail Nilai Ideal .....  | 127 |
| Tabel 5. 4 Pemetaan Gap .....  | 127 |
| Tabel 5. 5 Data Bobot Gap.....   | 128 |
| Tabel 5. 6 Data Pembobotan Gap .....   | 128 |
| Tabel 5. 7 Kriteria Kinerja Guru .....   | 129 |
| Tabel 5. 8 Tabel Perbandingan Bobot Kriteria .....                                     | 130 |
| Tabel 5. 9 Perhitungan Eigen Vektor Normalisasi.....                                   | 131 |
| Tabel 5. 10 Pengabungan Metode Profile Matching - AHP .....                            | 133 |
| Tabel A 1 Tabel Usecase Manajemen Data User (Update) .....                             | 146 |
| Tabel A 2 Usecase Skenario Manajemen Data Pribadi Guru (View).....                     | 148 |
| Tabel A 3 Usecase Skenario Manajemen Data Pribadi Guru (Edit).....                     | 149 |
| Tabel A 4 Usecase Skenario Update Password Pribadi (Edit).....                         | 151 |
| Tabel A 5 Usecase Skenario Melihat Kriteria Penilaian Kinerja Guru (View) .....        | 152 |
| Tabel A 6 Usecase Skenario Melihat Nilai Kinerja (View).....                           | 153 |
| Tabel A 7 Usecase Skenario Manajemen Nilai Ideal Profile Matching (View).....          | 154 |

|   |     |
|---|-----|
| Tabel A 8 Usecase Skenario Manajemen Nilai Ideal Profile Matching (Edit) .....            | 154 |
| Tabel A 9 Usecase Skenario Manajemen Bobot Kriteria AHP (View) .....                      | 156 |
| Tabel A 10 Usecase Skenario Manajemen Nilai Kinerja Profile Matching (Edit) ...           | 156 |
| Tabel A 11 Usecase Skenario Melihat Nilai Kinerja Profile Matching (View) .....           | 157 |
| Tabel A 12 Usecase Skenario Melihat Bobot Kriteria AHP (View) .....                       | 158 |
| Tabel A 13 Usecase Skenario A14 Melihat Peringkat Nilai Kinerja (View) .....              | 159 |
| Tabel A 14 Usecase Skenario Menejaman Verifikasi Peringkat Nilai Kinerja (View)<br>.....  | 160 |
| Tabel A 15 Menejaman Verifikasi Peringkat Nilai Kinerja (Update) .....                    | 160 |
| Tabel A 16 Usecase Skenario Melihat Peringkat Nilai Kinerja Terverifikasi (View)<br>..... | 161 |
| Tabel A 17 Usecase Skenario Melihat Update Rekap Hasil Peringkat (View) .....             | 162 |
| Tabel A 18 Usecase Skenario Manajemen Update Rekap Hasil Peringkat (Update)               | 163 |
| Tabel A 19 Usecase Skenario Melihat Rekap Penilaian (View).....                           | 164 |
| Tabel A 20 Usecase Skenario Logout.....   | 164 |
|   |     |
| Tabel D 1 Penulisan Kode Program class controllers/C_ahp .....                            | 194 |
| Tabel D 2 Penulisan Kode Program class controllers/C_AHPguru.....                         | 198 |
| Tabel D 3 Penulisan Kode Program class controllers/C_datapegawai .....                    | 202 |
| Tabel D 4 Penulisan Kode Program class controllers/C_datauser.....                        | 215 |
| Tabel D 5 Penulisan Kode Program class controllers/C_hasil .....                          | 217 |
| Tabel D 6 . Penulisan Kode Program class controllers/c_manager .....                      | 221 |
| Tabel D 7 Penulisan Kode Program class controllers/C_pm .....                             | 224 |
| Tabel D 8 Penulisan Kode Program class controllers/C_Pmguru .....                         | 226 |
| Tabel D 9 Penulisan Kode Program class controllers/C_profile .....                        | 229 |
| Tabel D 10 Penulisan Kode Program class controllers/C_rekap .....                         | 233 |
| Tabel D 11 Penulisan Kode Program class controllers/dashboard .....                       | 236 |
| Tabel D 12 Penulisan Kode Program class controllers/Login .....                           | 236 |
| Tabel D 13 Penulisan Kode Program class models/M_ahp.....                                 | 238 |
| Tabel D 14 Penulisan Kode Program class models/M_datapgawai.....                          | 240 |
| Tabel D 15 Penulisan Kode Program class models/M_hasil .....                              | 246 |
| Tabel D 16 Penulisan Kode Program class models/M_manager .....                            | 249 |
| Tabel D 17 Penulisan Kode Program class models/M_pm.....                                  | 251 |
| Tabel D 18 Penulisan Kode Program class models/M_rekap .....                              | 254 |
| Tabel D 19 Penulisan Kode Program class models/Mdatauser .....                            | 256 |
| Tabel D 20 Penulisan Kode Program class models/Mlogin.....                                | 257 |
| Tabel D 21 Penulisan Kode Program class models/Mprofile .....                             | 258 |

DAFTAR GAMBAR

|   |     |
|---|-----|
| Gambar 2. 1 Penerapan Profile Matching-AHP.....   | 20  |
| Gambar 3. 1 Tahapan Waterfall.....  | 23  |
| Gambar 4. 1 Bussines Procces Sistem Penilaian Pretasi Kerja Guru Sebagai Luaran Pemilihan Calon Kepala Sekolah..... | 32  |
| Gambar 4. 2 sistem penilaian prestasi kerja guru sebagai luaran pemilihan calon kepala sekolah.....                 | 35  |
| Gambar 4. 3 Activity <i>Diagram</i> Login.....  | 67  |
| Gambar 4. 4 Activity <i>Diagram</i> manajemen data guru (view) .....  | 68  |
| Gambar 4. 5 Activity <i>Diagram</i> manajemen data guru (input).....  | 68  |
| Gambar 4. 6 Activity <i>Diagram</i> manajemen data guru (edit) .....  | 69  |
| Gambar 4. 7 Activity <i>Diagram</i> manajemen tambah nilai guru.....  | 70  |
| Gambar 4. 8 Activity <i>Diagram</i> manajemen tambah nilai guru (input).....  | 70  |
| Gambar 4. 9 <i>Sequence Diagram</i> Login .....   | 74  |
| Gambar 4. 10 <i>Sequence Diagram</i> Manajemen Data Guru (Input).....   | 74  |
| Gambar 4. 11 <i>Sequence Diagram</i> Manajemen Data Guru (View)) .....  | 74  |
| Gambar 4. 12 <i>Sequence Diagram</i> Manajemen Data Guru (Edit).....  | 76  |
| Gambar 4. 13 <i>Sequence Diagram</i> Manajemen Tambah Nilai Guru (View) .....                                       | 76  |
| Gambar 4. 14 <i>Sequence Diagram</i> Manajemen Data Nilai Guru (Input) .....  | 76  |
| Gambar 4. 15 <i>Class Diagram</i> .....   | 77  |
| Gambar 4. 16 <i>Entity Relational Diagram</i> .....   | 78  |
| Gambar 4. 17 Listing program Metode AHP function bobot_kriteria_manager() .....                                     | 88  |
| Gambar 4. 18 Listing program metode Profile Matching <i>function index()</i> .....                                  | 88  |
| Gambar 4. 19 Diagram Alir Metode <i>Profile Matching</i> Function <i>index()</i> .....                              | 89  |
| Gambar 4. 20 Diagram Alir Metode AHP function bobot_kriteria().....   | 89  |
| Gambar 5. 1 Tampilan Awal Sistem.....   | 102 |
| Gambar 5. 2 Tampilan Home Level Admin .....   | 103 |
| Gambar 5. 3 Tampilan Home Level Kepsek.....   | 103 |
| Gambar 5. 4 Tampilan Home Level Guru .....  | 104 |
| Gambar 5. 5 Fitur Manajemen Data Guru .....   | 105 |
| Gambar 5. 6 Fitur Manajemen Data Guru .....   | 105 |
| Gambar 5. 8 Detail Data Guru .....  | 107 |
| Gambar 5. 10 Fitur Edit Data Guru.....  | 108 |
| Gambar 5. 11 Fitur Edit Data Guru.....  | 108 |

|   |     |
|---|-----|
| Gambar 5. 13 Fitur Non Aktifikan User .....   | 110 |
| Gambar 5. 14 Fitur Reset Passowrd User .....  | 110 |
| Gambar 5. 15 Fitur Manajemen Data Pribadi Guru Level Guru.....                      | 111 |
| Gambar 5. 16 Fitur Manajemen Data Pribadi Guru Level Kepsek.....                    | 111 |
| Gambar 5. 17 Fitur Update Password Pribadi Level Kepsek.....                        | 112 |
| Gambar 5. 18 Fitur Update Password Pribadi Level Guru .....                         | 113 |
| Gambar 5. 19 Fitur Melihat Kriteria Penilaian Kinerja .....                         | 114 |
| Gambar 5. 20 Fitur Manajemen Nilai Ideal <i>Profile Matching</i> .....              | 115 |
| Gambar 5. 21 Fitur Manajemen Nilai Ideal <i>Profile Matching</i> .....              | 115 |
| Gambar 5. 22 Fitur Melihat Nilai Kinerja <i>Profile Matching</i> .....              | 116 |
| Gambar 5. 23 Fitur Melihat Nilai Kinerja <i>Profile Matching</i> .....              | 116 |
| Gambar 5. 24 Fitur Manajemen Bobot Kriteria AHP .....                               | 117 |
| Gambar 5. 25 Fitur Melihat Bobot Kriteria AHP .....                                 | 118 |
| Gambar 5. 26 Fitur Melihat Peringkat Nilai Kinerja .....                            | 119 |
| Gambar 5. 27 Fitur Manajemen Verifikasi Peringkat Nilai Kinerja.....                | 119 |
| Gambar 5. 28 Fitur Manajemen Verifikasi Peringkat Nilai Kinerja.....                | 120 |
| Gambar 5. 29 Fitur Melihat Peringkat Nilai Kinerja Terverifikasi.....               | 121 |
| Gambar 5. 30 Fitur Manajemen Update Rekap Hasil Peringkat.....                      | 122 |
| Gambar 5. 31 Fitur Melihat Rekap Peringkat .....                                    | 123 |
| Gambar 5. 32 Fitur Melihat Rekap Peringkat .....                                    | 123 |
| Gambar 5. 33 Fitur Melihat Rekap Peringkat .....                                    | 124 |
| Gambar 5. 34 Kode Program Pemetaan Gap .....  | 136 |
| Gambar 5. 35 Kode Program Pembobotan Gap.....                                       | 137 |
| Gambar 5. 36 Kode Program Menghitung Jumlah Per Kolom.....                          | 138 |
| Gambar 5. 37 Kode Program Menghitung Nilai Bobot Kriteria.....                      | 139 |
| Gambar 5. 38 Implementasi kode program menghitung lamda, ci, dan cr .....           | 140 |
| Gambar 5. 39 Hasil Penentuan Peringkat .....  | 141 |
| <br>  |     |
| Gambar B 1 Activity Diagram Manajemen Data User (Update) Status Aktif User .        | 166 |
| Gambar B 2 Activity Diagram Manajemen Data User (Update) Status Nonaktif User ..... | 167 |
| Gambar B 3 Activity Diagram Manajemen Data User (Update) Reset Password User .....  | 167 |
| Gambar B 4 Activity Diagram Manajemen Data Pribadi (View).....                      | 168 |
| Gambar B 5 Activity Diagram Manajemen Data Pribadi (Edit).....                      | 171 |
| Gambar B 6 Activity Diagram Update Password Pribadi.....                            | 171 |
| Gambar B 7 Activity Diagram Melihat Kriteria Penilaian Kinerja Guru (View).....     | 172 |
| Gambar B 8 Activity Diagram Melihat Nilai Kinerja Guru (View) .....                 | 172 |
| Gambar B 9 Activity Diagram Manajemen Nilai Ideal Profile Matching (View)....       | 173 |
| Gambar B 10 Activity Diagram Manajemen Nilai Ideal Profile Matching (Edit)....      | 175 |

|  |     |
|--|-----|
| Gambar B 11 Activity Diagram Manajemen Nilai Ideal Profile Matching (View)..               | 175 |
| Gambar B 12 Activity Diagram Manajemen Bobot Kriteria AHP (View).....                      | 175 |
| Gambar B 13 Activity Diagram Manajemen Bobot Kriteria AHP (Edit).....                      | 176 |
| Gambar B 14 Activity Diagram Manajemen Bobot Kriteria AHP (View).....                      | 177 |
| Gambar B 15 Activity Diagram Manajemen Verifikasi Peringkat Nilai Kinerja (View)<br>.....  | 177 |
| Gambar B 16 Activity Diagram Manajemen Verifikasi Peringkat Nilai Kinerja<br>(Update)..... | 178 |
| Gambar B 17 Activity Diagram Melihat Peringkat Nilai Kinerja Terverifikasi (View)<br>..... | 178 |
| Gambar B 18 Activity Diagram Manajemen Update Rekap Hasil Peringkat (View)<br>.....        | 179 |
| Gambar B 19 . Activity Diagram Manajemen Update Rekap Hasil Peringkat (Update)<br>.....    | 180 |
| Gambar B 20 Activity Diagram Melihat Rekap Penilaian (View) .....                          | 181 |
| Gambar B 21 Activity Diagram Logout .....  | 182 |
|  |     |
| Gambar C 1 Sequence Diagram Manajemen Data User (Update) Status Aktif User                 | 182 |
| Gambar C 2 Sequence Diagram Manajemen Data User (Update) Nonaktif Aktif User<br>.....      | 183 |
| Gambar C 3 Sequence Diagram Manajemen Data User (Update) Reset Password User<br>.....      | 183 |
| Gambar C 4 Sequence Diagram Manajemen Data Pribadi (View).....                             | 183 |
| Gambar C 5 Sequence Diagram Manajemen Data Pribadi (Edit).....                             | 184 |
| Gambar C 6 Sequence Diagram Update Password Pribadi .....                                  | 185 |
| Gambar C 7 Sequence Diagram Melihat Kriteria Penilaian Kinerja Guru (View)...              | 185 |
| Gambar C 8 Sequence Diagram Melihat Nilai Kinerja Guru (View).....                         | 186 |
| Gambar C 9 Sequence Diagram Manajemen Nilai Ideal Profile Matching (View) .                | 186 |
| Gambar C 10 Sequence Diagram Manajemen Nilai Ideal Profile Matching (Edit) .               | 187 |
| Gambar C 11 Sequence Diagram Melihat Nilai Kinerja Profile Matching (View)..               | 188 |
| Gambar C 12 Sequence Diagram Manajemen Bobot Kriteria AHP (View) .....                     | 188 |
| Gambar C 13 Sequence Diagram Manajemen Bobot Kriteria AHP (Edit) .....                     | 189 |
| Gambar C 14 Sequence Diagram Manajemen Bobot Kriteria AHP (View) .....                     | 189 |
| Gambar C 15 Sequence Diagram Manajemen Verifikasi Peringkat Nilai Kinerja (View)<br>.....  | 190 |
| Gambar C 16 Sequence Diagram Manajemen Verifikasi Peringkat Nilai Kinerja<br>(Update)..... | 190 |
| Gambar C 17 Sequence Diagram Melihat Peringkat Nilai Kinerja Terverifikasi (View)<br>..... | 191 |



|  |     |
|--|-----|
| Gambar C 18 Sequence Diagram Manajemen Update Rekap Hasil Peringkat (View)   | 191 |
| Gambar C 19 Sequence Diagram Manajemen Update Rekap Hasil Peringkat (Update) | 192 |
| Gambar C 20 Sequence Diagram Melihat Rekap Penilaian (View)                  | 193 |
| Gambar C 21 Sequence Diagram Logout  | 193 |



## **BAB 1. PENDAHULUAN**

Bab pendahuluan adalah bab awal dari laporan tugas akhir. Pada bab ini membahas latar belakang, perumusan masalah, tujuan dan manfaat, batasan masalah, dan sistematika penulisan.

### **1.1 Latar Belakang**

Guru merupakan suatu profesi yang artinya suatu jabatan atau pekerjaan yang memerlukan keahlian khusus sebagai guru. Jenis pekerjaan ini tidak bisa dilakukan oleh sembarang orang diluar bidang kependidikan. Dalam Undang – Undang Sisdiknas Bab XI pasal 39, 40 dan 42 (Republik Indonesia, 2003) dinyatakan bahwa tugas dan fungsi guru adalah merencanakan dan melaksanakan proses pembelajaran, menilai hasil pembelajaran, melakukan pembimbingan dan pelatihan, serta melakukan penelitian dan pengabdian kepada masyarakat.

Kemampuan guru perlu ditingkatkan dan dikembangkan secara terus menerus menurut jabatan fungsional guru. Agar fungsi dan tugas guru dapat berjalan sesuai peraturan yang berlaku, maka diperlukan penilaian prestasi kerja guru untuk menilai hasil kerja pada masing-masing guru sehingga mewujudkan tenaga pendidik yang profesional. Penilaian prestasi kerja guru dapat meningkatkan pengetahuan dan keterampilan guru, sehingga kualitas pembelajaran yang dilakukan dapat meningkat. Hasil penilaian prestasi kerja guru dapat dimanfaatkan untuk menyusun profil kinerja guru sebagai input dalam penyusunan program Pengembangan Keprofesian Berkelanjutan.

Hasil penilaian prestasi kerja guru juga merupakan dasar penetapan perolehan nilai guru dalam rangka pengembangan karir guru sebagaimana yang tertulis dalam Peraturan Menteri Negara Pendayagunaan Aparatur Negara dan Reformasi Birokrasi Nomor 16 Tahun 2009 tentang Jabatan Fungsional Guru dan Angka Kreditnya.

Kepala sekolah merupakan pimpinan tertinggi didalam suatu sekolah. Kemajuan sekolah sangat dipengaruhi dari kemampuan kepala sekolah dalam

mengelola kebijakan di sekolah tersebut. Jabatan kepala sekolah merupakan pengembangan keprofesian kelanjutan karier bagi seorang guru. Diperlukan sumber daya manusia yang berkualitas dalam hal ini guru-guru yang memiliki potensi dan prestasi untuk menjadi kepala sekolah, sehingga dapat mewujudkan pendidikan yang berkualitas.

Mendapatkan kepala sekolah yang berkualitas bukan hal yang mudah. Diperlukan penyeleksian dalam pemilihan seorang kepala sekolah yang mampu membawa keberhasilan sekolah. Penilaian prestasi kerja guru merupakan cara untuk mengetahui seberapa pantas seorang guru agar memiliki hak untuk melanjutkan ke jenjang karier lebih tinggi yaitu sebagai kepala sekolah. Guru dengan syarat dan kriteria yang telah memenuhi standart syarat calon kepala sekolah dapat di beri hak mengajukan diri sebagai calon kepala sekolah berdasarkan seleksi penilaian prestasi kerja guru tersebut.

Tugas seleksi adminstrasi dalam pencalonan diri sebagai kepala sekolah dikerjakan pada tingkat sekolah. Kesulitan dan kesalahan lebih rentang terjadi karena banyaknya data kriteria dengan bobot berbeda-beda yang dihasilkan dari pelaporan penilaian selama menjadi guru sehingga menyebabkan keterlambatan dalam penilaian akhir. Hal itulah yang sama terjadi pada SMA Negeri Grujugan Bondowoso. Pada umumnya guru-guru yang menonjol disekolahnya dan mendapat penilaian yang baik dari pihak atasan mempunyai harapan untuk diangkat menjadi kepala sekolah. Hal itu diakibatkan dari perhitungan yang kurang memperhatikan bobot masing-masing kriteria penilaian, karena kesulitan dan banyaknya data dalam penilaian.

Berikut merupakan hasil dari wawancara yang dilakukan di SMA Negeri Grujugan Bondowoso mengenai hasil dari seleksi administrai pemilihan calon kepala sekolah tahun 2016 – 2017 melalui penilaian prestasi kerja guru yang dapat dilihat di tabel 1.1

Tabel 1.1 Hasil Penilaian Prestasi Kerja Guru

| No | Hasil Penilaian  | Jumlah Guru |
|----|--|-------------|
| 1  | Jumlah Keseluruhan Guru yang Melakukan Penilaian           | 20          |
| 2  | Jumlah Guru Berhak Mengajukan Menjadi Calon Kepala Sekolah | 5           |
| 3  | Jumlah Guru yang Diajukan Menjadi Calon Kepala Sekolah     | 3           |

Dalam data penilaian tersebut, terdapat 5 orang dengan nilai tertinggi yang berhak mencalon diri menjadi calon kepala sekolah, akan tetapi tidak semua guru yang berhak tersebut dapat mencalonkan diri menjadi kepala sekolah, karena maksimal hanya 3 guru lah yang boleh diajukan dari tiap sekolah untuk mengikuti seleksi lebih lanjut di tiap kabupaten kota. Jumlah guru yang berhak mengajukan menjadi calon kepala sekolah akhirnya dipilih ulang oleh kepala sekolah yang sedang menjabat dengan cara memilih guru dengan umur paling senior atau guru-guru yang menonjol disekolahnya untuk dicalonkan menjadi calon kepala sekolah

Kemudian, untuk detail hasil penilaian masing-masing guru yang diajukan menjadi calon kepala sekolah dapat dilihat pada tabel 1.2, 1.3, dan 1.4

Tabel 1.2 Detail Penilaian Masing-Masing Guru

| No | Guru         | Kriteria             | Nilai Angka | Penilaian      |
|----|--------------|----------------------|-------------|----------------|
|    | Guru Pertama | kualifikasi akademik | 2           | Tercapai       |
|    |              | Usia                 | 2           | Tercapai       |
|    |              | Kesehatan jasmani    | 3           | Tercapai       |
|    |              | Catatan hukum sipil  | 0           | Tidak Tercapai |

|  |  |                                 |   |                |
|--|--|---------------------------------|---|----------------|
|  |  | Kepemilikan sertifikat pendidik | 2 | Tercapai       |
|  |  | Lama pengalaman mengajar        | 0 | Tidak Tercapai |
|  |  | Jenis golongan                  | 3 | Tercapai       |
|  |  | DP3                             | 2 | Tercapai       |
|  |  | Kegiatan struktural tambahan    | 1 | Tercapai       |

Untuk tabel 1.2, tingkat pelanggaran wajar untuk catatan hukum sipil melebihi batas wajar sebagai calon kepala sekolah yaitu melebihi 3 kali maka mendapat nilai 0 dan lama pengalaman mengajar tidak sampai batas minimum syarat mencalonkan diri menjadi kepala sekolah yaitu 5 tahun juga mendapat nilai 0. Akan tetapi guru tersebut menonjol dalam proses pembelajaran sehari-hari disekolah. Maka Kepala Sekolah memberi hak lebih agar dia dapat mencalonkan diri sebagai kepala sekolah.

Tabel 1.3 Detail Penilaian Masing-Masing Guru

| No | Guru       | Kriteria                        | Nilai Angka | Penilaian      |
|----|------------|---------------------------------|-------------|----------------|
|    | Guru Kedua | kualifikasi akademik            | 2           | Tercapai       |
|    |            | Usia                            | 3           | Tercapai       |
|    |            | Kesehatan jasmani               | 0           | Tidak Tercapai |
|    |            | Catatan hukum sipil             | 2           | Tercapai       |
|    |            | Kepemilikan sertifikat pendidik | 2           | Tercapai       |
|    |            | Lama pengalaman mengajar        | 3           | Tercapai       |
|    |            | Jenis golongan                  | 7           | Tercapai       |
|    |            | DP3                             | 2           | Tercapai       |

|  |  |                              |   |          |
|--|--|------------------------------|---|----------|
|  |  | Kegiatan struktural tambahan | 1 | Tercapai |
|--|--|------------------------------|---|----------|

Untuk tabel 1.3, guru kedua memiliki nilai kriteria kesehatan jasmani tidak tercapai atau tidak sehat, akan tetapi guru tersebut memiliki umur tertua disekolah tersebut dengan nilai kriteria prestasi lainnya terbaik. Hal itu menyebabkan Kepala Sekolah memberi hak lebih agar guru tersebut dapat mencalonkan diri menjadi kepala sekolah untuk mengikuti tahap seleksi selanjutnya.

Tabel 1.4 Detail Penilaian Masing-Masing Guru

| No | Guru        | Kriteria                        | Nilai Angka | Penilaian |
|----|-------------|---------------------------------|-------------|-----------|
|    | Guru Ketiga | kualifikasi akademik            | 2           | Tercapai  |
|    |             | Usia                            | 2           | Tercapai  |
|    |             | Kesehatan jasmani               | 2           | Tercapai  |
|    |             | Catatan hukum sipil             | 3           | Tercapai  |
|    |             | Kepemilikan sertifikat pendidik | 2           | Tercapai  |
|    |             | Lama pengalaman mengajar        | 2           | Tercapai  |
|    |             | Jenis golongan                  | 4           | Tercapai  |
|    |             | DP3                             | 2           | Tercapai  |
|    |             | Kegiatan struktural tambahan    | 1           | Tercapai  |

Untuk tabel 1.4, guru ketiga mampu mencapai semua kriteria penilaian ideal sebagai calon kepala sekolah, dan otomatis dia langsung disetujui kepala sekolah untuk mencalonkan diri sebagai calon kepala sekolah.

Guru ketiga tidaklah mengalami masalah seleksi penilaian prestasi kerja guru, tetapi untuk guru pertama dan kedua yang juga lulus seleksi, hal tidak dibenarkan karena ada syarat mutlak batasan nilai prestasi kerja guru yang digunakan sebagai luaran pemilihan calon kepala sekolah. Guru pertama dan kedua tidak mencapai batas minimum nilai dari beberapa kriteria penilaian, akan tetapi mereka masih di ajukan menjadi calon kepala sekolah. Penilaian inilah yang menyebabkan kurang objective.

Kriteria administrasi lainnya yang harus dipenuhi dalam pencalonan diri sebagai kepala sekolah adalah kualifikasi akademik yang dimiliki, usia, kesehatan jasmani, catatan hukum sipil, kepemilikan sertifikat pendidik, lama pengalaman mengajar, jenis golongan, DP3, serta kegiatan struktural tugas tambahan yang dimiliki. Dari permasalahan tersebut maka dirancanglah Sistem Penilaian Prestasi Kerja Guru Sebagai Luaran Pemilihan Calon Kepala Sekolah ini menggunakan metode *Profile Matching* – AHP. Menurut Mann dan Watson, Sistem Penunjang Keputusan adalah Sistem yang interaktif, membantu pengambilan keputusan melalui penggunaan data dan model-model keputusan untuk memecahkan masalah-masalah yang sifatnya semi terstruktur dan tidak terstruktur.

Metode yang digunakan adalah penggabungan *Profile Matching* dan AHP. Pemilihan metode sendiri dilakukan karena dalam merekap dan menghitung hasil penilaian calon peserta nama guru yang boleh mencalonkan diri sebagai calon kepala sekolah terdiri dari berbagai macam kriteria penilaian dengan masing-masing bobot perbandingan hubungan antar kriteria satu dengan yang lain memiliki nilai berbeda, dari situ dibutuhkan metode AHP, sedangkan untuk membandingkan jarak selisih nilai ideal capaian prestasi kinerja guru yang dapat digunakan sebagai luaran calon kepala sekolah dengan masing-masing data nilai yang akan di hitung, yaitu hasil kerja prestasi guru menggunakan metode *Profile Matching*.

## 1.2 Rumusan Masalah

Berdasarkan dari beberapa permasalahan yang telah diuraikan diatas, maka dapat diambil rumusan masalah sebagai berikut :

1. Bagaimana menerapkan metode *Profile Matching-Analytic Hierarchy Process* (AHP) pada proses pemilihan calon kepala sekolah di SMA Negeri Grujungan Bondowoso?
2. Bagaimana merancang dan membangun sebuah Sistem Penilaian Prestasi Kerja Guru Sebagai Luaran Pemilihan Calon Kepala Sekolah di SMA Negeri Grujungan Bondowoso Menggunakan Metode *Profile Matching – AHP*?

## 1.3 Tujuan Dan Manfaat

Berikut merupakan tujuan yang ingin dicapai dan manfaat yang ingin diperoleh dalam penelitian ini.

### 1.3.1 Tujuan

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah:

- a. Menerapkan metode *Profile Matching-AHP (Analytic Hierarchy Process)* pada proses sistem penilaian prestasi kerja guru sebagai luaran pemilihan calon kepala sekolah pada SMA Negeri Grujungan Bondowoso
- b. Merancang dan membangun sebuah sistem penilaian prestasi kerja guru sebagai luaran pemilihan calon kepala sekolah pada SMA Negeri Grujungan Bondowoso ini menggunakan Metode *Profile Matching-Analytic Hierarchy Process* yang *user friendly*.

### 1.3.2 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Bagi Kepala Sekolah



- a. Membantu kepala sekolah dalam menyeleksi dan memberi kewenangan siapa guru yang berhak mencalonkan diri sebagai kepala sekolah.
  - b. Membantu memonitoring kinerja guru dari hasil peringkat guru keseluruhan yang didapat di sekolah.
2. Bagi Guru
    - a. Dapat sebagai informasi perbandingan keseluruhan nilai yang telah di capai selama menjadi guru, sehingga dapat mengetahui tolak ukur tingkat yang dimiliki dibanding guru lainnya di satu sekolah.
  3. Bagi Tata Usaha
    - a. Membantu dalam proses perhitungan data-data nilai kriteria yang masuk, sehingga pengerjaan untuk proses perhitungan seleksi lebih cepat dan obyektif
  4. Bagi Penelitian
    - a. Sebagai suatu pengetahuan dan pengalaman untuk membandingkan antar teori yang didapat selama perkuliahan untuk diterapkan secara nyata didunia nyata.

#### **1.4 Batasan Masalah**

Agar tidak terjadi penyimpangan dalam proses penelitian dan pembangunan Sistem Penilaian Prestasi Kerja Guru Sebagai Luaran Pemilihan Calon Kepala Sekolah Menggunakan Metode *Profile Matching* - AHP (Studi Kasus : SMA Negeri Grujukan Bondowoso), maka ditetapkan beberapa batasan permasalahan sebagai berikut:

1. Kriteria yang diambil didasarkan dengan Peraturan Menteri Pendidikan Nasional Nomer 16 Tahun 2007 Tentang Standar Kualifikasi Akademik dan Kompetensi Guru dan Peraturan Menteri Pendidikan Nasional Nomor 28 Tahun 2010 Tentang Penugasan Guru Sebagai Kepala Sekolah/Madrasah
2. Uji coba dilakukan di SMA Negeri Grujukan.
3. Sistem ini menyeleksi siapa guru yang paling berhak mengajukan diri sebagai kandidat calon kepala sekolah.

4. Sistem ini menampilkan hasil perangkingan penilaian prestasi kerja guru secara keseluruhan pada SMA Negeri Grujugan
5. Bobot kriteria di hitung menggunakan metode *Profile Matching-Analytic Hierarchy Process* (AHP).

#### 1.4 Sistematika Penulisan

Adapun sistematika penulisan skripsi ini adalah sebagai berikut:

1. Pendahuluan

Bab ini memuat uraian tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika penulisan skripsi yang masing-masing diuraikan dalam subbab tersendiri.

2. Tinjauan Pustaka

Bab ini memaparkan tinjauan terhadap hasil-hasil penelitian terdahulu berkaitan dengan masalah yang dibahas, landasan materi dan konsep pengelompokan resiko kehamilan, dan kajian teori metode analisis data yang berkaitan dengan masalah dalam penelitian.

3. Metodologi Penelitian

Bab ini menguraikan tentang tempat dan waktu penelitian, metode penelitian, metode pengumpulan data, metode analisis data, dan teknik pengembangan sistem yang digunakan dalam penelitian.

4. Pengembangan Sistem

Bab ini berisi uraian tentang tentang proses pengembangan sistem.

Pengembangan sistem dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, kemudian merancang *business process*, *usecase diagram*, *scenario*, *activity diagram*, *sequence diagram*, *class diagram* dan *entity relationship diagram* (ERD), kemudian penulisan kode program, lalu proses pengujian sistem.

5. Hasil dan Pembahasan

Bab ini memaparkan secara rinci pemecahan masalah melalui analisis yang disajikan dalam bentuk deskripsi dibantu dengan ilustrasi berupa tabel dan gambar untuk memperjelas hasil penelitian.

#### 6. Penutup

Bab ini berisi kesimpulan penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya.

### 1.5 Sistematika Penulisan

Adapun sistematika penulisan skripsi ini adalah sebagai berikut:

#### 1. Pendahuluan

Bab pendahuluan adalah bab awal dari laporan tugas akhir. Pada bab ini membahas latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika penulisan.

#### 2. Tinjauan Pustaka

Bab tinjauan pustaka adalah tinjauan dari hasil penelitian terdahulu dengan masalah yang dibahas dalam penelitian, seperti landasan teori dan konsep paket produk.

#### 3. Metodologi Penelitian

Bab metodologi penelitian adalah pembahasan tempat dan waktu penelitian, metode penelitian, metode pengumpulan data, metode analisis data, dan teknik pengembangan sistem yang digunakan dalam penelitian.

#### 4. Pengembangan Sistem

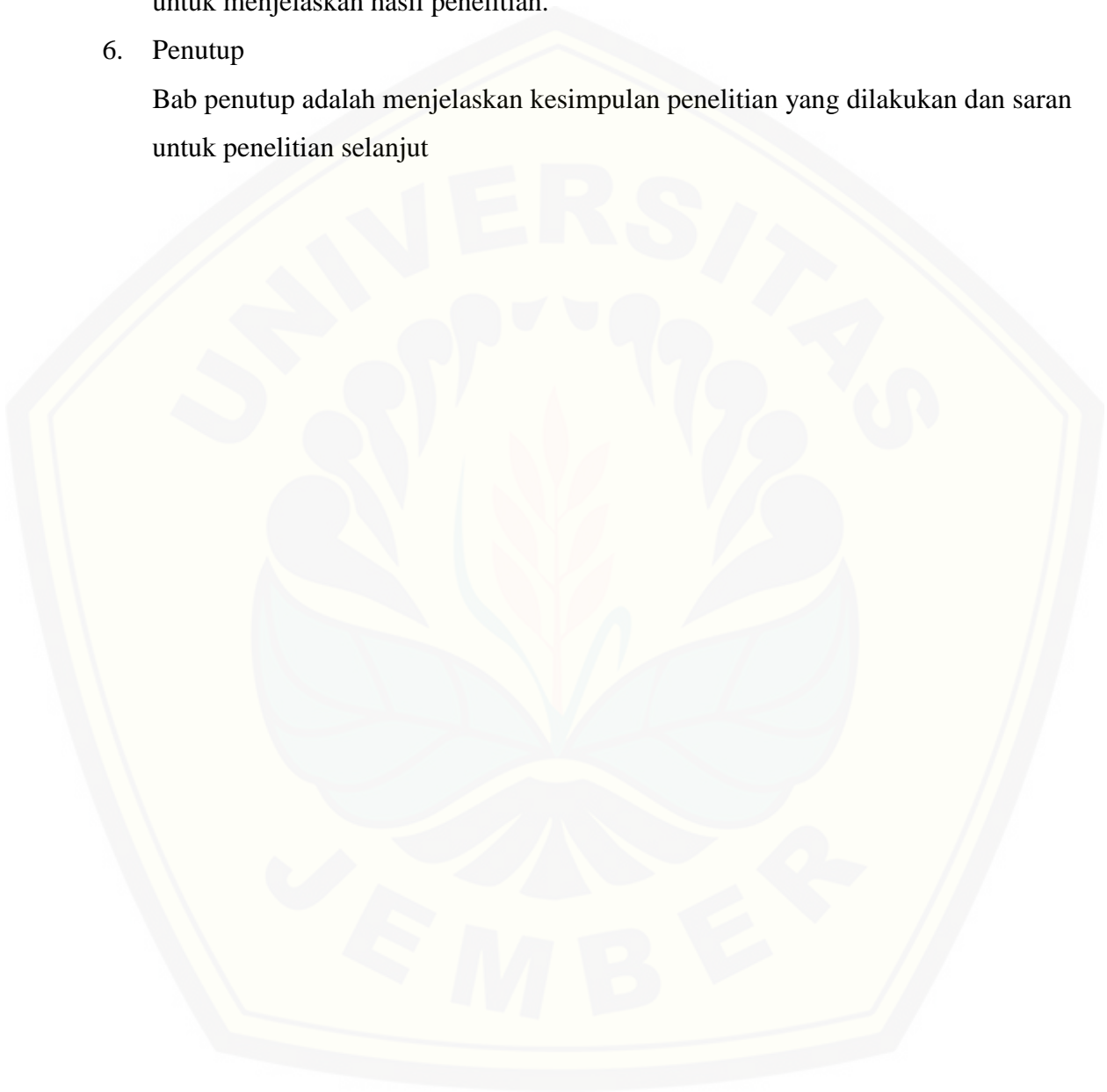
Bab pengembangan sistem adalah tentang desain sistem. Pengembangan sistem dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, kemudian merancang *business process*, *usecase diagram*, *scenario*, *activity diagram*, *sequence diagram*, *class diagram* dan *entity relationship diagram (ERD)*.

#### 5. Hasil dan Pembahasan

Bab hasil dan pembahasan adalah bab yang menjelaskan secara rinci pemecahan masalah melalui analisa yang disajikan dalam bentuk deskripsi, tabel dan gambar untuk menjelaskan hasil penelitian.

6. Penutup

Bab penutup adalah menjelaskan kesimpulan penelitian yang dilakukan dan saran untuk penelitian selanjut



## BAB 2. TINJAUAN PUSTAKA

Pada bagian ini dipaparkan tinjauan yang berkaitan dengan masalah yang dibahas, kajian teori yang berkaitan dengan masalah, dan juga penelitian-penelitian terdahulu.

### 2.1 Alur Pemilihan Kepala Sekolah

Pada (Peraturan Menteri Pendidikan Nasional, 2010) dijelaskan calon kandidat kepala sekolah di tiap sekolah di pilih atas pertimbangan Kepala Sekolah yang sedang menjabat pada periode itu melalui pertimbangan penilaian yang sesuai dengan Peraturan Menteri Pendidikan Nasional Nomer 16.

Guru akan mendapat penilaian prestasi kerja tiap tahun ajaran, yang sekaligus sebagai syarat pemilihan calon kepala sekolah pada masing-masing sekolah yang akan dilanjutkan ke tingkat kabupaten untuk diseleksi lebih lanjut.

Pada penilaian prestasi kerja guru, bagian administrasi di sekolah atau lebih dikenal dengan nama bagian tata usaha berperan sebagai aktor yang melakukan perhitungan berdasarkan hasil penilaian kepala sekolah dan beberapa syarat kriteria penilaian prestasi kerja guru. Hasil akhir yang didapat dari perhitungan oleh tata usaha akan disetujui oleh kepala sekolah dalam penentuan peringkat prestasi kerja guru yang dapat digunakan sebagai luaran pemilihan calon kepala sekolah pada sekolah tersebut.

### 2.2 Penelitian Terdahulu

Pada penelitian terdahulu pernah dibuat sistem dengan judul “*Sistem Pendukung Keputusan Penetapan Calon Peserta Sertifikasi Guru Sekolah Dasar Menggunakan Metode Analytical Hierarchy Process*” (Hanafi, Tyroni, Pranomo, 2011). Sistem itu berjalan pada SD yang berada dikecamatan Buduran, fungsinya yaitu untuk menerima usulan calon guru yang ingin mengajukan sertifikasi berdasarkan bobot pada masing-

masing kriteria dan subkriteria, dengan mengimplementasikan pembentukan nilai skala perbandingan matriks kriteria berdasarkan rekomendasi dari pimpinan instansi tempat penelitian. Metode AHP ini dapat diimplementasikan sebagai sistem pendukung keputusan multi kriteria. Pada sistem akan dihasilkan peringkat guru untuk dijadikan pendukung keputusan dalam pemilihan calon guru peserta sertifikasi.

Penelitian lainnya dengan judul “*Aplikasi Pendukung Pengambilan Keputusan Pemilihan Kepala Sekolah Menggunakan Metode Profile Matching*” (Ima, Ermanu, Husniah, 2011). Aplikasi ini digunakan untuk membantu penyeleksi dalam melakukan penilaian siapa guru yang berhak menjadi kepala sekolah. *Profile Matching* dapat membandingkan antara kompetensi yang dimiliki guru dengan kompetensi yang telah ditentukan oleh regulasi Peraturan Menteri No. 28 tahun 2010 tentang Pedoman dan Panduan Pelaksanaan Pengadaan Kepala Sekolah, sehingga diperoleh perbedaan kompetensi yang disebut dengan gap.

Penelitian selanjutnya dengan judul “*Sistem Pendukung Keputusan Penilaian Proposal Kegiatan PNPM MPd Menggunakan Metode Profile Matching dan Analytic Hierarchy Process*” (Mufid, 2014). Tujuan pembuatan sistem ini adalah membuat perangkat lunak untuk membantu tim verifikasi dalam melakukan penilaian proposal dalam kegiatan PNPM-MPd secara obyektif menggunakan metode *Profile Matching* dan *Analytic Hierarchy Process* (AHP). Penggabungan 2 metode ini dikarenakan profil matching memiliki kelemahan yaitu tidak mempunyai kemampuan untuk memecahkan masalah yang diteliti multi objek dan multi kriteria yang berdasar pada perbandingan preferensi dari tiap elemen dalam hierarki. Maka dari itu digabungkanlah dengan metode AHP yang dapat memberi bobot pada perbandingan matrik berpasangan pada masing-masing kriteria.

Berdasarkan penelitian tersebut dapat disimpulkan bahwa untuk merekap dan menghitung hasil penilaian sehingga dapat memutuskan calon peserta nama guru yang boleh mencalonkan diri sebagai calon kepala sekolah dari berbagai macam kriteria dan sub kriteria dengan masing-masing bobot perbandingan penilaian yang berbeda dibutuhkan metode *Analytic Hierarchy Process* (AHP), sedangkan untuk

membandingkan jarak selisih nilai ideal prestasi kinerja guru dengan masing-masing data nilai yang diperoleh oleh guru yaitu hasil kerja prestasi guru pribadi menggunakan metode *Profile Matching*.

### 2.3 *Profile Matching* (Pencocokan Profil)

Pencocokan profil (*Profile Matching*) adalah sebuah mekanisme pengambilan keputusan dengan mengansumsikan bahwa terdapat tingkat variabel prediktor yang ideal yang harus dimiliki oleh subyek, bukannya tingkat minimal yang harus dipenuhi atau dilewati (Kusrini, 2007). Dalam proses *Profile Matching*, akan dilakukan proses perbandingan antara kompetensi individu ke dalam kompetensi standar, dalam hal ini nilai pada masing-masing kriteria, sehingga dapat diketahui perbedaan kompetensinya (disebut juga gap). Semakin kecil gap yang dihasilkan maka bobot nilainya semakin besar. Calon yang memiliki bobot nilai yang besar berarti memiliki hak lebih besar untuk dapat mencalonkan diri sebagai kandidat calon kepala sekolah. Berikut merupakan langkah-langkah perhitungan dalam *Profile Matching* :

#### 1. Pemetaan Gap Kompetensi

Gap yang dimaksud di sini adalah perbedaan/selisih value masing-masing aspek/atribut dengan value target. Contoh: Perbedaan value capaian kriteria nilai guru dengan nilai sasaran capaian nilai guru yang didapat, dengan perhitungan pada persamaan 1 :

$$\text{Gap} = \text{Value Atribut} - \text{Value Target} \dots\dots\dots \text{persamaan 1}$$

#### 2. Pembobotan

Setelah diperoleh Gap pada masing-masing nilai capaian guru, setiap nilai diberi bobot nilai sesuai ketentuan, seperti contoh pada Tabel 2.1. Contoh Bobot Nilai Gap.

Tabel 2. 1 Contoh Bobot Nilai Gap

| No. | Selisih | Bobot Nilai | Keterangan   |
|-----|---------|-------------|--|
| 1   | 1       | 4           | Kompetensi individu kelebihan 1 tingkat/level                |
| 2   | 0       | 3           | Tidak ada selisih (kompetensi sesuai dengan yang dibutuhkan) |
| 3   | -1      | 2           | Kompetensi individu kekurangan 1 tingkat/level               |
| 4   | -2      | 1           | Kompetensi individu kekurangan 2 tingkat/level               |
| 5   | -3      | 0           | Kompetensi individu kekurangan 3 tingkat/level               |

#### 2.4 Metode Analytic Hierarchy Process (AHP)

AHP merupakan model yang dikembangkan oleh Thomas L. Saaty. AHP membantu dalam menentukan prioritas dari berbagai variabel dengan melakukan analisa perbandingan berpasangan dari masing-masing variabel menjadi suatu bentuk susunan hierarki, kemudian memberikan nilai numerik untuk penilaian subjektif terhadap kepentingan dari setiap variabel dan serta memberikan prioritas tertinggi pada masing-masing kriteria yang akan mempengaruhi penyelesaian dari situasi tersebut. AHP menggabungkan pertimbangan dan penilaian pribadi dengan cara yang logis dan dipengaruhi imajinasi, pengalaman, dan pengetahuan untuk menyusun hierarki dari suatu masalah yang berdasarkan logika, intuisi dan juga pengalaman untuk memberikan pertimbangan. Prosedur dalam menggunakan metode AHP terdiri dari beberapa tahap yaitu:

##### 1. Menyusun hirarki dari permasalahan yang dihadapi

Penyusunan hirarki yaitu dengan menentukan tujuan yang merupakan sasaran sistem secara keseluruhan pada level teratas. Level berikutnya terdiri dari kriteria-kriteria untuk menilai atau mempertimbangkan alternatif-alternatif yang ada dan menentukan alternatif-alternatif tersebut. Setiap kriteria dapat memiliki subkriteria dibawahnya dan setiap kriteria dapat memiliki nilai intensitas masing-masing.



2. Menentukan prioritas elemen dengan langkah-langkah sebagai berikut:

1. Membuat perbandingan berpasangan

Langkah pertama dalam menentukan prioritas elemen adalah membuat perbandingan berpasangan, yaitu membandingkan elemen secara berpasangan sesuai kriteria yang di berikan. Untuk perbandingan berpasangan digunakan bentuk matriks. Untuk memulai proses perbandingan berpasangan, dimulai dari level paling atas hirarki untuk memilih kriteria, misalnya C, kemudian dari level dibawahnya diambil elemen-elemen yang akan dibandingkan, misal A1, A2, A3, A4, A5, maka susunan elemen-elemen pada sebuah matrik seperti Tabel 2.2

Tabel 2. 2 Matrik Perbandingan Berpasangan

| C  | A1 | A2 | A3 | A4 | A5 |
|----|----|----|----|----|----|
| A1 | 1  |    |    |    |    |
| A2 |    | 1  |    |    |    |
| A3 |    |    | 1  |    |    |
| A4 |    |    |    | 1  |    |
| A5 |    |    |    |    | 1  |

2. Mengisi Matrik Perbandingan Berpasangan

Untuk mengisi matrik perbandingan berpasangan yaitu dengan menggunakan bilangan untuk merepresentasikan kepentingan relatif dari satu elemen terhadap elemen lainnya yang dimaksud dalam bentuk skala dari 1 sampai dengan 9. Skala ini mendefinisikan dan menjelaskan nilai 1 sampai 9 untuk pertimbangan dalam perbandingan berpasangan elemen pada setiap level hirarki terhadap suatu kreteria di level yang lebih tinggi. Apabila suatu elemen dalam matrik dan dibandingkan dengan dirinya sendiri, maka diberi nilai 1. Jika i dibanding j mendapatkan nilai tertentu, maka j dibanding i merupakan kebalikkannya. Pada

tabel 2.3 memberikan definisi dan penjelasan skala kuantitatif 1 sampai dengan 9 untuk menilai tingkat kepentingan suatu elemen dengan elemen lainnya.

Tabel 2. 3 Skala Prioritas Kriteria

| Nilai   | Tingkat Prioritas  |
|---------|--|
| 1       | <b>KRITERIA A</b> sama penting dibanding dengan <b>KRITERIA B</b>          |
| 3       | <b>KRITERIA A</b> sedikit lebih penting dibanding dengan <b>KRITERIA B</b> |
| 5       | <b>KRITERIA A</b> lebih penting dibanding dengan <b>KRITERIA B</b>         |
| 7       | <b>KRITERIA A</b> sangat penting dibanding dengan <b>KRITERIA B</b>        |
| 9       | <b>KRITERIA A</b> jauh sangat penting dibanding dengan <b>KRITERIA B</b>   |
| 2,4,6,8 | Nilai diantara dua penilaian yang berdekatan                               |

### 3. Sintesis

Pertimbangan-pertimbangan terhadap perbandingan berpasangan di sintesis untuk memperoleh keseluruhan prioritas dengan langkah-langkah sebagai berikut:

1. Menjumlahkan nilai-nilai dari setiap kolom pada matriks.
2. Membagi setiap nilai dari kolom dengan total kolom yang bersangkutan untuk memperoleh normalisasi matriks.
3. Menghitung Eigen Vektor dengan menjumlahkan nilai-nilai dari setiap matriks dan membaginya dengan jumlah elemen untuk mendapatkan nilai rata-rata.
4. Menghitung Bobot Prioritas dengan membagi nilai Eigen Vektor dengan total Eigen Vektor pada semua kriteria
5. Mengukur konsistensi dalam pembuat keputusan merupakan hal yang penting. Konsistensi penting untuk mendapatkan hasil yang valid dalam dunia nyata. AHP mengukur konsistensi pertimbangan dengan rasio konsistensi (*consistency ratio*). Langkah-langkah menghitung nilai rasio konsistensi yaitu:

- a. Mengkalikan nilai pada kolom pertama dengan prioritas relatif elemen pertama, nilai pada kolom kedua dengan prioritas relatif elemen kedua, dan seterusnya.
- b. Menjumlahkan setiap baris.
- c. Hasil dari penjumlahan baris dibagikan dengan elemen prioritas relatif yang bersangkutan.
- d. Membagi hasil diatas dengan banyak elemen yang ada, hasilnya disebut eigen value ( $\lambda_{max}$ ).
- e. Menghitung indeks konsistensi (consistency index) dengan persamaan 2 :

$$CI = \frac{\lambda_{maksimum} - n}{n - 1} \dots \dots \dots \text{persamaan 2}$$

Keterangan :

CI : Consistensi Index

$\lambda$  maksimum : Eigen Value

n : Banyak elemen kriteria

- f. Menghitung konsistensi ratio (CR) dengan persamaan 3 :

$$CR = \frac{CI}{RC} \dots \dots \dots \text{persamaan 3}$$

Keterangan :

CR : Consistency Ratio

CI : Consistency Index

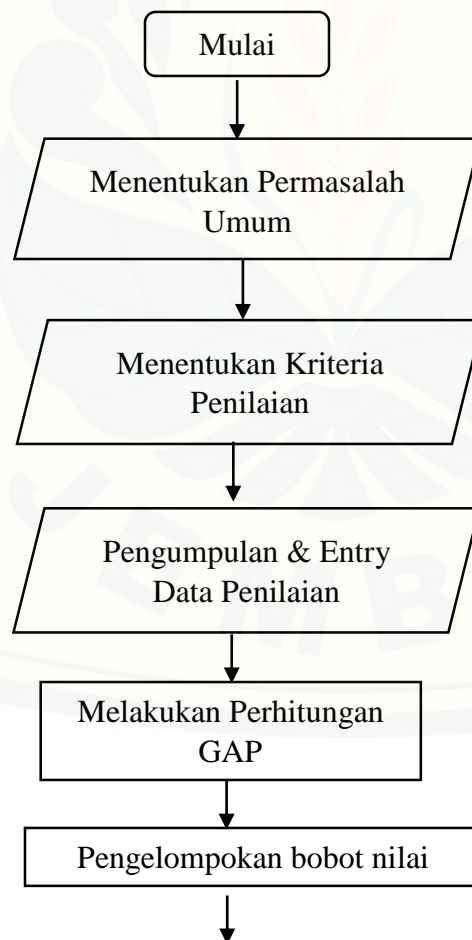
RC : Random Consistency

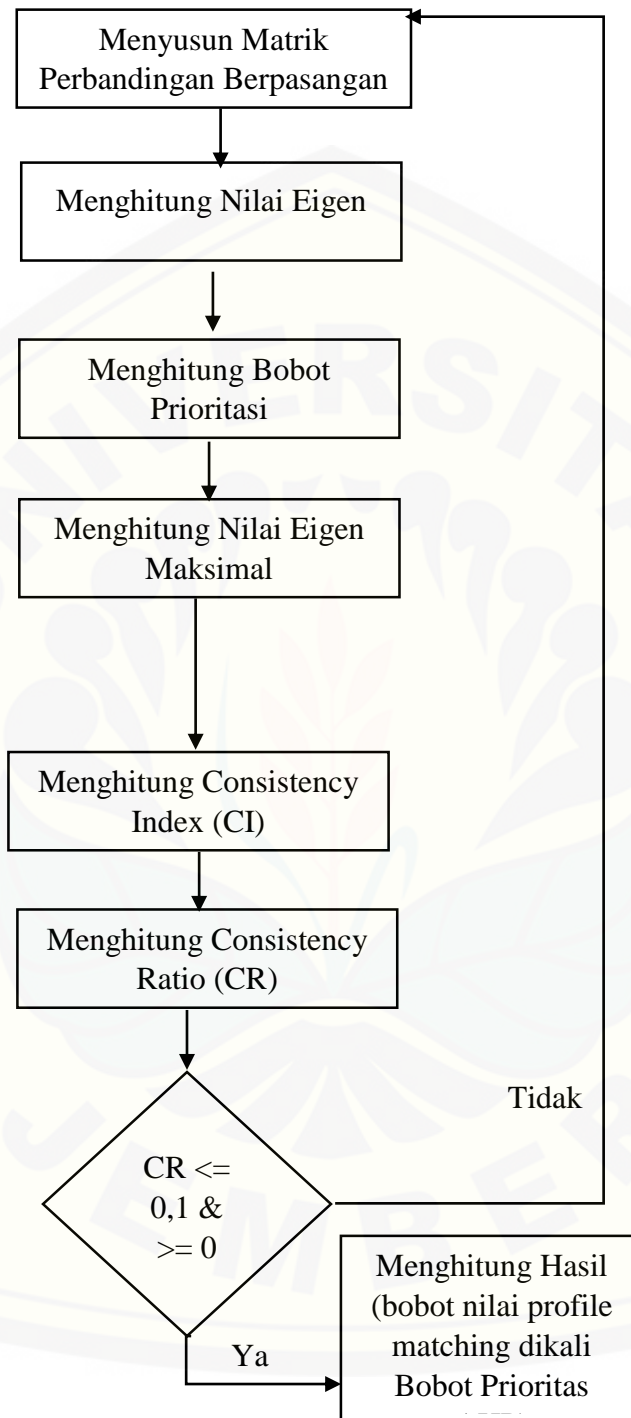
Matriks random dengan skala penilaian 1 sampai 9 beserta kebalikkannya sebagai random consistency (RC). Berdasarkan perhitungan saaty dengan menggunakan 500 sampel, jika pertimbangan memilih secara acak dari skala 1/9, 1/8, ... , 1, 2, ... , 9 akan diperoleh ratarata konsistensi untuk matriks yang berbeda seperti pada Tabel 2.4.

Tabel 2. 4 Nilai Rata-Rata Konsistensi

| Ukuran Matriks | Konsistensi acak<br>( <i>Random Consistency</i> ) |
|----------------|---|
| 1              | 0,00  |
| 2              | 0,00  |
| 3              | 0,58  |
| 4              | 0,90  |
| 5              | 1,12  |
| 6              | 1,24  |
| 7              | 1,32  |
| 8              | 1,41  |
| 9              | 1,45  |
| 10             | 1,49  |

### 3.4 Penerapan *Profile Matching* dan AHP



Gambar 2. 1 Penerapan *Profile Matching-AHP*

### **2.5 Sistem Penunjang Keputusan**

Sistem Penunjang Keputusan adalah Sistem yang interaktif, membantu pengambilan keputusan melalui penggunaan data dan model-model keputusan untuk memecahkan masalah-masalah yang sifatnya semi terstruktur dan tidak terstruktur . Dapat juga dikatakan sebagai sistem komputer yang mengolah data menjadi informasi untuk mengambil keputusan dari masalah semi-terstruktur yang spesifik (Jogiyanto, 2004). Menurut Moore and Chang, SPK dapat digambarkan sebagai sistem yang berkemampuan mendukung analisis data, dan pemodelan keputusan, berorientasi keputusan, orientasi perencanaan masa depan, dan digunakan pada saat-saat yang tidak biasa.

## BAB 3 METODE PENELITIAN

Bab ini menjelaskan tentang metode yang digunakan dalam merancang dan membangun sistem yaitu metode pengembangan.

### 3.1 Jenis Penelitian

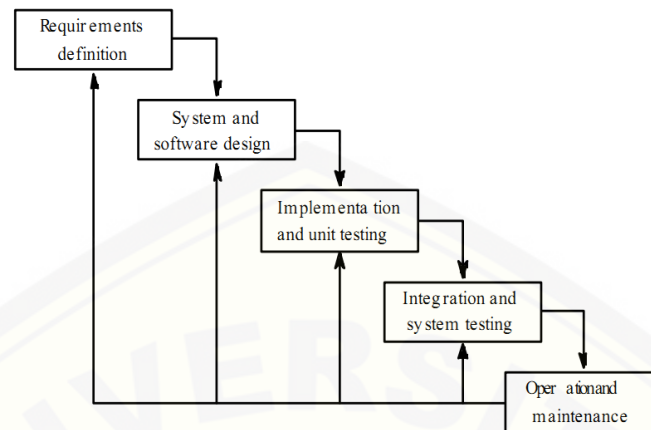
Jenis penelitian yang digunakan pada penelitian ini merupakan jenis penelitian pengembangan, karena tujuan penelitian adalah untuk membangun sebuah Sistem Penilaian Prestasi Kerja Guru Sebagai Luaran Pemilihan Calon Kepala Sekolah. Penelitian pengembangan bertujuan untuk membuat dan mengembangkan suatu produk yang efektif untuk digunakan. Penelitian ini bukan jenis penelitian yang ditunjukkan untuk menemukan teori atau menguji kebenaran dari suatu teori dalam bentuk eksperimentasi.

### 3.2 Tempat dan Waktu Penelitian

Penelitian dilakukan di SMA Negeri Grujungan Bondowoso. Waktu penelitian dilakukan selama 9 (sembilan), dimulai pada bulan September 2016 sampai dengan bulan Mei 2017.

### 3.3 Teknik Penyajian dan Analisis Data

Teknik penyajian dan analisis data yang digunakan menggunakan model implementasi sistem *waterfall*. Model *waterfall* merupakan model yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem sampai pada analisis, desain, kode, test dan pemeliharaan (Pressman, 2002). Berikut ini adalah tahapan *Waterfall* pada gambar 3.1:



Gambar 3. 1 Tahapan Waterfall

Langkah-langkah pada model *Waterfall* sebagai berikut :

### 3.3.1 Analisis Kebutuhan (*Requirments Definition*)

Pada tahap ini, ditentukan berbagai persyaratan atau kebutuhan yang telah dianalisis sebelum Analisis kebutuhan merupakan tahap untuk mengumpulkan data, informasi, serta mencari kebutuhan fungsional dan non fungsional sistem. Pada tahap ini, peneliti mencari permasalahan yang ada untuk dapat dianalisis kebutuhan yang diperlukan, sebagai solusi dari permasalahan yang muncul. Data-data yang telah didapat kemudian dikelompokkan menjadi kebutuhan fungsional dan non-fungsional sistem. Metode yang digunakan adalah sebagai berikut:

#### a. Metode Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam mencapai tujuan penelitian. Pengumpulan data pada penelitian ini dilakukan melalui beberapa teknik pengumpulan data yaitu:

1. Melakukan studi literatur, jurnal, serta peraturan perundang-undangan menteri tentang persyaratan penugasan guru sebagai kepala sekolah
2. Melakukan wawancara dari Kepala Sekolah di SMA Negeri Grugugan terkait peraturan persyaratan penugasan guru sebagai kepala sekolah dan wawancara



dari bagian tata usaha yang bertugas sebagai pengelola laporan penilaian prestasi kerja guru.

b. Metode Analisis Data

Tahapan analisis data dimulai ketika data telah terkumpul sehingga dapat diperoleh permasalahan yang ada. Dari hasil analisis tersebut, maka metode *Profile Matching* – AHP digunakan sebagai dasar perhitungan penentuan peringkat prestasi kerja guru sebagai luaran calon kandidat kepala sekolah.

### 3.3.2 Desain Sistem (*System and Software Design*)

Pembuatan desain sistem menggunakan *Unified Modeling Language (UML)* yang dirancang menggunakan konsep *Object-Oriented Programming (OOP)*. Pemodelan UML yang digunakan antara lain :

a. *Business Process*

*Business Process* dalam perancangan sistem ini digunakan untuk menggambarkan seluruh proses yang dibutuhkan hingga proses goal tercapai.

b. *Use Case Scenario*

*Use case scenario* merupakan penjelasan alur sistem sesuai dengan yang tertera pada *use case diagram*, skenario juga menjelaskan reaksi yang akan terjadi pada sistem setelah menerima perlakuan dari aktor. *Use case scenario* menggambarkan keadaan normal saat setiap aktor mengakses aplikasi dan keadaan alternatif yang terjadi pada kondisi tertentu.

c. *Sequence Diagram*

*Sequence diagram* merupakan diagram yang digunakan untuk menggambarkan rangkaian interaksi yang dikirim antar object dan interaksi antar object. Interaksi yang terjadi di dalam sistem sesuai dengan urutan dijalankannya sistem tersebut.

d. *Activity Diagram*

*Activity diagram* menggambarkan alur aktivitas dalam sistem yang sedang dirancang, meliputi awal alur dimulai, *decision* yang terjadi, dan bagaimana alur

berakhir yang dapat dilakukan oleh masing-masing aktor. *Activity diagram* juga menggambarkan aktivitas sistem setelah menerima perlakuan dari aktor tersebut.

e. *Class Diagram*

*Class diagram* merupakan sebuah spesifikasi yang menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan kelas, *function*, dan atribut yang dibutuhkan oleh seorang programmer dalam membangun sistem.

f. *Entity Relationship Diagram* (ERD)

*Entity relationship diagram* merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.

### 3.3.3 Implementasi dan Pengujian Unit (*Implementation and Unit Testing*)

Pada tahap ini, ditentukan pencodingan dalam pembuatan sistem berbasis web menggunakan framework CodeIgniter dan basis data menggunakan MySQL dengan memanfaatkan aplikasi XAMPP.

### 3.3.4 Integritas dan Pengujian Sistem (*Integration and System Testing*)

Pada tahap ini dilakukan uji coba sistem yang telah dibuat dengan pengujian *white box* dan *black box*.

1. *White Box Testing* merupakan cara pengujian dengan melihat modul untuk yang telah dibuat dengan program – program yang ada. Pengujian ini dilakukan oleh (*developer*) pembuat program. Pengujian *white box* dilakukan dengan menggunakan metode siklomatik. Metode siklomatik menggunakan pengujian jalur dasar (*basis path testing*) yang didalamnya terdapat beberapa tahapan pengujian antara lain pembuatan diagram alir atau grafik alir, penentuan jalur independen, penghitungan kompleksitas siklomatik jalur independen dan *test case*. Jika ada modul yang menghasilkan output yang

tidak sesuai, maka baris-baris program, variabel dan parameter yang terlibat pada unit tersebut satu persatu akan di cek dan diperbaiki, kemudian di *compile* ulang

2. *Black Box Testing* adalah metode pengujian perangkat lunak yang memeriksa fungsionalitas dari aplikasi yang berkaitan dengan struktur internal atau kerja. Pengetahuan khusus dari kode aplikasi, struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Metode ini memfokuskan pada keperluan fungsionalitas dari *software*. (Agissa, 2013)

### 3.3.5 Pemeliharaan (Operational Maintance)

Pada tahap ini, ditentukan apakah terdapat kesalahan-kesalahan pada sistem yang telah dibuat. Jika terdapat kesalahan, maka sistem akan diubah sehingga kesalahan tersebut dapat teratasi. Pemeliharaan dilakukan sesuai dengan kebutuhan user

## BAB 4 PENGEBANGAN SISTEM

Bab ini menguraikan mengenai analisis kebutuhan, desain, implementasi, dan pengujian sistem yang digunakan dalam proses pengembangan sistem penilaian prestasi kerja guru sebagai luaran pemilihan calon kepala sekolah menggunakan metode *Profile Matching* - AHP. Tahapan analisis hingga pengujian dilakukan sesuai dengan model *waterfall*.

### 4.1 Analisis Kebutuhan

Berdasarkan pengembangan sistem model *waterfall*, tahapan awal yang dilakukan adalah tahap analisis. Tahap analisis ini dilakukan terhadap objek penelitian untuk memperoleh kebutuhan sistem yang dibangun, berupa kebutuhan fungsional dan nonfungsional. Dimana hasil analisa tersebut sangat mempengaruhi fungsionalitas sistem yang dibangun agar dapat digunakan sesuai fungsi dan kebutuhan pengguna.

#### 4.1.1 SOP (*Statement of purpose*)

Sistem Penilaian Prestasi Kerja Guru Sebagai Keluaran Pemilihan Calon Kepala Sekolah Menggunakan Metode *Profile Matching*-AHP ini dibangun untuk mempermudah tugas kepala sekolah selaku penilai kinerja guru di SMA Negeri Grujungan sebagai keputusan menentukan guru yang berhak mencalonkan diri menjadi kandidat calon kepala sekolah tanpa harus melakukan perhitungan yang rumit dan mengurangi rentang kesalahan dalam penilaiannya subyektif berdasarkan ketetapan perundang-undangan kementerian pendidikan.

Sistem ini akan diaplikasikan di SMA Negeri Grujungan sesuai data nilai guru yang didapat pada masing-masing kriteria penilaian kinerja guru yang nanti akan dihitung menggunakan *Profile Matching* dan AHP. Aktor yang menjalankan sistem ini ada tiga aktor, yaitu kepala sekolah, admin yang disini merupakan bagian tata usaha di

sekolah SMA Negeri Grujukan dan guru. Fitur yang ditawarkan sistem ini adalah manajemen data guru, manajemen tambah nilai guru, manajemen data user, manajemen data pribadi guru, manajemen nilai ideal *Profile Matching*, manajemen bobot kriteria AHP, manajemen verifikasi peringkat nilai kinerja, dan manajemen update rekap hasil peringkat. Output yang akan dihasilkan adalah menyeleksi nama guru yang telah berhak untuk bisa mencalonkan diri sebagai kandidat calon kepala sekolah dengan peringkat nilai tertinggi sampai terendah yang didapat berdasarkan hasil penilaian prestasi kerja guru. Sistem ini berbasis web dikarenakan bagian tata usaha di SMA Negeri Grujukan yang bertugas untuk menghitung keseluruhan nilai yang didapat semua guru, dengan arahan dari atasan yaitu Kepala Sekolah. Sehingga dalam proses komunikasi lebih mudah menggunakan web.

#### 4.1.2 Kebutuhan Fungsional

Kebutuhan fungsional sistem berisi fitur-fitur inti yang harus dipenuhi dalam sistem agar sistem berfungsi sesuai dengan tujuan dan kebutuhan pengguna terhadap sistem. Kebutuhan fungsional dari sistem penilaian prestasi kerja guru sebagai luaran pemilihan calon kepala sekolah menggunakan metode *Profile Matching* – AHP yaitu :

1. Sistem mampu melakukan perhitungan nilai yang didapat dari data pribadi masing-masing guru dengan membandingkan nilai ideal syarat menjadi calon kepala sekolah menggunakan metode *Profile Matching*.
2. Sistem mampu membandingkan tingkat prioritas bobot kriteria satu dengan yang lain dan melakukan perbandingan nilai yang didapat guru menggunakan metode AHP.
3. Sistem mampu mengelola data profil guru yang sekaligus berguna sebagai penilaian kinerja guru (input, edit, view).
4. Sistem mampu melakukan penilaian tambahan untuk tahun ajaran selanjutnya tanpa melakukan penghapusan penilaian di tahun ajaran sebelumnya (input, view)

5. Sistem mampu mengelolah status aktif nonaktif guru dan reset password bagi data pengguna user (update).
6. Sistem mampu mengelola profil pribadi masing-masing yang telah diinputkan admin sebelumnya (edit, view).
7. Sistem mampu untuk mengganti password default masing-masing user saat admin mendaftarkan user pengguna di awal kerja sistem (input, view)
8. Sistem mampu mengganti status verifikasi dari hasil peringkat nilai kinerja guru yang nantinya peringkat kinerja tersebut akan diteruskan kepada masing-masing guru (edit, view)
9. Sistem yang mampu membuat rekap hasil penilaian prestasi kerja guru tiap tahun ajaran (update, view)

#### **4.1.3 Kebutuhan Non-Fungsional**

Kebutuhan non-fungsional merupakan fitur-fitur yang dimiliki untuk mendukung sistem dalam memenuhi fungsionalitasnya. Kebutuhan non-fungsional dari sistem penunjang keputusan pemilihan calon kepala sekolah berdasarkan penilaian kinerja guru menggunakan metode *Profile Matching* – AHP yaitu:

1. Sistem memiliki tampilan dan bahasa komunikasi sistem yang mudah dimengerti oleh pengguna agar memberikan kenyamanan pemakaian dan memudahkan pengoperasian.
2. Sistem dapat digunakan oleh banyak pengguna pada saat yang bersamaan pada komputer yang berbeda.

#### **4.1.4 Fungsi Sistem**

Fungsi utama dari sistem yang dibangun dalam penelitian ini terletak pada fitur login sesuai dengan hak akses dari setiap pengguna dalam sistem. Ketika pengguna melakukan login, maka sistem melakukan autentifikasi username dan password

pengguna. Selanjutnya sistem akan menyajikan tampilan sistem sesuai dengan hak akses pengguna yang meliputi:

#### 1. Admin

Admin merupakan pengguna sistem yang telah terdaftar, dan ketika login berhasil maka sistem akan menampilkan halaman dashboard dan halaman-halaman yang sesuai dengan level admin yaitu level satu. Admin dapat mengakses halaman data guru (sebagai inputan data profil pribadi masing-masing guru yang nanti akan digunakan sebagai penilaian kinerja sekaligus user login), perhitungan profil (hasil dari perhitungan nilai kinerja dari metode *Profile Matching*), perhitungan AHP (menu untuk melakukan pembobotan masing-masing kriteria penilaian kinerja), hasil peringkat yang berisi peringkat data nilai kinerja guru dari tertinggi sampai terendah sekaligus sebagai menu khusus untuk melakukan update rekap penilaian tiap tahun ajaran yang telah di verifikasi oleh Kepala Sekolah agar data dapat tersimpan di menu rekap. Admin bertanggung jawab atas semua pengelolaan data yang dibutuhkan

#### 2. Kepala Sekolah

Kepala Sekolah merupakan pengguna sistem yang telah terdaftar oleh admin berdasarkan level pada saat input data guru, dan ketika login berhasil maka sistem akan menampilkan halaman dashboard dan halaman-halaman yang sesuai dengan level kepala sekolah yaitu level dua. Kepala Sekolah dapat mengakses menu nilai kinerja yang berisi nilai pribadi dari nilai kinerja yang dia dapat, kemudian dapat juga mengakses peringkat nilai kinerja yang berisi hasil peringkat dan detail nilai masing-masing guru, disana terdapat fitur verifikasi agar nilai peringkat dari semua guru dapat tampil dihalaman sistem masing-masing user guru saat login, biodata pribadi (berisi biodata pribadi kepala sekolah), kriteria penilaian, perhitungan profile, dan perhitungan AHP. Status kepala sekolah bisa diganti oleh admin

### 3. Guru

Guru merupakan pengguna sistem yang telah terdaftar oleh admin berdasarkan level pada saat input data guru, dan ketika login berhasil maka sistem akan menampilkan halaman dashboard dan halaman-halaman yang sesuai dengan level kepala sekolah yaitu level tiga. Guru dapat mengakses halaman nilai kinerja guru (untuk melihat nilai masing-masing), halaman peringkat nilai kinerja guru (melihat hasil peringkat seluruh guru di sekolah termasuk pribadi), dan biodata pribadi.

## 4.2 Desain Sistem

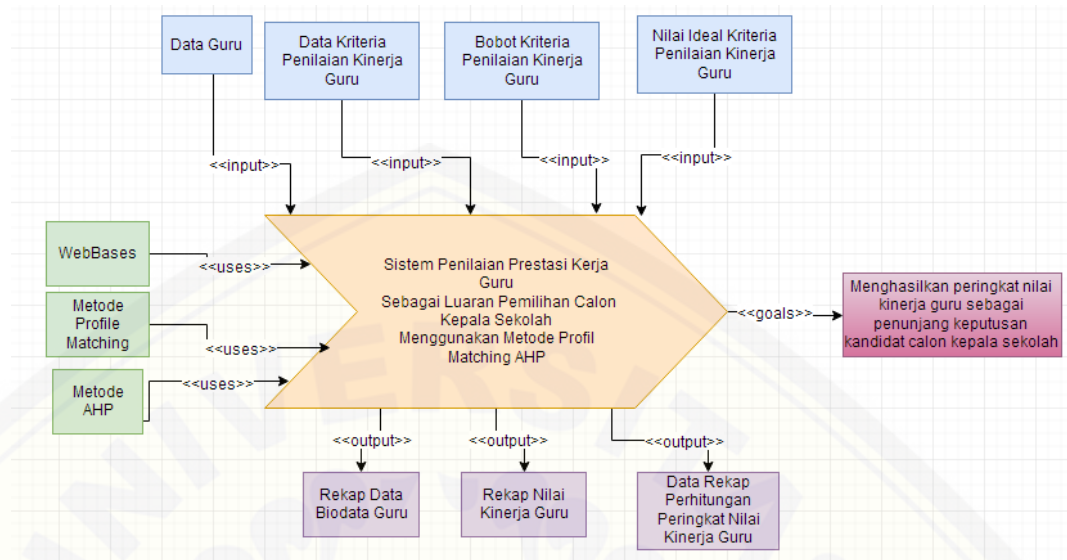
Tahapan yang dilakukan setelah melakukan analisis kebutuhan sistem yaitu tahap perencanaan pembangunan sistem yang digambarkan dengan desain sistem. Desain sistem penunjang keputusan pemilihan calon kepala sekolah berdasarkan penilaian kinerja guru menggunakan metode *Profile Matching - AHP* ini meliputi *use case diagram*, *use case skenario*, *activity diagram*, *sequence diagram*, *class diagram*, dan *entity relationship diagram*.

### 4.2.1 Business Process

*Business process* merupakan diagram yang menggambarkan kebutuhan data yang dibutuhkan oleh sistem. *Business process* didalamnya terdiri atas beberapa poin diantaranya:

1. *Input* : Data yang dimasukkan ke dalam sistem
2. *Output* : Data yang dihasilkan oleh sistem
3. *Goal* : Tujuan dibangun suatu sistem
4. *Uses* : *Platform* yang menjadi basis sistem
5. *Process* : Sistem yang bekerja





Gambar 4. 1 Bussines Procces Sistem Penilaian Pretasi Kerja Guru Sebagai Luaran Pemilihan Calon Kepala Sekolah

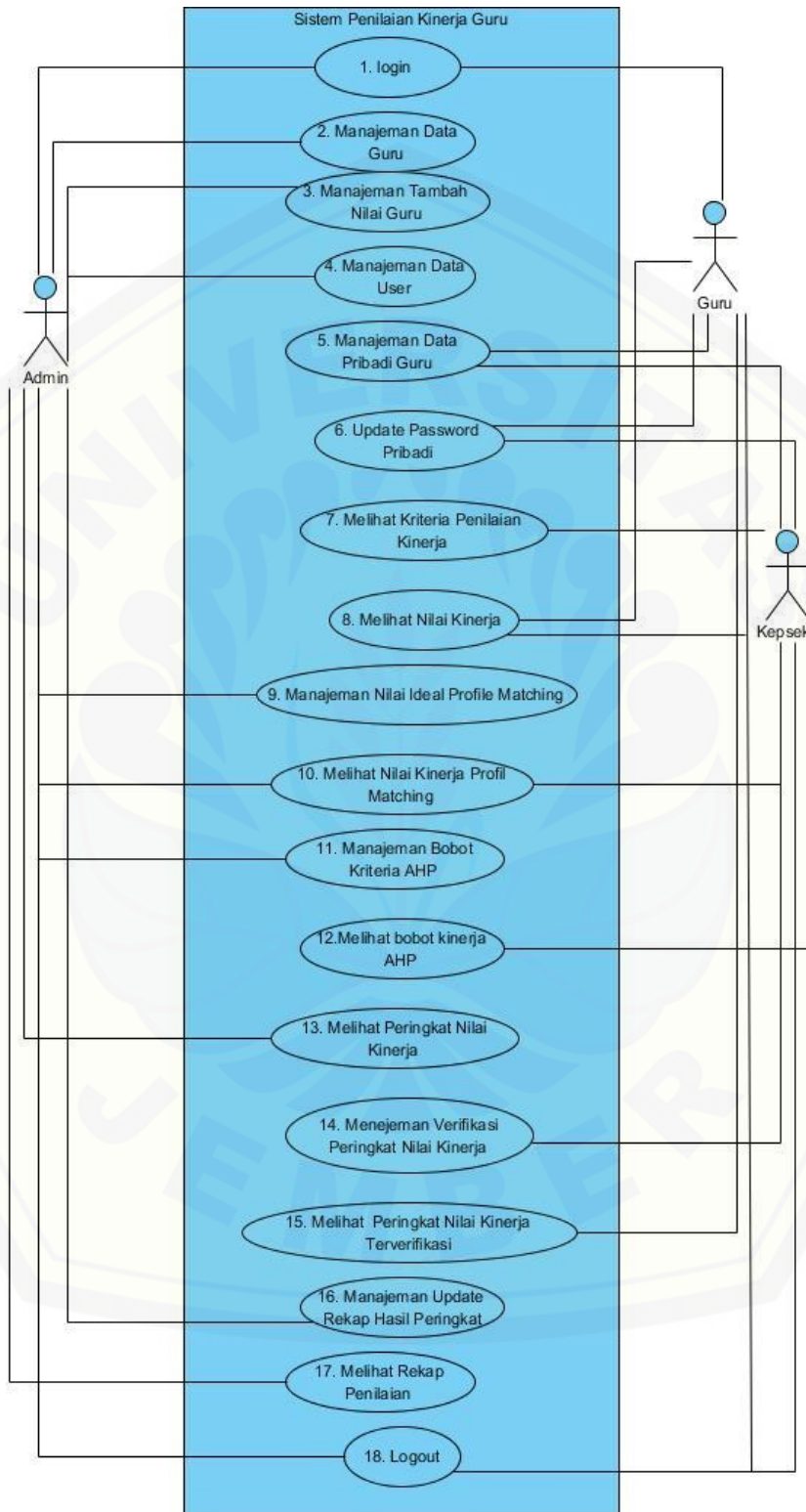
Gambar 4.1 ini merupakan *business process* dari sistem penilaian prestasi kerja guru sebagai luaran pemilihan calon kepala sekolah menggunakan metode *Profile Matching – AHP*. *Business process* menjelaskan proses *input*, *output*, *goal* dan *uses* yang diaplikasikan pada sistem. *Business process* menjelaskan proses *input*, *output*, *goal* dan *uses* yang diaplikasikan pada sistem. *Input* berupa data guru, Data kriteria penilaian kinerja guru, bobot kriteria penilaian kinerja guru, nilai ideal kriteria penilaian kinerja guru. *Output* berupa rekap data biodata guru, rekap nilai kinerja guru, data rekap perhitungan peringkat nilai kinerja guru. *Uses* yang dipakai adalah *web based*, metode *profile mathcing*, dan metode *AHP*. *Goal* pembuatan sistem adalah menghasilkan peringkat nilai kinerja guru sebagai penunjang keputusan kandidat calon kepala sekolah di SMA Negeri Grugugan.

#### 4.2.2 Usecase Diagram

*Usecase diagram* merupakan pemodelan yang digunakan untuk menggambarkan interaksi antara aktor dengan sistem yang dibangun. Melalui *usecase diagram* dapat diketahui interaksi yang dapat dilakukan aktor sesuai dengan hak akses

yang dimiliki oleh masing-masing aktor atau pengguna. *Usecase* diagram ditunjukkan pada Gambar 4.2





Gambar 4. 2 sistem penilaian prestasi kerja guru sebagai luaran pemilihan calon kepala sekolah

Gambar 4.2 menunjukkan *usecase diagram* sistem penilaian prestasi kerja guru sebagai luaran pemilihan calon kepala sekolah yang terdiri dari tiga aktor dengan delapan belas *usecase*.

Berdasarkan *usecase diagram* terdapat tiga aktor atau pengguna, yaitu Admin, Kepala Sekolah, dan Guru. Adapun deskripsi dari masing-masing aktor dapat dilihat pada Tabel 4.1

Tabel 4. 1 Deskripsi Aktor

| No | Aktor          | Deskripsi  |
|----|----------------|--|
| 1  | Admin          | Aktor yang memiliki hak penuh terhadap perhitungan peringkat kinerja guru, manajemen data guru, manajemen tambah nilai guru, manajemen data user, manajemen nilai ideal profil matching, melihat nilai kinerja profil matching, manajemen bobot kinerja AHP, dan melihat hasil peringkat kinerja guru baik belum dan sesudah terverifikasi, update rekap hasil peringkat, dan melihat rekap penilaian guru |
| 2  | Kepala Sekolah | Aktor yang memiliki hak melihat biodata pribadi, melihat nilai kerja pribadi, update password pribadi, melihat kriteria penilaian kinerja guru, melihat nilai kinerja hasil perhitungan profil matching, melihat nilai kinerja hasil perhitungan AHP, melihat hasil peringkat kinerja guru, dan manajemen verifikasi peringkat kinerja guru  |

|   |      |  |
|---|------|--|
| 3 | Guru | Melihat biodata pribadi, melihat nilai kinerja guru pribadi, update password pribadi, dan melihat hasil peringkat kinerja guru terverifikasi |
|---|------|--|

Selain memiliki tiga aktor, dalam Usecase diagram juga terdapat delapan belas *Usecase*. Deskripsi dari masing – masing *Usecase* pada sistem penilaian prestasi kinerja guru dapat di lihat pada Tabel 4.2

Tabel 4. 2 Deskripsi Usecase

| No | Usecase                     | Penjelasan  |
|----|-----------------------------|---|
| 1  | Login                       | Fitur yang berfungsi melakukan autentifikasi hak akses aktor dalam menggunakan sistem   |
| 2  | Manajemen Data Guru         | Fitur yang dapat diakses oleh admin yang berfungsi untuk manajemen (input, edit, view) data profil guru yang sekaligus berguna sebagai penilaian kinerja guru |
| 3  | Manajemen Data User         | Fitur yang dapat diakses oleh admin yang berfungsi untuk manajemen (update) status aktif nonaktif guru dan reset password data pengguna user                  |
| 4  | Manajemen Tambah Nilai Guru | Fitur yang dapat diakses oleh admin yang berfungsi untuk manajemen (input) nilai kinerja masing-masing guru pada periode penilaian tahun ajaran selanjutnya   |
| 5  | Manajemen Data Pribadi Guru | Fitur yang diakses oleh guru dan kepala sekolah yang berfungsi untuk manajemen (edit, view)   |

|    |   |  |
|----|---|--|
|    |   | profil pribadi masing-masing yang telah diinputkan admin sebelumnya  |
| 6  | Update Password Pribadi                       | Fitur yang diakses oleh guru dan kepala sekolah yang berfungsi untuk mengganti password default masing-masing saat admin mendaftarkan di awal  |
| 7  | Melihat Kriteria Penilaian Kinerja Guru       | Fitur yang diakses oleh kepala sekolah untuk melihat macam kriteria dari penilaian kinerja guru  |
| 8  | Melihat Nilai Kinerja                         | Fitur yang dapat diakses oleh guru dan kepala sekolah yang berfungsi melihat nilai kinerja pribadi yang didapat  |
| 9  | Manajemen Nilai Ideal <i>Profile Matching</i> | Fitur yang di akses admin yang berfungsi untuk manajemen (edit, view) nilai ideal pada masing-masing kriteria pada perhitungan <i>Profile Matching</i>   |
| 10 | Melihat Nilai Kinerja Profil Matching         | Fitur yang dapat diakses oleh admindan kepala sekolah untuk melihat hasil proses perhitungan sistem menggunakan metode profil matching dari hasil nilai kinerja yang telah diinputkan pada data guru |
| 11 | Manajemen Bobot Kriteria AHP                  | Fitur yang dapat diakses oleh admin yang berfungsi untuk manajemen (input, edit, view) masing-masing bobot dari kriteria pada perhitungan AHP  |

|    |   |   |
|----|---|---|
| 12 | Melihat Bobot Kriteria AHP                    | Fitur yang dapat diakses oleh kepala sekolah untuk dapat melihat bobot kriteria penilaian guru yang didapat dari perhitungan AHP  |
| 13 | Melihat Peringkat Nilai Kinerja               | Fitur yang dapat diakses oleh admin dan kepala sekolah untuk melihat hasil urutan peringkat tertinggi hingga terendah berdasarkan hasil proses penggabungan perhitungan <i>Profile Matching</i> dan AHP |
| 14 | Menejaman Verifikasi Peringkat Nilai Kinerja  | Fitur yang dapat diakses oleh kepala sekolah yang berfungsi untuk manajemen (edit, view) yaitu menggati status verifikasi dari hasil peringkat nilai kinerja guru.                                      |
| 15 | Melihat Peringkat Nilai Kinerja Terverifikasi | Fitur yang dapat diakses oleh admin dan guru untuk dapat melihat hasil dari peringkat nilai dari kinerja guru dan sebagai acuan calon kandidat kepala sekolah.  |
| 16 | Update Rekap Hasil Peringkat                  | Fitur yang dapat diakses oleh admin untuk melakukan update hasil peringkat sehingga tersimpan ke dalam menu database rekap penilaian tiap tahun ajaran  |
| 17 | Melihat Rekap Penilaian                       | Fitur yang dapat diakses oleh admin untuk melihat rekap penilaian seluruh guru tiap periode tahun ajaran  |
| 18 | Logout  | Fitur yang digunakan untuk keluar dari sistem   |

### 3.2.3 Usecase Skenario

*Usecase* skenario adalah dokumentasi terhadap kebutuhan fungsional sistem yang menjelaskan alur yang menggambarkan aksi aktor dan reaksi sistem. *Use case* skenario sistem informasi penilaian prestasi kinerja guru sebagai luaran pemilihan calon kepala sekolah dengan menggunakan metode *Profile Matching* dan AHP adalah sebagai berikut:

#### 1. *Usecase* Skenario Login

*Usecase* skenario ini menjelaskan alur untuk proses login untuk dapat mengakses fitur sistem. Aktor yang melakukan login adalah admin, kepala sekolah, dan guru. Setiap *user* harus memiliki username dan password yang terdaftar pada sistem. Hanya user berstatus aktif yang memiliki hak akses terhadap sistem. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif usecase skenario login terdapat pada tabel 4.3

Tabel 4. 3 Tabel Skenario Usecase Login

|                |   |
|----------------|---|
| Nomer Usecase  | USC-01  |
| Nama           | Login   |
| Aktor          | Admin, Kepala Sekolah, dan Guru                       |
| Pre Condition  | Admin, Kepala Sekolah, dan Guru akan melakukan login  |
| Post Condition | Admin, Kepala Sekolah, dan Guru telah melakukan login |

|                        |                              |
|------------------------|------------------------------|
| <b>SKENARIO NORMAL</b> |                              |
| <b>“Login”</b>         |                              |
| 1. Menjalankan sistem  |                              |
|                        | 2. Menampilkan halaman login |



|                                   |   |
|-----------------------------------|---|
| 3. Inputkan username dan password |   |
| 4. Menekan tombol Login           |   |
|                                   | 5. Melakukan pengecekan username, password, dan level di database tabel login dan nama sesuai id_user di tabel capres dengan function proses()  |
|                                   | 6. Untuk Login Level Admin :<br>Menampilkan halaman menu Home Sistem Penunjang Keputusan Pemilihan Calon Kepala Sekolah yang berisi gambar bertuliskan “Selamat Datang di sistem penilaian prestasi kinerja guru sebagai luaran pemilihan calon kepala sekolah menggunakan metode <i>Profile Matching – AHP</i> ” diikuti gambar icon input data guru, perhitungan <i>Profile Matching-AHP</i> , dan view report sesuai dengan nama user login masing-masing serta menampilkan menu Data Guru, Perhitungan Profile, |

|  |  |
|--|--|
|  | <p>Perhitungan AHP, Hasil Peringkat, Rekap, dan Logout.</p> <p>Untuk Level Kepala Sekolah :</p> <p>Menampilkan halaman menu Home dengan informasi login sesuai level serta berisi gambar bertuliskan Selamat Datang di sistem penilaian prestasi kinerja guru sebagai luaran pemilihan calon kepala sekolah menggunakan metode <i>Profile Matching – AHP</i>” diikuti gambar icon input data guru, perhitungan <i>Profile Matching-AHP</i>, dan view report dengan nama sesuai dengan user login masing-masing dan informasi login sesuai level, serta menampilkan menu Home, Nilai Kinerja, Biodata Pribadi, Kriteria Penilaian, Perhitungan Profile, Perhitungan AHP, dan Logout</p> <p>Untuk Level Guru :</p> <p>Menampilkan halaman Menu Home dengan informasi login</p> |
|--|--|

|   |  |
|---|--|
|   | <p>sesuai level serta berisi gambar bertuliskan Selamat Datang di sistem penilaian prestasi kinerja guru sebagai luaran pemilihan calon kepala sekolah menggunakan metode <i>Profile Matching – AHP</i>” diikuti gambar icon input data guru, perhitungan <i>Profile Matching-AHP</i>, dan view report dengan nama sesuai dengan user login di database tabel capres masing-masing dan informasi login sesuai level, serta menampilkan menu Home, Nilai Kinerja, Biodata Pribadi, dan Logout</p> |
| <p><b>SKENARIO ALTERNATIF</b><br/> <b>“Salah Username dan Password”</b></p> |  |
| 4a. Menekan button Login  |  |
|   | 5a. Menampilkan alert “Username dan Password Salah”  |
| <p><b>SKENARIO ALTERNATIF</b><br/> <b>“Belum Mengisi Field”</b></p>         |  |
| 4b. Menekan button Login  |  |
|   | 5b. Menampilkan pesan “Isi Isian Ini”  |

## 2. Usecase Manajemen Data Guru

Usecase skenario ini menjelaskan alur untuk proses melihat data guru, menambah data guru, dan mengedit jika terjadi kesalahan input. Aktor yang dapat melakukan ini adalah admin. Pada input data guru ini, data yang inputkan sekaligus berfungsi sebagai penilaian kinerja guru.

### a. Usecase Manajemen Data Guru (View)

Usecase skenario ini menjelaskan alur untuk proses melihat data guru. Aktor yang dapat melakukan ini adalah admin. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif usecase manajemen data guru (view) dapat dilihat pada tabel 4.4

Tabel 4. 4 Tabel Usecase Skenario Manajemen Data Guru (View)

|                |   |
|----------------|---|
| Nomer Usecase  | USC-02                                  |
| Nama           | Usecase Manajemen Data Guru (View)      |
| Aktor          | Admin                                   |
| Pre Condition  | Admin akan melakukan view ke data guru  |
| Post Condition | Admin telah melakukan view ke data guru |

| <b>SKENARIO DATA NORMAL</b>         |  |
|-------------------------------------|--|
| <b>“Manajemen Data Guru (View)”</b> |  |
| 1. Klik menu “Data Guru”            |  |
|                                     | 2. Mengambil data berupa nip, nama pegawai dan status dari tabel capres di database, dengan menggunakan function index() serta menampilkan no secara urut sesuai banyaknya data yang ditampilkan |

|   |  |
|---|--|
|   | <p>3. Menampilkan halaman data daftar guru dan tabel berisi no, nip, nama pegawai, status, detail yang didalamnya terdapat tombol icon mata (detail), edit, silang (non aktifkan user), centang (mengaktifkan user), refresh (menganti password menjadi default = 1234), serta tombol “nilai baru”</p>   |
| <p>4. Klik icon mata (Detail) pada salah satu baris dikolom</p> |  |
|   | <p>5. Mengambil data berupa level, NIP, nama pegawai, jenis kelamin, tanggal lahir, alamat kelahiran, alamat tinggal sekarang, nomer telepon rumah / nomer ponsel, dan tanggal mulai kerja, tahun ajaran, nilai kualifikasi, nilai kesehatan, nilai sertifikat, nilai golongan, nilai usia, nilai catatan hukum, nilai pengalaman, nilai DP3, dan nilai tambahan dari function detail_pegawai dengan menggunakan atribut id_user yang berasal dari tabel capres,</p> |

|  |   |
|--|---|
|  | tb_nilai_pm, login, dan<br>tb_subkriteria   |
|  | 6. Menampilkan halaman detail data guru berupa level, NIP, nama pegawai, jenis kelamin, tanggal lahir, alamat kelahiran, alamat tinggal sekarang, nomer telepon rumah / nomer ponsel, dan tanggal mulai kerja, tahun ajaran, nilai kualifikasi, nilai kesehatan, nilai sertifikat, nilai golongan, nilai usia, nilai catatan hukum, nilai pengalaman, nilai DP3, dan nilai tambahan |

b. Usecase Manajemen Data Guru (Input)

Usecase skenario ini menjelaskan alur untuk proses menambah data guru. Aktor yang dapat melakukan ini adalah admin. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif usecase manajemen data guru (input) dapat dilihat pada tabel 4.5

Tabel 4. 5 Tabel Usecase Skenario Manajemen Data Guru (Input)

|                |  |
|----------------|--|
| Nomer Usecase  | USC-02                                   |
| Nama           | Usecase Manajemen Data Guru (Input)      |
| Aktor          | Admin                                    |
| Pre Condition  | Admin akan melakukan input ke data guru  |
| Post Condition | Admin telah melakukan input ke data guru |

| <b>SKENARIO DATA NORMAL</b><br><b>“Manajemen Data Guru (Input)”</b> |   |
|---|---|
| 1. Klik menu “Data Guru”  |   |
|   | 2. Mengambil data berupa nip, nama pegawai dan status dari tabel capres di database, dengan menggunakan function index() serta menampilkan no secara urut sesuai banyaknya data yang ditampilkan  |
|   | 3. Menampilkan halaman data daftar guru dan tabel berisi no, nip, nama pegawai, status, detail yang didalamnya terdapat tombol icon mata (detail), edit, silang (non aktifkan user), centang (mengaktifkan user), refresh (menganti password menjadi default = 1234), serta tombol “nilai baru” |
| 4. Klik tombol “Tambah Baru”  |   |
|   | 5. Menampilkan halaman form tambah data guru yang terdiri dari level (berupa combo box yang berisi pilihan Kepala Sekolah dan Guru), tahun ajaran (berupa combo box yang berisi   |

|  |   |
|--|---|
|  | <p>pilihan tahun ajaran), nip, nama pegawai, jenis kelamin (berupa combo box yang berisi pilihan pria dan wanita), tanggal lahir, alamat kelahiran, alamat tinggal sekarang, nomer telepon rumah / nomer ponsel, tanggal mulai kerja, kualifikasi akademik (combo box berisi pilihan D4, S1, S2), kesehatan (combo box berisi pilihan sangat sehat, sehat, kurang sehat, tidak sehat), catatan hukum (combo box berisi pilihan tidak pernah, hukuman ringan 1-3 kali, dan hukuman ringan &gt; 3 kali), sertifikat (combo box berisi pilihan memiliki sertifikat pendidik dan tidak memiliki sertifikat pendidik), golongan (combo box berisi pilihan IIIC, IIID, IVA, IVB, IVC, IVD, IVE) DP3 (combo box berisi pilihan (nilai amat baik untuk unsur kesetiaan dan nilai amat baik untuk unsur lainnya dan nilai amat baik untuk unsur kesetiaan dan nilai baik untuk</p> |
|--|---|



|  |   |
|--|---|
|  | <p>usur lainnya), tugas tambahan (combo box yang berisi pilihan memiliki tugas tambahan dan tidak memiliki tugas tambahan) yang berasal dari function <code>gettambahpegawai()</code> dengan menggunakan tabel <code>tb_subkriteria</code> dari database. Serta menampilkan tombol <i>save</i> dan <i>cancel</i> di bawah form</p>  |
| 6. Mengisi seluruh isian form tambah data guru |   |
| 7. Klik tombol “ <i>save</i> ”                 |   |
|  | <p>8. Menyimpan data ke dalam database menggunakan function <code>submit()</code> dengan menggunakan attribute <code>id_user</code>, <code>nip</code>, <code>namaPeg</code>, <code>jenisKel</code>, <code>tanggalL</code>, <code>tempatTL</code>, <code>alamatS</code>, <code>nope</code>, <code>tanggalMKerja</code>, <code>status_user</code> ke tabel <code>capres</code>, attribute <code>id_user</code>, <code>Username</code>, <code>id_level</code>, <code>Password</code>, dan <code>status</code> ke tabel <code>login</code> dan tabel <code>tb_nilai_pm</code> dengan attribute <code>id_kriteria_pm</code>, <code>id_user</code>, <code>nilai</code>, <code>hasil</code>, dan <code>tahun_ajaran</code></p> |
|  | <p>9. Menampilkan pesan “SUKSES ENTRY DATA”</p>   |

|  |   |
|--|---|
|  | <p>10. Menampilkan halaman data daftar guru dan tabel berisi no, nip, nama pegawai, status, detail yang didalamnya terdapat tombol icon mata (detail), edit, silang (non aktifkan user), centang (mengaktifkan user), refresh (menganti password menjadi default = 1234), serta tombol “nilai baru”</p> |
|  | <p>11. Mengambil data berupa nip, nama pegawai dan status dari tabel capres di database, dengan menggunakan function index() serta menampilkan no secara urut sesuai banyaknya data yang ditampilkan</p>  |
| <p><b>SKENARIO ALTERNATIF</b><br/><b>“Cancel”</b></p>  |   |
| <p>5a. Mengisi seluruh isian form tambah data guru</p> |   |
| <p>6a. Klik tombol “cancel”</p>                        |   |
|  | <p>7a. Mengambil data berupa nip, nama pegawai dan status dari tabel capres di database, dengan menggunakan function index() serta menampilkan no secara urut sesuai banyaknya data yang ditampilkan</p>  |

|  |  |
|--|--|
|  | 8a. Menampilkan halaman data daftar guru dan tabel berisi no, nip, nama pegawai, status, detail yang didalamnya terdapat tombol icon mata (detail), edit, silang (non aktifkan user), centang (mengaktifkan user), refresh (menganti password menjadi default = 1234), serta tombol “nilai baru” |
| <b>SKENARIO ALTERNATIF</b><br><b>“Belum Mengisi Field”</b> |  |
| 5b. Kik tombol “save”                                      |  |
|  | 6b. Menampilkan pesan “Isi Isian Ini” (untuk textfield) atau “pilih salah satu dari item terdaftar” (untuk combo box)  |
| <b>SKENARIO ALTERNATIF</b><br><b>“Inputan Tidak Valid”</b> |  |
| 5c. Klik tombol “save”                                     |  |
|  | 6c. Menampilkan pesan “Masukkan angka!” (Untuk inputan NIP dan Nomer Telepon jika bukan angka)   |
| <b>SKENARIO ALTERNATIF</b><br><b>“Inputan NIP SAMA”</b>    |  |
| 5d. Klik tombol “save”                                     |  |

|  |  |
|--|--|
|  | 6d. Menampilkan pesan “GAGAL!<br>Data Sudah Ada” |
|--|--|

c. *Usecase* Skenario Manajemen Data Guru (*edit*)

*Usecase* skenario ini menjelaskan alur untuk proses mengubah data profil masing-masing guru. Aktor yang dapat mengubah data pengguna adalah admin. Apabila terjadi kesalahan dalam pengisian data, maka Admin dapat mengedit data guru tersebut. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif usecase skenario manajemen data guru (*edit*) dapat dilihat pada Tabel 4.6

Tabel 4. 6 Tabel Usecase Skenario Manajemen Data Guru (Edit)

|                |   |
|----------------|---|
| Nomer Usecase  | USC-02                                  |
| Nama           | Usecase Manajemen Data Guru (Edit)      |
| Aktor          | Admin                                   |
| Pre Condition  | Admin akan melakukan edit ke data guru  |
| Post Condition | Admin telah melakukan edit ke data guru |

| <b>SKENARIO DATA NORMAL</b><br><b>“Manajemen Data Guru (Edit)”</b> |  |
|--|--|
| 1. Klik menu “Data Guru”   |  |
|  | 2. Mengambil data berupa nip, nama pegawai dan status dari tabel capres di database, dengan menggunakan function index() serta menampilkan no secara urut sesuai banyaknya data yang ditampilkan |

|  |  |
|--|--|
|  | <p>3. Menampilkan halaman data daftar guru dan tabel berisi no, nip, nama pegawai, status, detail yang didalamnya terdapat tombol icon mata (detail), edit, silang (non aktifkan user), centang (mengaktifkan user), refresh (menganti password menjadi default = 1234), serta tombol “nilai baru”</p>   |
| <p>4. Klik icon “Edit” pada salah satu baris yang ingin diedit</p> |  |
|  | <p>5. Mengambil data berupa level, NIP, nama pegawai, jenis kelamin, tanggal lahir, alamat kelahiran, alamat tinggal sekarang, nomer telepon rumah / nomer ponsel, dan tanggal masuk kerja, tahun ajaran, nilai kualifikasi, nilai kesehatan, nilai sertifikat, nilai golongan, nilai usia, nilai catatan hukum, nilai pengalaman, nilai DP3, dan nilai tambahan dari function detail_pegawai dengan menggunakan atribut id_user yang berasal dari tabel</p> |

|  |  |
|--|--|
|  | capres, tb_nilai_pm, login, dan tb_subkriteria   |
|  | 6. Menampilkan halaman edit data guru yang berisi form biodata profil guru yang telah terisi level, NIP, Nama Pegawai, Jenis Kelamin, Tanggal Lahir, Alamat Kelahiran, Alamat Tinggal Sekarang, Nomer Telepon Rumah / Nomer Ponsel, Tanggal Mulai Kerja, Tahun Ajaran, Nilai Kualifikasi, Nilai Kesehatan, Nilai Sertifikat, Nilai Golongan, Nilai Catatan Hukum, Nilai DP3, dan Nilai Tambahan serta tombol <i>save</i> dan <i>Cancel</i> pada bawah form |
| 7. Menggubah isian form edit data guru |  |
| 8. Klik tombol “ <i>Save</i> ”         |  |
|  | 9. Menyimpan data ke dalam database menggunakan function edit() dengan menggunakan attribute id_user, nip, namaPeg, jenisKel, tanggalL, tempatTL, alamatS, nope, tanggalMKerja,  |

|                                   |   |
|-----------------------------------|---|
|                                   | <p>status_user ke tabel capres, kemudian attribute id_user, Username, id_level, Password, status ke tabel login dan tabel tb_nilai_pm dengan attribute nilai, tahun_ajaran, dan id_user</p>   |
|                                   | <p>10. Menampilkan pesan “Sukses Edit Data Guru”</p>  |
|                                   | <p>11. Menampilkan halaman data daftar guru dan tabel berisi no, nip, nama pegawai, status, detail yang didalamnya terdapat tombol icon mata (detail), edit, silang (non aktifkan user), centang (mengaktifkan user), refresh (menganti password menjadi default = 1234), serta tombol “nilai baru”</p> |
|                                   | <p>12. Mengambil data berupa nip, nama pegawai dan status dari tabel capres di database, dengan menggunakan function index() serta menampilkan no secara urut sesuai banyaknya data yang ditampilkan</p>  |
| <p><b>SKENARIO ALTERNATIF</b></p> |   |

| <b>“Cancel”</b>                                |  |
|--|--|
| 5a. Menggubah isian form edit data daftar guru |  |
| 6a. Klik tombol “Cancel”                       |  |
|  | 7a. Menampilkan halaman data daftar guru dan tabel berisi no, nip, nama pegawai, status, detail yang didalamnya terdapat tombol icon mata (detail), edit, silang (non aktifkan user), centang (mengaktifkan user), refresh (menganti password menjadi default = 1234), serta tombol “nilai baru” |
|  | 8a. Mengambil data berupa nip, nama pegawai dan status dari tabel capres di database, dengan menggunakan function index() serta menampilkan no secara urut sesuai banyaknya data yang ditampilkan  |
| <b>SKENARIO ALTERNATIF</b>                     |  |
| <b>“Belum Mengisi Field”</b>                   |  |
| 6b. Klik tombol “save”                         |  |
|  | 7b. Menampilkan pesan “Isi Isian Ini”  |
| <b>SKENARIO ALTERNATIF</b>                     |  |
| <b>“Inputan Tidak Valid”</b>                   |  |
| 6c. Klik tombol “save”                         |  |



|   |   |
|---|---|
|   | 7c. Menampilkan pesan<br>“Masukkan angka!” (Untuk<br>inputan NIP dan Nomer<br>Telepon jika bukan angka) |
| <b>SKENARIO ALTERNATIF</b><br><b>“Inputan NIP SAMA”</b> |   |
| 6d. Klik tombol “save”                                  |   |
|   | 7d. Menampilkan pesan “Gagal,<br>NIP sudah digunakan”   |

### 3. Usecase Manajemen Tambah Nilai Guru

Usecase skenario ini menjelaskan alur untuk proses melihat tambah nilai guru, dan menambah data nilai guru pada tahun ajaran baru. Aktor yang dapat melakukan ini adalah admin. Pada input tambah nilai guru ini, data yang di inputkan berfungsi sebagai penilaian kinerja guru pada tahun ajaran terbaru.

#### a. Usecase Manajemen Tambah Nilai Guru (View)

Usecase skenario ini menjelaskan alur untuk proses melihat tambah nilai guru. Aktor yang dapat melakukan ini adalah admin. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif usecase manajemen tambah nilai guru (view) dapat dilihat pada tabel 4.7

Tabel 4. 7 Tabel Usecase Skenario Manajemen Tambah Nilai Guru (View)

|                |   |
|----------------|---|
| Nomer Usecase  | USC-03  |
| Nama           | Usecase Manajemen Tambah Nilai Guru (View)      |
| Aktor          | Admin   |
| Pre Condition  | Admin akan melakukan view ke tambah nilai guru  |
| Post Condition | Admin telah melakukan view ke tambah nilai guru |

| <b>SKENARIO DATA NORMAL</b>                            |   |
|--|---|
| <b>“Manajemen Tambah Nilai Guru (View)”</b>            |   |
| 1. Klik menu “Data Guru”                               |   |
|  | 2. Mengambil data berupa nip, nama pegawai dan status dari tabel capres di database, dengan menggunakan function index() serta menampilkan no secara urut sesuai banyaknya data yang ditampilkan  |
|  | 3. Menampilkan halaman data daftar guru dan tabel berisi no, nip, nama pegawai, status, detail yang didalamnya terdapat tombol icon mata (detail), edit, silang (non aktifkan user), centang (mengaktifkan user), refresh (menganti password menjadi default = 1234), serta tombol “nilai baru” |
| 4. Klik tombol “Tambah Nilai” pada salah baris ditabel |   |
|  | 5. Menampilkan halaman Tambah Nilai Guru berupa form berisi level, NIP, nama pegawai, jenis kelamin, tanggal lahir, alamat lahir, nomer telepon rumah/nomer ponsel, tanggal   |

|  |  |
|--|--|
|  | <p>mulai kerja dari function <code>gettambahnilai()</code> yang berasal dari tabel <code>capres</code>, <code>tb_nilai_pm</code>, dan login beserta form berupa tahun ajaran, kualifikasi akademik, kesehatan, catatan hukum, sertifikat, golongan, DP3, dan tugas tambahan yang berasal dari tabel <code>tb_subkriteria</code> didatabase. Serta tombol <i>save</i> dan <i>Cancel</i></p> |
|--|--|

b. Usecase Manajemen Tambah Nilai Guru (Input)

Usecase skenario ini menjelaskan alur untuk proses menambah nilai guru. Aktor yang dapat melakukan ini adalah admin. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif usecase manajemen tambah nilai guru (input) dapat dilihat pada tabel 4.8

Tabel 4. 8 Tabel Usecase Skenario Manajemen Tambah Nilai Guru (Input)

|                |  |
|----------------|--|
| Nomer Usecase  | USC-03   |
| Nama           | Usecase Manajemen Tambah Nilai Guru (Input)      |
| Aktor          | Admin  |
| Pre Condition  | Admin akan melakukan input ke tambah nilai guru  |
| Post Condition | Admin telah melakukan input ke tambah nilai guru |

|  |  |
|--|--|
| <b>SKENARIO DATA NORMAL</b>                  |  |
| <b>“Manajemen Tambah Nilai Guru (Input)”</b> |  |
| 1. Klik menu “Data Guru”                     |  |

|   |  |
|---|--|
|   | <p>2. Mengambil data berupa nip, nama pegawai dan status dari tabel capres di database, dengan menggunakan function index() serta menampilkan no secara urut sesuai banyaknya data yang ditampilkan</p>  |
|   | <p>3. Menampilkan halaman data daftar guru dan tabel berisi no, nip, nama pegawai, status, detail yang didalamnya terdapat tombol icon mata (detail), edit, silang (non aktifkan user), centang (mengaktifkan user), refresh (menganti password menjadi default = 1234), serta tombol “nilai baru”</p> |
| <p>4. Klik tombol “Tambah Nilai” pada salah baris ditabel</p> |  |
|   | <p>5. Menampilkan halaman Tambah Nilai Guru berupa form berisi level, NIP, nama pegawai, jenis kelamin, tanggal lahir, alamat lahir, nomer telepon rumah/nomer</p>   |

|                                   |  |
|-----------------------------------|--|
|                                   | ponsel, tanggal mulai kerja dari function <code>gettambahnilai()</code> yang berasal dari tabel <code>capres</code> , <code>tb_nilai_pm</code> , dan login beserta form berupa tahun ajaran, kualifikasi akademik, kesehatan, catatan hukum, sertifikat, golongan, DP3, dan tugas tambahan yang berasal dari tabel <code>tb_subkriteria</code> didatabase dan tombol <i>save</i> serta <i>Cancel</i> |
| 6. Mengisi form tambah nilai guru |  |
| 7. Klik tombol “ <i>Save</i> ”    |  |
|                                   | 8. Mengubah data dalam database berdasarkan <code>id_user</code> berupa attribute <code>tahunajaran</code> dan nilai pada tabel <code>tb_nilai_pm</code>   |
|                                   | 9. Menampilkan pesan “Berhasil Entry Data”   |
|                                   | 10. Mengambil data berupa <code>nip</code> , nama pegawai dan status dari tabel <code>capres</code> di database, dengan menggunakan function <code>index()</code> serta menampilkan no secara urut   |

|                                    |  |
|------------------------------------|--|
|                                    | sesuai banyaknya data yang ditampilkan   |
|                                    | 11. Menampilkan halaman data daftar guru dan tabel berisi no, nip, nama pegawai, status, detail yang didalamnya terdapat tombol icon mata (detail), edit, silang (non aktifkan user), centang (mengaktifkan user), refresh (menganti password menjadi default = 1234), serta tombol “nilai baru” |
| <b>SKENARIO ALTERNATIF</b>         |  |
| <b>“Cancel”</b>                    |  |
| 5a. Mengisi form tambah nilai guru |  |
| 6a. Klik tombol “cancel”           |  |
|                                    | 7a. Menampilkan halaman data daftar guru dan tabel berisi no, nip, nama pegawai, status, detail yang didalamnya terdapat tombol icon mata (detail), edit, silang (non aktifkan user), centang (mengaktifkan user), refresh (menganti password menjadi default = 1234), serta tombol “nilai baru” |

|  |  |
|--|--|
|  | 8a. Mengambil data berupa nip, nama pegawai dan status dari tabel capres di database, dengan menggunakan function index() serta menampilkan no secaraurut sesuai banyaknya data yang ditampilkan |
| <b>SKENARIO ALTERNATIF</b><br><b>“Tahun Ajaran Sama”</b> |  |
| 6b. Klik tombol “save”                                   |  |
|  | 7b. Menampilkan pesan<br>“GAGAL! Data Sudah Ada”   |

4. Tabel Usecase Skenario Manajemen Data User (Update)

Penjelasan urutan aksi aktor untuk scanerio Usecase Manajemen Data User dan reaksi sistem pada scenario normal dan scenario alternatif terdapat pada lampiran A (Scenario)

5. Tabel Usecase Skenario Manajemen Data Pribadi Guru

Penjelasan urutan aksi aktor untuk scanerio Usecase Manajemen Data Pribadi Guru dan reaksi sistem pada scenario normal dan scenario alternatif terdapat pada lampiran A (Scenario)

6. Tabel Usecase Skenario Update Password Pribadi

Penjelasan urutan aksi aktor untuk scanerio Usecase Update Password Pribadi dan reaksi sistem pada scenario normal dan scenario alternatif terdapat pada lampiran A (Scenario)

7. Tabel Usecase Skenario Melihat Kriteria Penilaian Kinerja Guru

Penjelasan urutan aksi aktor untuk skenario Usecase Melihat Kriteria Penilaian Kinerja Guru dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada lampiran A (Scenario)

8. Tabel Usecase Skenario Melihat Nilai Kinerja Guru

Penjelasan urutan aksi aktor untuk skenario Usecase Melihat Nilai Kinerja Guru dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada lampiran A (Scenario)

9. Usecase Manajemen Nilai Ideal *Profile Matching*

Penjelasan urutan aksi aktor untuk skenario Usecase Manajemen Nilai Ideal *Profile Matching* dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada lampiran A (Scenario)

10. Tabel Usecase Skenario Manajemen Bobot Kriteria AHP

Penjelasan urutan aksi aktor untuk skenario Usecase Manajemen Bobot Kriteria AHP dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada lampiran A (Scenario)

11. Tabel Usecase Skenario Melihat Nilai Kinerja *Profile Matching* (View)

Penjelasan urutan aksi aktor untuk skenario Usecase Melihat Nilai Kinerja *Profile Matching* (View) dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada lampiran A (Scenario)

12. Tabel Usecase Skenario Melihat Bobot Kriteria AHP

Penjelasan urutan aksi aktor untuk skenario Usecase Melihat Bobot Kriteria AHP dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada lampiran A (Scenario)



13. Tabel Usecase Skenario Melihat Peringkat Nilai Kinerja

Penjelasan urutan aksi aktor untuk scanerio usecase Melihat Peringkat Nilai Kinerja dan reaksi sistem pada scenario normal dan scenario alternatif terdapat pada lampiran A (*Scenario*)

14. Tabel Usecase Skenario Menejaman Verifikasi Peringkat Nilai Kinerja

Penjelasan urutan aksi aktor untuk scanerio Usecase Menejaman Verifikasi Peringkat Nilai Kinerja dan reaksi sistem pada scenario normal dan scenario alternatif terdapat pada lampiran A (*Scenario*)

15. Tabel Usecase Skenario Melihat Peringkat Nilai Kinerja Terverifikasi

Penjelasan urutan aksi aktor untuk scanerio Usecase Melihat Peringkat Nilai Kinerja Terverifikasi dan reaksi sistem pada scenario normal dan scenario alternatif terdapat pada lampiran A (*Scenario*)

16. Tabel Usecase Skenario Manajeman Update Rekap Hasil Peringkat

Penjelasan urutan aksi aktor untuk scanerio Usecase Manajeman Update Rekap Hasil Peringkat dan reaksi sistem pada scenario normal dan scenario alternatif terdapat pada lampiran A (*Scenario*)

17. Tabel Usecase Skenario Melihat Rekap Penilaian

Penjelasan urutan aksi aktor untuk scanerio Usecase Melihat Rekap Penilaian dan reaksi sistem pada scenario normal dan scenario alternatif terdapat pada lampiran A (*Scenario*)

18. Tabel Usecase Skenario Logout

Penjelasan urutan aksi aktor untuk scanerio Usecase logout dan reaksi sistem pada scenario normal dan scenario alternatif terdapat pada lampiran A (*Scenario*)

#### 4.2.4 Activity Diagram

*Activity diagram* adalah dokumentasi desain yang menggambarkan aliran aktivitas dalam sistem penilaian prestasi kerja guru sebagai luaran pemilihan calon kepala sekolah menggunakan metode *Profile Matching* – AHP yang akan dibangun. Sistem penilaian prestasi kerja guru sebagai luaran pemilihan calon kepala sekolah menggunakan metode *Profile Matching* – AHP memiliki 26 *activity diagram* yaitu sebagai berikut:

1. *Activity Diagram* login

*Activity Diagram* login ini menjelaskan alur untuk proses login untuk dapat mengakses fitur sistem. Aktor yang melakukan login adalah admin, kepala sekolah, dan guru. Setiap *user* harus memiliki username dan password yang terdaftar pada sistem. Hanya user berstatus aktif yang memiliki hak akses terhadap sistem. *Activity diagram* login terdapat pada Gambar 4.3

2. *Activity Diagram* manajemen data guru (view)

*Activity Diagram* ini menjelaskan alur untuk proses melihat data guru. Aktor yang dapat melakukan ini adalah admin. *Activity diagram* manajemen data guru (view) terdapat pada Gambar 4.4

3. *Activity Diagram* manajemen data guru (input)

*Activity Diagram* ini menjelaskan alur untuk proses menambah data guru. Aktor yang dapat melakukan ini adalah admin. *Activity diagram* manajemen data guru (input) terdapat pada Gambar 4.5

4. *Activity Diagram* Manajemen Data Guru (edit)

*Activity Diagram* ini menjelaskan alur untuk proses mengubah data profil masing-masing guru. Aktor yang dapat mengubah data pengguna adalah admin.

Apabila terjadi kesalahan dalam pengisian data, maka Admin dapat mengedit data guru tersebut. *Activity* diagram Manajemen Data Guru (edit) terdapat pada Gambar 4.6

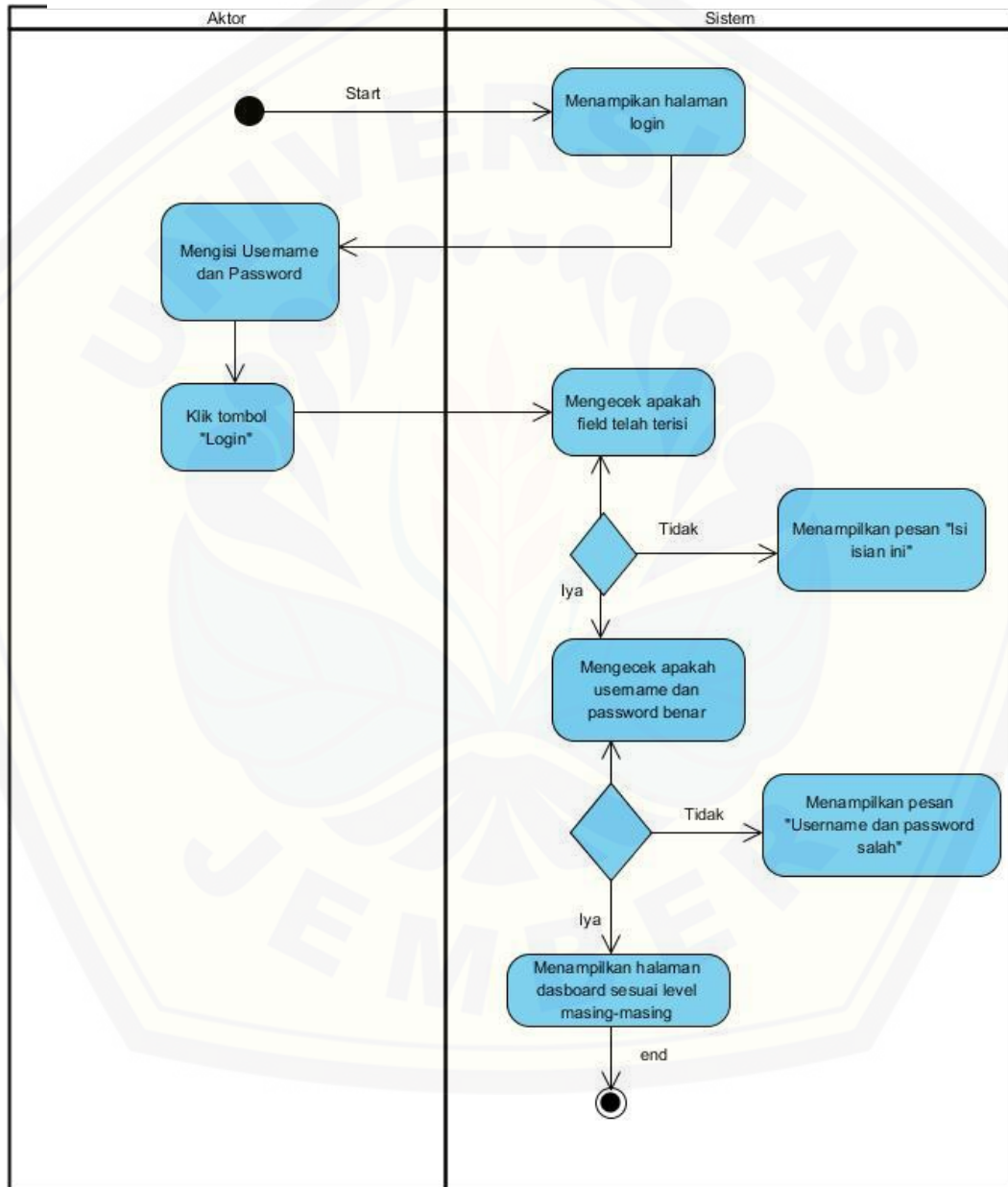
5. *Activity* Diagram Manajemen Tambah Nilai Guru (View)

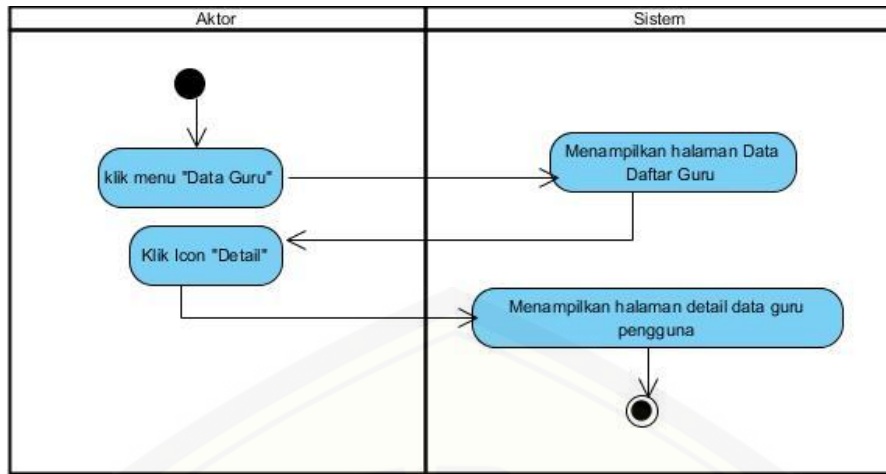
*Activity* Diagram ini menjelaskan alur untuk proses melihat tambah nilai guru, dan menambah data nilai guru pada tahun ajaran baru. Aktor yang dapat melakukan ini adalah admin. Pada input tambah nilai guru ini, data yang di inputkan berfungsi sebagai penilaian kinerja guru pada tahun ajaran terbaru. *Activity* diagram Diagram Manajemen Tambah Nilai Guru (View) terdapat pada Gambar 4.7

6. *Activity* Diagram Manajemen Tambah Nilai Guru (Input)

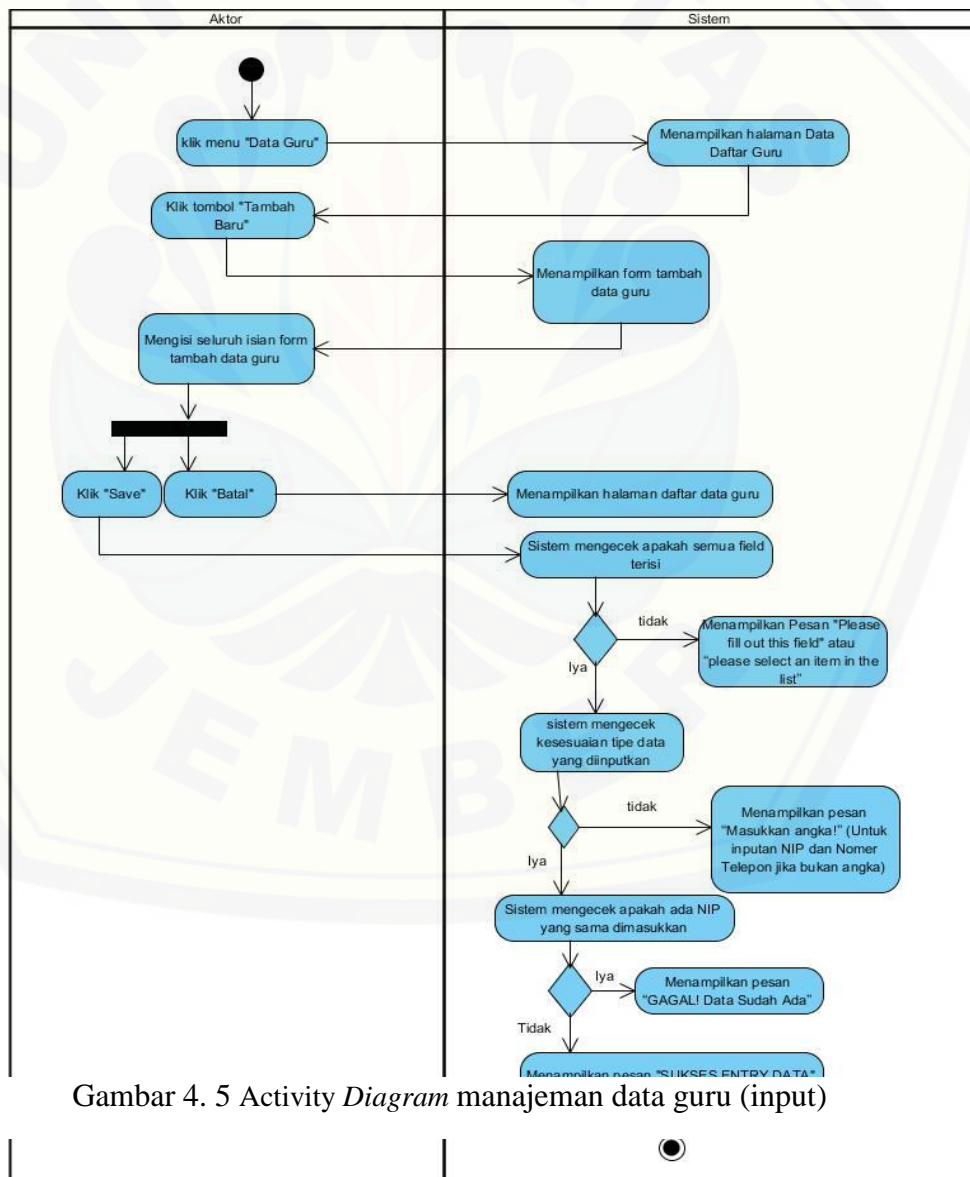
*Activity Diagram* ini menjelaskan alur untuk proses menambah nilai guru. Aktor yang dapat melakukan ini adalah admin. *Activity diagram* Manajemen Tambah Nilai Guru (Input) terdapat pada Gambar 4.8

Gambar 4. 3 *Activity Diagram* Login

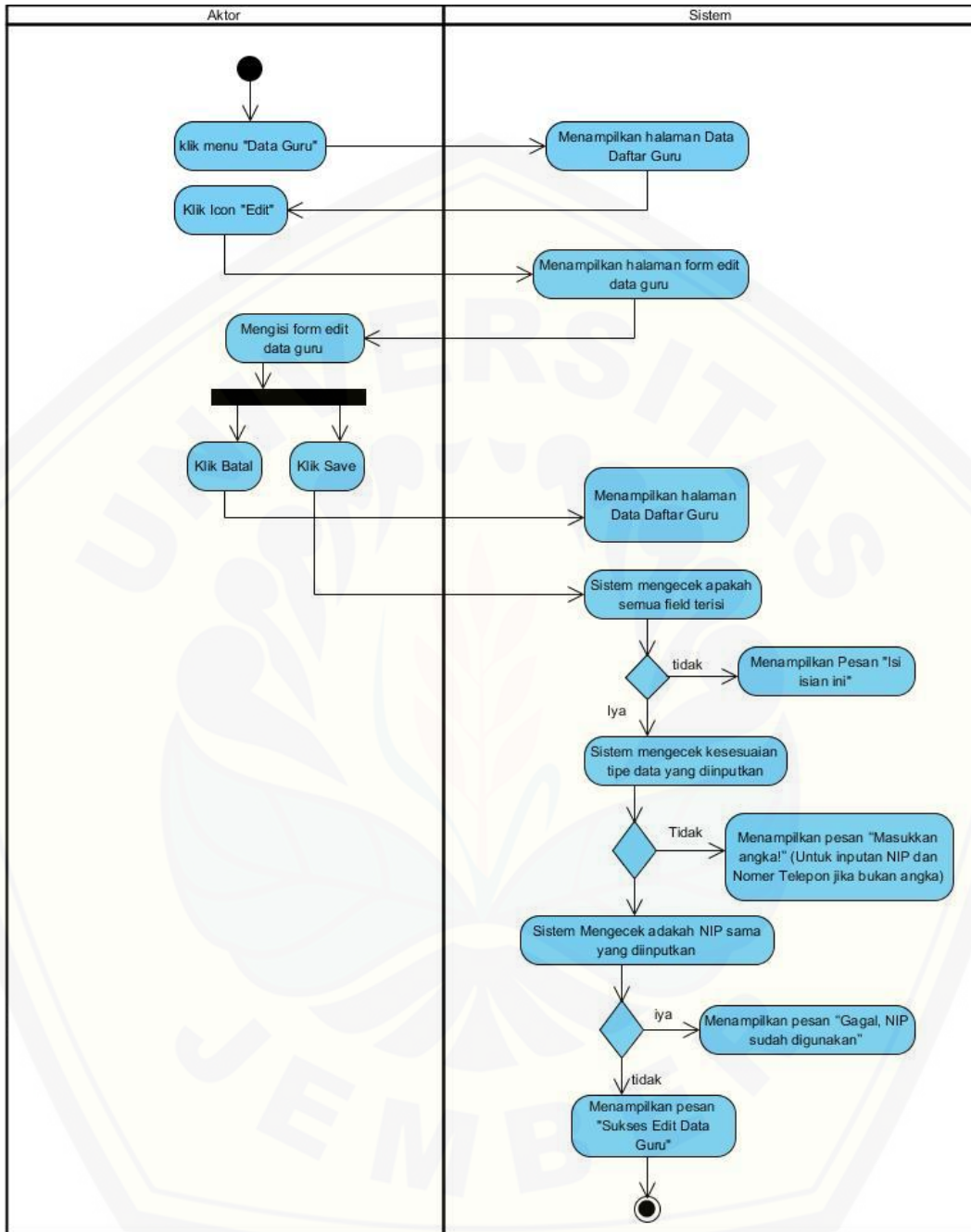




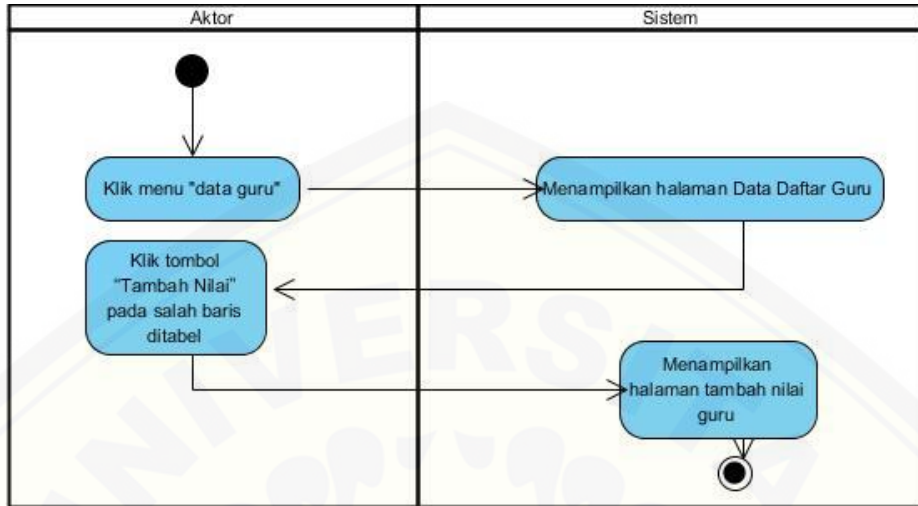
Gambar 4. 4 Activity Diagram manajemen data guru (view)



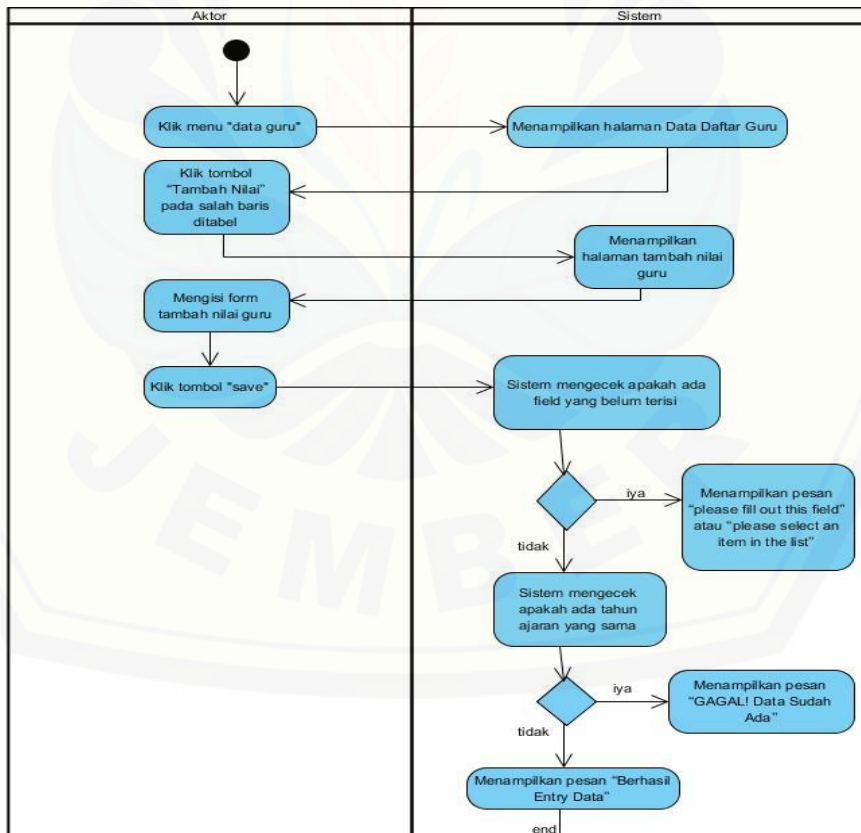
Gambar 4. 5 Activity Diagram manajemen data guru (input)



Gambar 4. 6 Activity Diagram manajemen data guru (edit)



Gambar 4. 7 Activity Diagram manajemen tambah nilai guru



Gambar 4. 8 Activity Diagram manajemen tambah nilai guru (input)

#### 4.2.5 Sequence Diagram

*Sequence diagram* merupakan dokumentasi desain berbentuk diagram terurut yang menampilkan interaksi-interaksi antar objek dalam sistem. *Sequence diagram* digunakan untuk menggambarkan Scenario dan memodelkan aliran logika dalam sistem secara *visual*. *Sequence diagram* dari sistem Penilaian Prestasi Kerja Guru Sebagai Luaran Pemilihan Calon Kepala Sekolah Menggunakan Metode *Profile Matching* – AHP adalah sebagai berikut :

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

*Sequence diagram* ini menggambarkan Skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan login ke dalam sistem sesuai dengan user login masing-masing. Aktor yang dapat melakukan login adalah admin, Kepala Sekolah, dan guru. Pada *sequence* ini terdapat class view vlogin.php dan vhome, class controller login dan dashboard, dan class model Mlogin, serta dalam class tersebut terdapat berbagai method yang dibutuhkan dalam proses ini. *Sequence diagram login* (Admin, Kepala Sekolah, Guru) dapat dilihat pada Gambar 4.9. Diagram *sequence* lainnya terdapat pada Lampiran C (*Sequence Diagram*).

##### 2. *Sequence Diagram Manajemen Data Guru (View)*

*Sequence diagram* ini menggambarkan Skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan view ke data guru. Aktor yang dapat melakukan view ke data guru adalah Admin. Pada *sequence* ini terdapat class view header.php, V\_datapegawai.php, dan V\_detailpegawai.php, class controller C\_datapegawai, dan class model M\_datapegawai.php serta dalam class tersebut terdapat berbagai method yang dibutuhkan dalam proses ini. *Sequence diagram manajemen data guru (view)* dapat dilihat pada Gambar 4.10. Diagram *sequence* lainnya terdapat pada Lampiran C (*Sequence Diagram*).



### 3. *Sequence* Diagram Manajemen Data Guru (Input)

*Sequence* diagram ini menggambarkan Skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan input ke data guru. Aktor yang dapat melakukan input ke data guru adalah Admin. Pada *sequence* ini terdapat class view Header.php, V\_datapegawai.php, dan V\_inputpegawai.php, class controller C\_datapegawai.php dan class model M\_datapegawai.php serta dalam class tersebut terdapat berbagai method yang dibutuhkan dalam proses ini. *Sequence* diagram manajemen data guru (input) dapat dilihat pada Gambar 4.11. Diagram *sequence* lainnya terdapat pada Lampiran C (Sequence Diagram).

### 4. *Sequence* Diagram Manajemen Data Guru (Edit)

*Sequence* diagram ini menggambarkan Skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan edit ke data guru. Aktor yang dapat melakukan edit ke data guru adalah Admin. Pada *sequence* ini terdapat class view Header.php, V\_datapegawai.php, V\_inputPegawai.php, dan V\_editPegawai.php, class controller C\_datapegawai.php, dan class model M\_datapegawai.php serta dalam class tersebut terdapat berbagai method yang dibutuhkan dalam proses ini. *Sequence* diagram manajemen data guru (edit) dapat dilihat pada Gambar 4.12. Diagram *sequence* lainnya terdapat pada Lampiran C (Sequence Diagram)

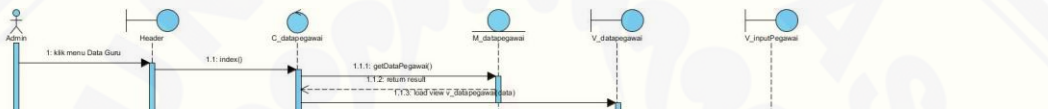
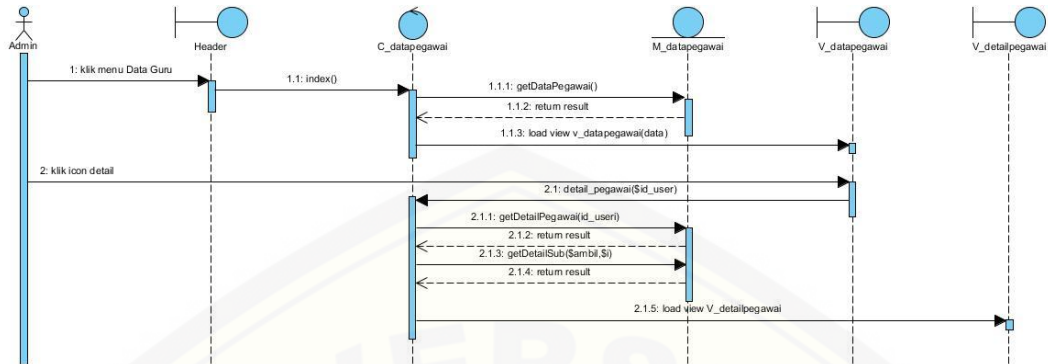
### 5. *Sequence* Diagram Manajemen Tambah Nilai Guru (View)

*Sequence* diagram ini menggambarkan Skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan view tambah nilai guru. Aktor yang dapat melakukan view tambah nilai

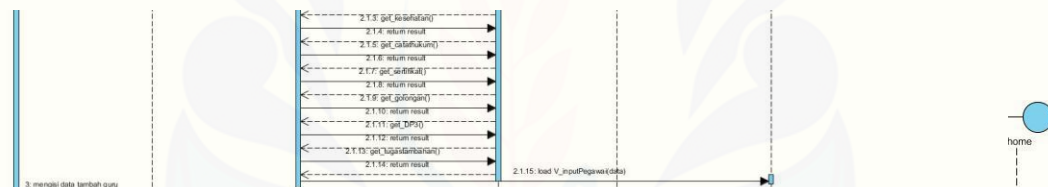
guru adalah Admin. Pada *sequence* ini terdapat class view V\_datapegawai.php dan V\_inputpegawai.php, class controller C\_datapegawai.php, dan class model M\_datapegawai.php serta dalam class tersebut terdapat berbagai method yang dibutuhkan dalam proses ini. *Sequence* diagram manajemen tambah nilai guru (view) dapat dilihat pada Gambar 4.13. Diagram *sequence* lainnya terdapat pada Lampiran C (Sequence Diagram)

#### 6. *Sequence* Diagram Manajemen Tambah Nilai Guru (Input)

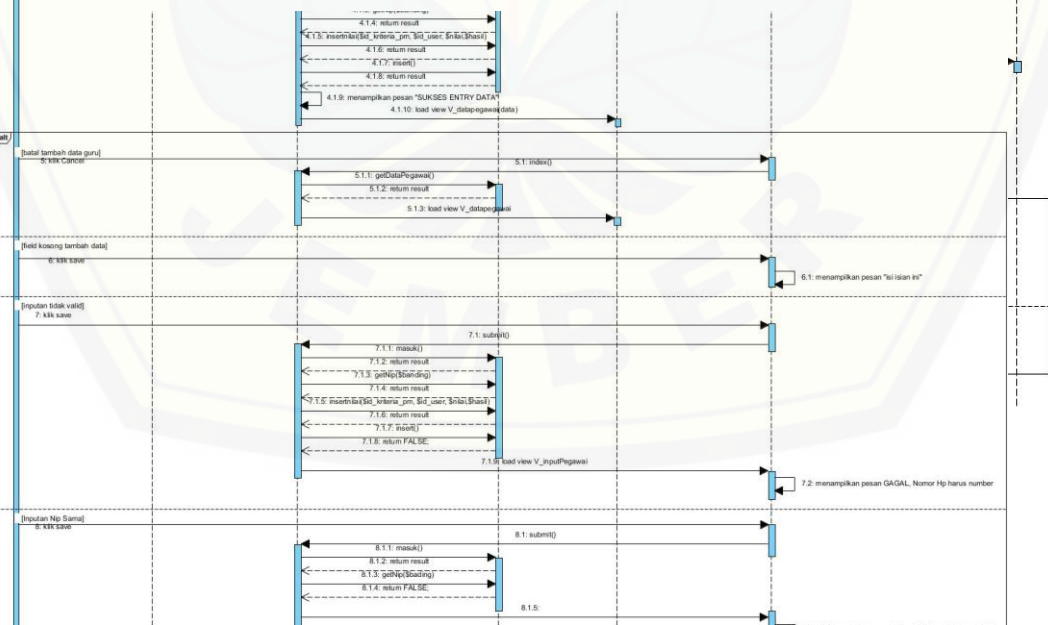
*Sequence* diagram ini menggambarkan Skenario atau rangkaian langkah langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan input tambah nilai guru. Aktor yang dapat melakukan input tambah nilai guru adalah Admin. Pada *sequence* ini terdapat class view view V\_datapegawai.php dan V\_inputpegawai.php, class controller C\_datapegawai.php, dan class model M\_datapegawai.php serta dalam class tersebut terdapat berbagai method yang dibutuhkan dalam proses ini. *Sequence* diagram manajemen tambah nilai guru (input) dapat dilihat pada Gambar 4.14. Diagram *sequence* lainnya terdapat pada Lampiran C (Sequence Diagram)



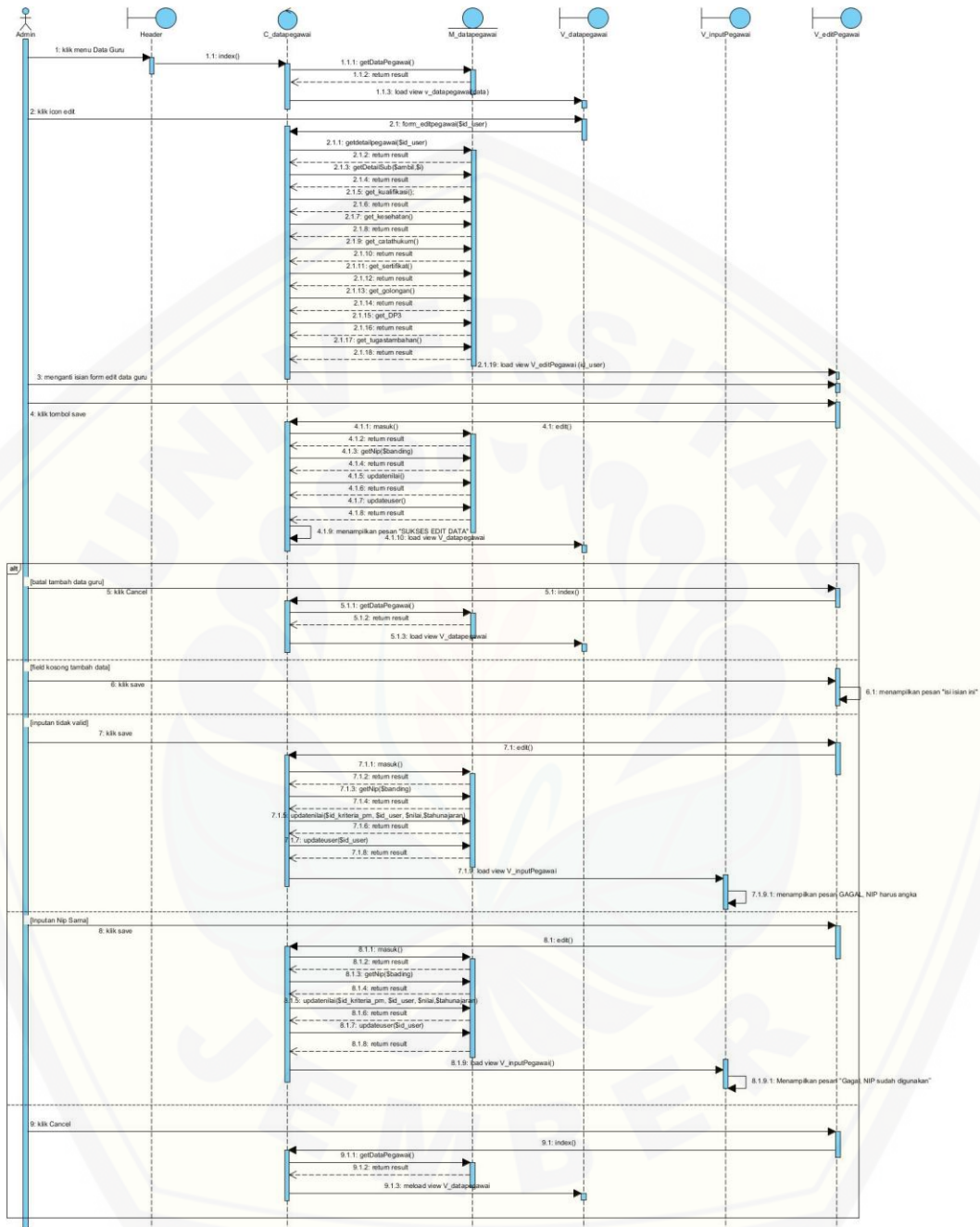
Gambar 4. 9 Sequence Diagram Login



Gambar 4. 11 Sequence Diagram Manajemen Data Guru (View))

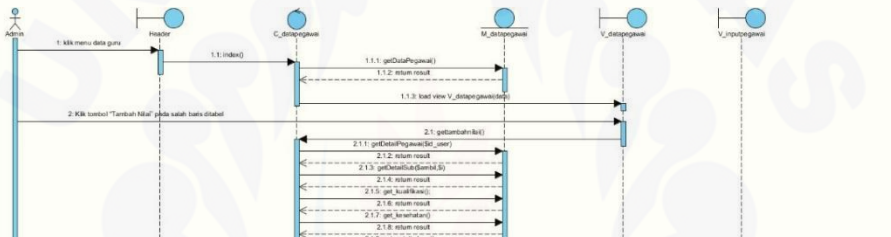
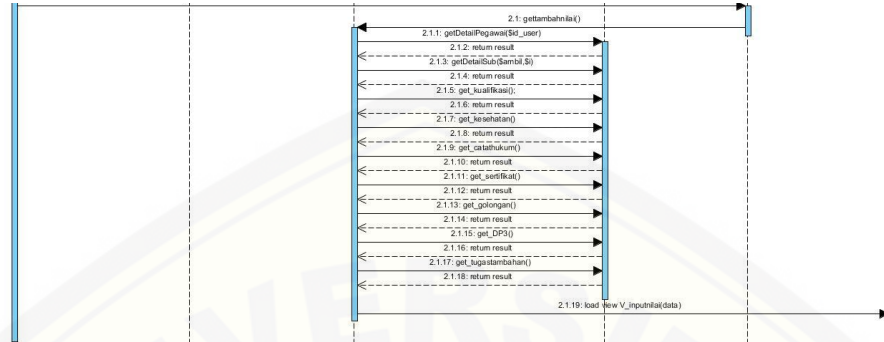


Gambar 4. 10 Sequence Diagram Manajemen Data Guru (Input)

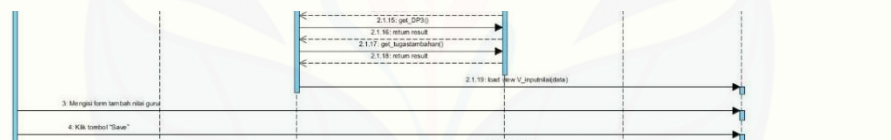




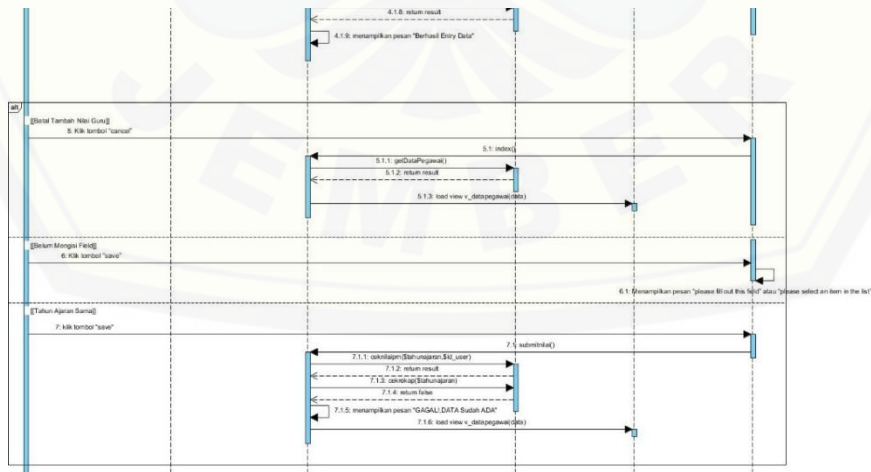
Gambar 4. 12 Sequence Diagram Manajemen Data Guru (Edit)



Gambar 4. 14 Sequence Diagram Manajemen Data Nilai Guru (Input)

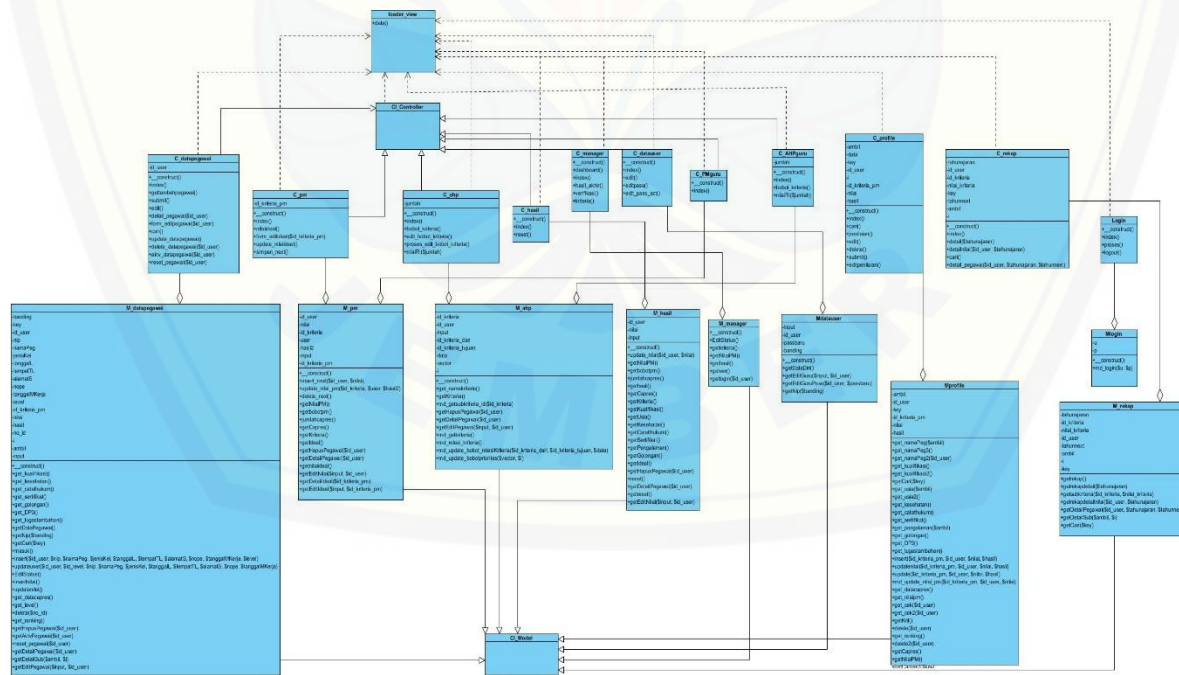


Gambar 4. 13 Sequence Diagram Manajemen Tambah Nilai Guru (View)



### 4.2.6 Class Diagram

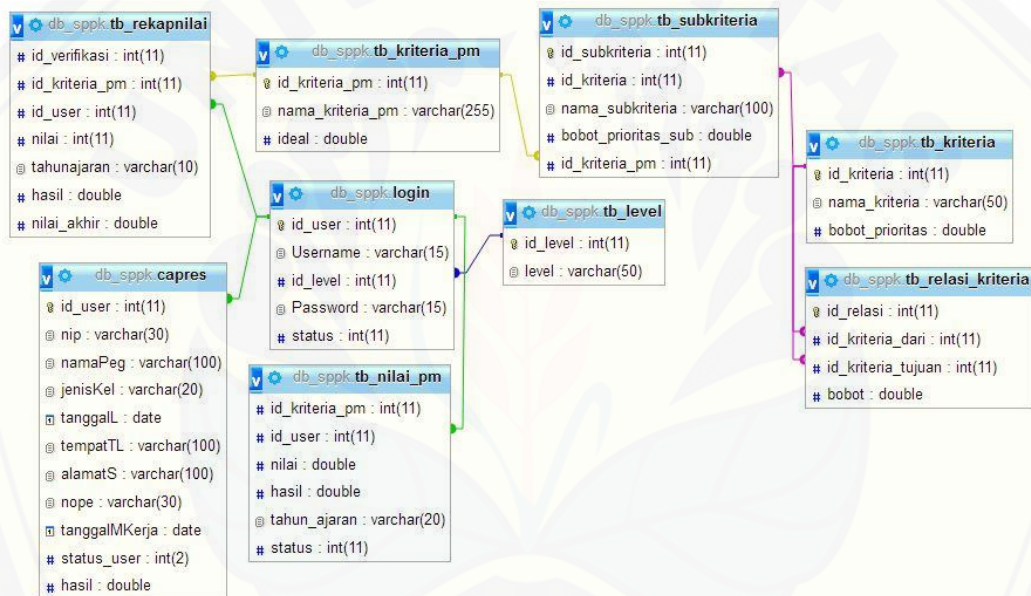
*Class diagram* menggambarkan hubungan antar kelas yang digunakan untuk membangun suatu sistem, kelas yang digambarkan berisi atribut dan nama method pada setiap kelasnya. Dalam paradigma OOD (*Object Oriented Desain*) terdapat 3 jenis kelas yaitu *model*, *view* dan *controller* dan memiliki berbagai macam relasi yang menggambarkan hubungan antar kelas. Berdasarkan *sequence* diagram yang telah dibangun, *class* diagram sistem Penilaian Prestasi Kerja Guru Sebagai Luran Pemilihan Calon Kepala Sekolah Menggunakan Metode *Profile Matching* – AHP terdiri dari 8 class model, yaitu : M\_datapegawai, M\_pm, M\_ahp, M\_hasil, Mdatauser, M\_profile, M\_rekap, dan M\_login, sementara view terdiri dari 50 class yang digambarkan dengan loader view dan 11 class *controller* yaitu C\_datapegawai, C\_pm, C\_ahp, C\_hasil, C\_manager, C\_datauser, C\_AHPguru, C\_profile, C\_rekap, dan Login. *Class* diagram sistem penilaian prestasi kerja guru sebagai luran pemilihan calon kepala sekolah menggunakan metode *Profile Matching* – ahp dapat dilihat pada Gambar 4.15



Gambar 4. 15 Class Diagram

#### 4.2.7 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) sistem penilaian prestasi kerja guru sebagai luaran pemilihan calon kepala sekolah menggunakan metode *Profile Matching* – AHP merupakan gambaran komponen dan struktur *database* yang saling berhubungan untuk digunakan dalam pembuatan sistem. ERD yang diimplementasikan pada sistem ini



terdiri dari sepuluh entitas yang dapat dilihat pada Gambar 4.16

Gambar 4. 16 Entity Relational Diagram

Gambar 4.16 menunjukkan ERD sistem penilaian prestasi kerja guru sebagai luaran pemilihan calon kepala sekolah menggunakan metode *Profile Matching* – AHP yang terdiri dari sembilan entitas. Sembilan entitas tersebut adalah *tb\_capres* yang berfungsi untuk menyimpan biodata profil guru, *tb\_rekapnilai* yang berfungsi untuk

menyimpan hasil rekap nilai yang didapat tiap tahun ajaran, `tb_kriteria_pm` berfungsi untuk menyimpan macam kriteria beserta nilai ideal tiap kriteria, `login` berfungsi untuk menyimpan hak akses username dan password masing-masing user untuk melakukan login, `tb_nilai_pm` berisi hasil nilai yang didapat berdasarkan hasil perhitungan *Profile Matching* pada masing-masing kriteria penilaian, `tb_subkriteria` berisi macam subkriteria dari kriteria penilaian, `tb_kriteria` berisi nama kriteria dan bobot prioritas masing-masing kriteria, `tb_relasi_kriteria` berfungsi untuk menyimpan alur relasi perbandingan tingkat kepetingan antara relasi kriteria satu dengan yang lain, dan `tb_level` yang berfungsi memberi tingkatan level user saat melakukan login sesuai hak akses masing-masing level.

### 4.3 Penulisan Kode Program

Desain yang telah dibuat akan diimplementasikan ke dalam kode program. Beberapa hal yang dilakukan dalam tahap implementasi antara lain:

- A. Penulisan kode program (*coding*) menggunakan bahasa pemrograman *Page HyperText Pre-Processor (PHP)*, *HyperText Markup Language (HTML)*, *Cascading Style Sheet (CSS)* dan dengan bantuan *framework Code Igniter (CI)*.
- B. Manajemen basisdata menggunakan *DBMS MySQL*.

Kode program perhitungan metode algoritma *Profile Matching* dan AHP terdapat di *class M\_pm* dan *M\_ahp* pada package *models*. Perhitungan metode *Profile Matching* dan AHP terdapat di *class C\_pm* dan *C\_ahp* pada package *controllers*. Penulisan kode program perhitungan metode AHP terdapat pada tabel 4.9 sebagai berikut :

Tabel 4. 9 Kode program function `bobot_kriteria()`

| No | Kode   |
|----|--|
| 29 | <code>public function bobot_kriteria() {</code>  |
| 30 | <code>  \$data['daftar_kriteria'] = \$this-&gt;M_ahp-&gt;md_getkriteria()-&gt;result();</code>     |
| 31 | <code>  \$data['relasi_kriteria'] = \$this-&gt;M_ahp-&gt;md_relasi_kriteria()-&gt;result();</code> |
| 32 | <code>  \$data['jumlahKriteria'] = count(\$data['daftar_kriteria']);</code>                        |
| 33 | <code>  \$jumlah = array();</code>   |



```

34 //jumlah itu nama untuk data kriterianya
35 //id kriteria dari itu (yg kanan) dan id kriteria tujuan (yg atas)
36 foreach ($data['daftar_kriteria'] as $kriteria) {
37     $jumlah1 = 0;
38     foreach ($data['relasi_kriteria'] as $relasi) {
39         //memfilter kriteria per kolom
40         if($relasi->id_kriteria_tujuan == $kriteria->id_kriteria){
41             $jumlah1 = $jumlah1 + $relasi->bobot;
42         }
43     }
44     array_push($jumlah, $jumlah1);
45 }
46 $data['jumlah_per_kolom'] = $jumlah;
47
48 //=====Menghitung nilai per cell (nilai cell / jumlah)=====
49 $arrayVector = array();
50 foreach ($data['daftar_kriteria'] as $kriteria) {
51     $nilai = array();
52     $a = 0;
53     foreach ($data['relasi_kriteria'] as $relasi) {
54         if($relasi->id_kriteria_dari == $kriteria->id_kriteria){
55             $nilai1 = $relasi->bobot / $jumlah[$a];
56             array_push($nilai, $nilai1);
57             $a++;
58         }
59     }
60
61 //=====Menghitung nilai vector=====
62 //BOBOT SINTESA
63 $vector = 0;
64 for($i = 0; $i < count($nilai); $i++){
65     $vector = $vector + $nilai[$i];
66 }
67 //EIGEN VECTOR
68 $vector = $vector / $data['jumlahKriteria'];
69 $vector = number_format($vector, 3, '.', '');
70 array_push($arrayVector, $vector);
71 for ($i=0; $i < 9 ; $i++) {
72     if(isset($arrayVector[$i])){
73         $this->M_ahp->md_update_bobotprioritas($arrayVector[$i], $i+1);
74     }
75 }

```

```

76     }
77
78     //=====Menghitung Nilai Lamda=====
79     $lamda = 0;
80     for($i = 0; $i < count($arrayVector); $i++){
81         $lamda = $lamda + ($arrayVector[$i] * $jumlah[$i]);
82     }
83
84     //=====Menghitung Nilai CI=====
85     $sci = ($lamda - $data['jumlahKriteria']) / ($data['jumlahKriteria'] - 1);
86     $sci = number_format($sci, 3, '.', '');
87
88     //=====Menghitung Nilai CR=====
89     $scr = $sci / $this->nilaiRI($data['jumlahKriteria']);
90     $scr = number_format($scr, 3, '.', '');
91
92     $data['nilai_lamda'] = $lamda;
93     $data['nilai_ci'] = $sci;
94     $data['nilai_cr'] = $scr;
95     $data['nilai_vector'] = $arrayVector;
96     $data['kriteria']=$this->M_ahp->md_getkriteria();
97     $this->load->view('admin/header');
98     $this->load->view('admin/V_ahp',$data);
99     $this->load->view('footer');
100    }

```

*Function* bobot\_kriteria\_manager() pada Tabel 4.9 merupakan kode program yang digunakan untuk melakukan perhitungan bobot pada kriteria dengan menggunakan metode AHP, dimana user memasukkan nilai kepentingan antar kriteria dan sistem memberikan respon dengan menghasilkan bobot prioritas. Pada kode program diatas tahap perhitungan metode AHP dimulai dari baris 33-46 yaitu menghitung jumlah setiap kolom dari nilai kepentingan yang telah dimasukkan sebelumnya, kemudian perhitungan dilanjutkan pada baris 49-59 yaitu menghitung nilai per-cell dengan pada masing-masing kriteria, kemudian dilanjutkan pada baris 63-76 yaitu menghitung nilai vektor, hasil dari tahap ini yaitu bobot prioritas yang akan disimpan atau diperbaharui, untuk tahap selanjutnya yaitu menghitung kekonsistenan dari nilai kepentingan yang

telah dimasukkan, dimulai dari menghitung nilai lamda yang ada pada baris 79-82, nilai CI pada baris 85-86 dan menghitung nilai CR pada baris 89-90. Nilai perbandingan dapat dinyatakan konsisten apabila hasil perhitungan dari CR tidak boleh lebih dari 0,01. Apabila nilai melebihi 0,01 maka dinyatakan tidak konsisten dan harus mengubah nilai kepentingan antar kriteria sampai nilai tersebut dapat dikatakan konsisten.

Tahap perhitungan untuk mengubah nilai bobot prioritas yang tidak konsisten terdapat pada *function* `edit_bobot_kriteria()` dan *function* `update_bobot_kriteria` dapat dilihat pada Tabel 4.10.

Tabel 4. 10 *Kode program function edit\_bobot\_kriteria()*

| No  | Kode   |
|-----|--|
| 103 | <code>public function edit_bobot_kriteria(){</code>  |
| 104 | <code>    \$data['daftar_kriteria'] = \$this-&gt;M_ahp-&gt;md_getkriteria()-&gt;result();</code>     |
| 105 | <code>    \$data['relasi_kriteria'] = \$this-&gt;M_ahp-&gt;md_relasi_kriteria()-&gt;result();</code> |
| 106 | <code>    \$this-&gt;load-&gt;view('admin/header');</code>   |
| 107 | <code>    \$data['jumlah'] = count(\$data['daftar_kriteria']);</code>                                |
| 108 | <code>    \$this-&gt;load-&gt;view('admin/V_admin_editkriteria', \$data);</code>                     |
| 109 |  |
| 110 | <code>    //\$this-&gt;load-&gt;view('admin/V_ahp',\$data);</code>                                   |
| 111 |  |
| 112 | <code>    \$this-&gt;load-&gt;view('footer');</code>   |
| 113 | <code>    }</code>   |
| 114 |  |
| 115 | <code>public function proses_edit_bobot_kriteria(){</code>   |
| 116 | <code>    // i mewakili id kriteria dari</code>  |
| 117 | <code>    // j mewakili id kriteria tujuan</code>  |
| 118 | <code>    for(\$i = 1; \$i &lt;= 9; \$i++){</code>   |
| 119 | <code>        for(\$j = 1; \$j &lt;= 9; \$j++){</code>   |

|     |   |
|-----|---|
| 120 | <code>\$data['bobot'] = \$this-&gt;input-&gt;post(\$i.\$j);</code>                  |
| 121 | <code>\$this-&gt;M_ahp-&gt;md_update_bobot_relasiKriteria(\$i, \$j, \$data);</code> |
| 122 | <code>}</code>  |
| 123 | <code>}</code>  |
| 124 | <code>redirect(site_url().'/C_ahp/bobot_kriteria?suksesupdate');</code>             |
| 125 | <code>}</code>  |

Kode program pada tabel 4.10 digunakan untuk mengubah nilai bobot prioritas dengan cara mengubah nilai kepentingan antar kriteria apabila terjadi nilai yang tidak konsisten atau jika nilai kepentingan hendak diubah, *function* `edit_bobot_kriteria()` digunakan untuk menampilkan halaman update bobot, sedangkan *function* `proses_edit_bobot_kriteria()` digunakan untuk menyimpan nilai kepentingan yang telah dimasukkan atau diinputkan pada halaman bobot kriteria sehingga diperoleh bobot prioritas yang baru dan konsisten

Tahap perhitungan metode AHP selanjutnya adalah pada *model* `m_produkivitas` yang terdapat pada *function* `md_relasi_kriteria()`, `md_update_bobot_relasiKriteria()`, `md_update_bobotprioritas()` dapat dilihat pada Tabel 4.11

Tabel 4. 11 *Kode program function md\_relasi\_kriteria()*

| No | Kode  |
|----|---|
| 55 | <code>function md_relasi_kriteria(){</code>   |
| 56 | <code>    \$this-&gt;db-&gt;select('*');</code>                                       |
| 57 | <code>    \$this-&gt;db-&gt;from('tb_relasi_kriteria');</code>                        |
| 58 | <code>    return \$this-&gt;db-&gt;get();</code>                                      |
| 59 | <code>}</code>  |
| 60 | <code>function md_update_bobot_relasiKriteria(\$id_kriteria_dari,</code>              |
| 61 | <code>    \$id_kriteria_tujuan, \$data){</code>                                       |
| 62 | <code>    \$this-&gt;db-&gt;where('id_kriteria_dari', \$id_kriteria_dari);</code>     |
| 63 | <code>    \$this-&gt;db-&gt;where('id_kriteria_tujuan', \$id_kriteria_tujuan);</code> |
| 64 | <code>    \$this-&gt;db-&gt;update('tb_relasi_kriteria', \$data);</code>              |
| 65 | <code>}</code>  |

```
66  
67  
68 function md_update_bobotprioritas($vector, $i){  
69     $query = $this->db->query("UPDATE `tb_kriteria`  
70         SET `bobot_prioritas` = '$vector'  
71         where id_kriteria = '$i'  
72         ");  
73     return $query;  
74 }
```

Kode program pada Tabel 4.11 adalah *function* md\_relasi\_kriteria() pada baris 55-59 yang digunakan untuk mengambil data relasi kriteria yang sudah tersimpan di *database*. *Function* md\_update\_bobot\_relasiKriteria() pada baris 60-65 digunakan untuk menyimpan bobot prioritas yang telah dihitung dengan cara memasukkan nilai kepentingan antar kriteria, sedangkan *function* md\_update\_bobotprioritas () pada baris 68-74 digunakan untuk menyimpan nilai kepentingan yang telah dilakukan dalam *database*. Untuk penulisan kode program metode *Profile Matching* dan lainnya terdapat pada Lampiran D (Penulisan Kode Program)

#### 4.4 Pengujian

Tahapan pengujian sistem merupakan suatu tahapan yang dilakukan secara sistematis untuk mengetahui ketidaksesuaian yang telah diimplementasikan dalam kode program dan rancangan sistem. Pengujian dilakukan dengan dua metode yaitu pengujian *White Box* dan *Black Box*. Hasil pengujian sebagai berikut:

##### 4.4.1 White Box

Pengujian sistem dengan metode *white box* dilakukan untuk menguji sistem dari segi desain dan kode program. Pengujian bertujuan untuk mengevaluasi apakah sistem mampu menghasilkan fungsi, inputan, dan keluaran yang sesuai dengan spesifikasi dari kebutuhan sistem. Tahapan pengujian metode *white box* meliputi:

### 1. *Listing* program

*Listing* program merupakan kumpulan baris kode yang diuji. Thapa *listing* program Setiap akan memberikan nomor ketika menjalankan *statement* atau penggunaan kondisi dalam program. *Listing* program pada bagian metode AHP dapat dilihat pada Gambar 4.17

### 2. Diagram alir

Diagram alir merupakan notasi yang digunakan untuk merepresentasikan aliran control yang digambarkan dari hasil penomoran dari *listing* program. Diagram alir digambarkan dengan *node* (simpul) yang dihubungkan dengan *edge* (garis). Diagram alir pada bagian metode dapat dilihat pada Gambar 4.18

### 3. Kompleksitas siklomatik (*cyclomatic complexity*)

Kompleksitas siklomatik merupakan metrik perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logis suatu program. Kompleksitas siklomatik mendefinisikan jumlah jalur independen dalam suatu program. Perhitungan kompleksitas siklomatik menggunakan persamaan 3 sebagai berikut :

$$V(G) = E - N + 2 \quad \dots \text{Persamaan 3}$$

Keterangan:

$V(G)$  = Kompleksitas siklomatik

$E$  = Jumlah *edge* (garis)

$N$  = Jumlah *node* (simpul)

Kompleksitas siklomatik pada bagian metode dapat dilihat pada sub bab C. Kompleksitas Siklomatik (Cyclomatic Complexity).

### 4. Jalur independen (*Independent Path*)

Jalur independen adalah setiap jalur yang melalui program, menunjukkan setidaknya satu kumpulan pernyataan-pernyataan pemrosesan atau kondisi baru. Jalur independen dalam grafik alir bergerak setidaknya satu *edge* yang belum dilintasi

sebelum jalur tersebut didefinisi. Jalur independen pada bagian metode dapat dilihat pada sub bab D. Jalur Independen (Independent Path).

#### 5. Pengujian Basis Set (*Test Case*)

Pengujian Basis Set menggambarkan pelaksanaan jalur. Data yang dieksekusi dimasukkan ke dalam grafik alir apakah sudah melewati basis set yang tersedia. Sistem telah memenuhi syarat kelayakan perangkat lunak jika salah satu jalur yang dieksekusi setidaknya satu kali. Pengujian Basis Set pada bagian metode dapat dilihat pada sub bab E. Pengujian Basis Set (Test Case).

Pengujian sistem penilaian prestasi kerja guru sebagai luaran pemilihan calon kepala sekolah menggunakan metode *Profile Matching – AHP* akan diterapkan pada fitur yang dinilai dapat mewakili sistem yaitu pada bagian metode sebagai berikut:

## A. Listing Program

```

29 public function bobot_kriteria() {
30 $data['daftar_kriteria'] = $this->M_ahp->md_getkriteria()->result();
31 $data['relasi_kriteria'] = $this->M_ahp->md_relasi_kriteria()->result();
32 $data['jumlahkriteria'] = count($data['daftar_kriteria']);
33 $jumlah = array();
34 //jumlah itu nama untuk data kriterianya
35 //id kriteria dari itu (yg kanan) dan id kriteria tujuan (yg atas)
36 foreach ($data['daftar_kriteria'] as $kriteria) {
37     $jumlah1 = 0;
38     foreach ($data['relasi_kriteria'] as $relasi) {
39         //memfilter kriteria per kolom
40         if($relasi->id_kriteria_tujuan == $kriteria->id_kriteria){
41             $jumlah1 = $jumlah1 + $relasi->bobot;
42         }
43     }
44     array_push($jumlah, $jumlah1);
45 }
46 $data['jumlah_per_kolom'] = $jumlah;
47
48 //=====Menghitung nilai per cell (nilai cell / jumlah)=====
49 $arrayVector = array();
50 foreach ($data['daftar_kriteria'] as $kriteria) {
51     $nilai = array();
52     $a = 0;
53     foreach ($data['relasi_kriteria'] as $relasi) {
54         if($relasi->id_kriteria_dari == $kriteria->id_kriteria){
55             $nilai1 = $relasi->bobot / $jumlah[$a];
56             array_push($nilai, $nilai1);
57             $a++;
58         }
59     }
60 }
61 //=====Menghitung nilai vector=====
62 //BOBOT SINTESA
63 $vector = 0;
64 for($i = 0; $i < count($nilai); $i++){
65     $vector = $vector + $nilai[$i];
66 }
67 //EIGEN VECTOR
68 $vector = $vector / $data['jumlahkriteria'];
69 $vector = number_format($vector, 3, '.', '');
70 array_push($arrayVector, $vector);
71 for ($i=0; $i < 9 ; $i++) {
72     if(isset($arrayVector[$i])){
73         $this->M_ahp->md_update_bobotprioritas($arrayVector[$i], $i+1);
74     }
75 }
76 }
77
78 //=====Menghitung Nilai Lamda=====
79 $lamda = 0;
80 for($i = 0; $i < count($arrayVector); $i++){
81     $lamda = $lamda + ($arrayVector[$i] * $jumlah[$i]);
82 }
83
84 //=====Menghitung Nilai CI=====
85 $ci = ($lamda - $data['jumlahkriteria']) / ($data['jumlahkriteria'] - 1);
86 $ci = number_format($ci, 3, '.', '');
87
88 //=====Menghitung Nilai CR=====
89 $cr = $ci / $this->nilaiRI($data['jumlahkriteria']);
90 $cr = number_format($cr, 3, '.', '');
91
92 $data['nilai_lamda'] = $lamda;
93 $data['nilai_ci'] = $ci;
94 $data['nilai_cr'] = $cr;
95 $data['nilai_vector'] = $arrayVector;
96 $data['kriteria']=$this->M_ahp->md_getkriteria();
97 $this->load->view('admin/header');
98 $this->load->view('admin/V_ahp',$data);
99 $this->load->view('footer');
100 }

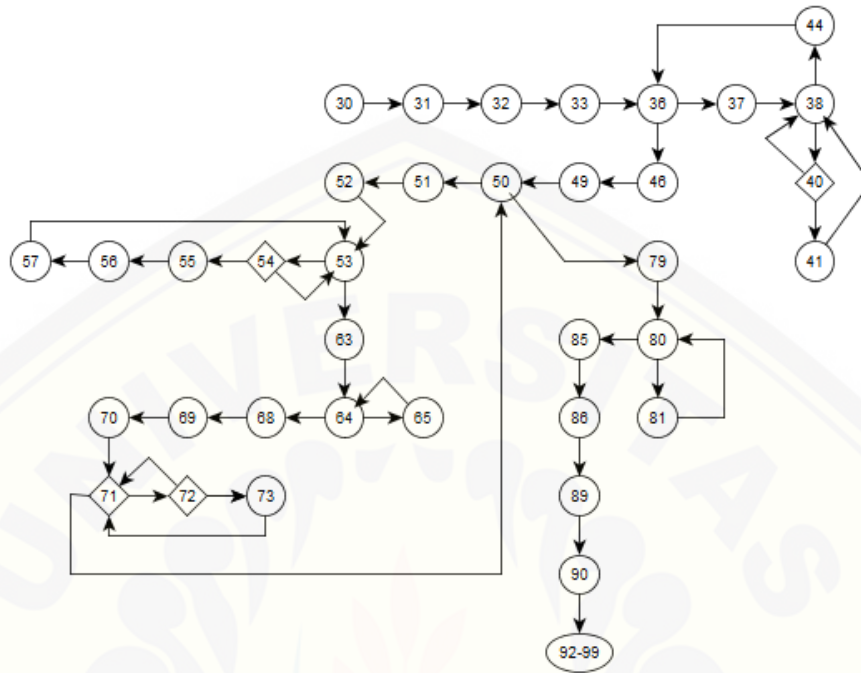
```



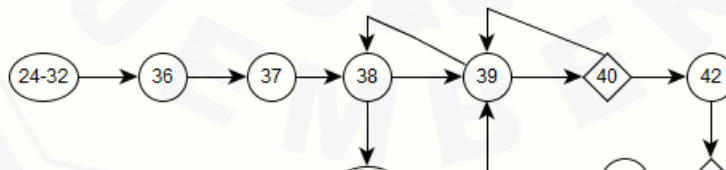
Gambar 4. 17 Listing program Metode AHP function bobot\_kriteria\_manager()

```
23 public function index(){
24     $data['kriteria'] = $this->M_pm->getKriteria();
25     $data['tahun'] = $this->M_pm->gettahun();
26     $data['jum_capres'] = $this->M_pm->jumlahcapres();
27     $data['jumlah_capres'] = count($data['jum_capres']);
28     $data['data_capres'] = $this->M_pm->getCapres();
29     $data['pm'] = $this->M_pm->getNilaiPM();
30     $data['ideal'] = $this->M_pm->getIdeal();
31     $data['jumlahkriteriapm'] = count($data['kriteria']);
32     $data['calon'] = count($data['pm'])/$data['jumlahkriteriapm'];
33     // $data['bobot'] = $this->M_pm->getbobotpm();
34     //foreach ($data['bobot'] as $gap){
35
36     $hasil = array();
37     $hasil_bobot = array();
38     foreach ($data['kriteria'] as $ideal){
39         foreach ($data['pm'] as $nilai){
40             if($nilai['id_kriteria_pm'] == $ideal['id_kriteria_pm']){//itu dikurangnya kriteranya sama
41                 // $kurang=0;
42                 $kurang=$nilai['nilai']-$ideal['ideal'];
43                 if ($kurang != 0){
44                     $hasil_nilai_pm= 5 + ($kurang*0.5);//gap nilai 1/2
45                 }else{
46                     $hasil_nilai_pm= 5;
47                 }
48                 array_push($hasil, $kurang);//
49                 array_push($hasil_bobot, $hasil_nilai_pm);
50                 $user = $nilai['id_user'];//tanya
51                 $id_kriteria = $nilai['id_kriteria_pm'];
52                 $hasil2 = $hasil_nilai_pm;
53                 $this->M_pm->update_nilai_pm($id_kriteria, $user, $hasil2);
54             }
55         }
56     }
57
58     $data['nilai_bobot'] = $hasil_bobot;
59     $data['jumlahbobot'] = count($data['nilai_bobot']);
60     $data['pengurangan']=$hasil;
61     $data['cetak_pm']=$hasil_bobot;
62
63
64     $this->load->view('admin/header');
65     $this->load->view('admin/V_pm',$data);
66     $this->load->view('footer');
67 }
```

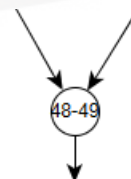
Gambar 4. 18 Listing program metode Profile Matching function index()



B. Diagram Alir



Gambar 4. 19 Diagram Alir Metode *Profile Matching Function index()*



Gambar 4. 20 Diagram Alir Metode AHP function bobot\_kriteria()

C. Kompleksitas Siklomatik (*Cyclomatic Complexity*)

a. Function `bobot_kriteria()`

$$V(G) = E - N + 2$$

$$V(G) = 16 - 13 + 2 = 5$$

b. Function `bobot_kriteria()`

$$V(G) = E - N + 2$$

$$V(G) = 46 - 37 + 2 = 11$$

D. Pengujian Basis Set (*Test Case*)

Tabel 4. 12 Test case function *metode Profile Matching index()*

*Test case function bobot\_kriteria\_manager()*

Jalur 1

|                        |      |  |
|------------------------|------|--|
| Test Case              |      | Jika berhasil menghitung nilai kinerja dengan nilai ideal maka akan ditampilkan hasil nilai pembobotan |
| Target yang diharapkan | yang | Menampilkan hasil nilai kinerja pembobotan masing-masing guru  |
| Hasil Pengujian        |      | Benar  |
| Path / jalur           |      | (24-32)-36-37-38-39-40-42-43-44-(48-49)-(50-53)- 39-40-42-43-44-(48-49)-(50-53)-39-38-(58-61)          |
| <b>Jalur 2</b>         |      |  |
| Test Case              |      | Jika berhasil menghitung nilai kinerja dengan nilai ideal maka akan ditampilkan hasil nilai pembobotan |
| Target yang diharapkan | yang | Menampilkan hasil nilai kinerja pembobotan masing-masing guru  |
| Hasil Pengujian        |      | Benar  |
| Path / jalur           |      | (24-32)-36-37-38-(58-61)   |
| <b>Jalur 3</b>         |      |  |
| Test Case              |      | Jika berhasil menghitung nilai kinerja dengan nilai ideal maka akan ditampilkan hasil nilai pembobotan |
| Target yang diharapkan | yang | Menampilkan hasil nilai kinerja pembobotan masing-masing guru  |
| Hasil Pengujian        |      | Benar  |
| Path / jalur           |      | (24-32)-36-37-38-39-40-39-38-(58-61)   |
| <b>Jalur 4</b>         |      |  |
| Test Case              |      | Jika berhasil menghitung nilai kinerja dengan nilai ideal maka akan ditampilkan hasil nilai pembobotan |
| Target yang diharapkan | yang | Menampilkan hasil nilai kinerja pembobotan masing-masing guru  |
| Hasil Pengujian        |      | Benar  |
| Path / jalur           |      | (24-32)-36-37-38-39-40-42-43-44-(48-49)-(50-53)-(58-61)  |

| Jalur 5                |      |  |
|------------------------|------|--|
| Test Case              |      | Jika berhasil menghitung nilai kinerja dengan nilai ideal maka akan ditampilkan hasil nilai pembobotan |
| Target yang diharapkan | yang | Menampilkan hasil nilai kinerja pembobotan masing-masing guru  |
| Hasil Pengujian        |      | Benar  |
| Path / jalur           |      | (23-32)-36-37-38-39-40-42-43-46-(48-49)-(50-53)  |

Tabel 4. 13 Test case function *metode AHP bobot\_kriteria\_manager()*

| <i>Test case function bobot_kriteria_manager()</i> |      |  |
|--|------|--|
| Jalur 1  |      |  |
| Test Case  |      | Jika berhasil menghitung bobot prioritas masing-masing kriteria, menghitung nilai $\lambda_{max}$ , CI, CR                               |
| Target yang diharapkan                             | yang | Menampilkan hasil perhitungan bobot prioritas kriteria hingga hasil pengujian konsistensi  |
| Hasil Pengujian                                    |      | Benar  |
| Path / jalur                                       |      | 30-31-32-33-36-37-38-40-41-38-44-36-46-49-50-51-52-53-54-55-56-57-53-63-64-65-64-68-69-70-71-72-73-71-50-79-80-81-80-85-86-89-90-(92-99) |
| Jalur 2  |      |  |
| Test Case  |      | Jika berhasil menghitung bobot prioritas masing-masing kriteria, menghitung nilai $\lambda_{max}$ , CI, CR.                              |
| Target yang diharapkan                             | yang | Menampilkan hasil perhitungan bobot prioritas kriteria hingga hasil pengujian konsistensi  |
| Hasil Pengujian                                    |      | Benar  |
| Path / jalur                                       |      | 30-31-32-33-36-37-38-40-38-44-36-46-49-50-51-52-53-54-55-56-57-53-63-64-65-64-68-69-70-71-72-73-71-50-79-80-81-80-85-86-89-90-(92-99)    |
| Jalur 3  |      |  |

|                        |   |
|------------------------|---|
| Test Case              | Jika berhasil menghitung bobot prioritas masing-masing kriteria, menghitung nilai $\lambda_{max}$ , CI, CR                      |
| Target yang diharapkan | Menampilkan hasil perhitungan bobot prioritas hingga hasil pengujian konsistensi  |
| Hasil Pengujian        | Benar   |
| Path / jalur           | 30-31-32-33-36-37-38-40-41-38-44-36-46-49-50-51-52-53-54-53-63-64-65-64-68-69-70-71-72-73-71-50-79-80-81-80-85-86-89-90-(92-99) |

#### Jalur 4

|                        |   |
|------------------------|---|
| Test Case              | Jika berhasil menghitung bobot prioritas masing-masing kriteria, menghitung nilai $\lambda_{max}$ , CI, CR                            |
| Target yang diharapkan | Menampilkan hasil perhitungan bobot prioritas hingga hasil pengujian konsistensi  |
| Hasil Pengujian        | Benar   |
| Path / jalur           | 30-31-32-33-36-37-38-40-41-38-44-36-46-49-50-51-52-53-54-55-56-57-53-63-64-65-64-68-69-70-71-72-71-50-79-80-81-80-85-86-89-90-(92-99) |

#### Jalur 5

|                        |  |
|------------------------|--|
| Test Case              | Jika berhasil menghitung bobot prioritas masing-masing kriteria, menghitung nilai $\lambda_{max}$ , CI, CR                   |
| Target yang diharapkan | Menampilkan hasil perhitungan bobot prioritas hingga hasil pengujian konsistensi   |
| Hasil Pengujian        | Benar  |
| Path / jalur           | 30-31-32-33-36-37-38-40-38-44-36-46-49-50-51-52-53-54-53-63-64-65-64-68-69-70-71-72-73-71-50-79-80-81-80-85-86-89-90-(92-99) |

#### Jalur 6

|           |  |
|-----------|--|
| Test Case | Jika berhasil menghitung bobot prioritas masing-masing kriteria, menghitung nilai $\lambda_{max}$ , CI, CR |
|-----------|--|

|                   |      |  |
|-------------------|------|--|
| Target diharapkan | yang | Menampilkan hasil perhitungan bobot prioritas hingga hasil pengujian konsistensi   |
| Hasil Pengujian   |      | Benar  |
| Path / jalur      |      | 30-31-32-33-36-37-38-40-41-38-44-36-46-49-50-51-52-53-54-55-56-57-53-63-64-65-64-68-69-70-71-50-79-80-81-80-85-86-89-90-(92-99)    |
| <b>Jalur 7</b>    |      |  |
| Test Case         |      | Jika berhasil menghitung bobot prioritas masing-masing kriteria, menghitung nilai $\lambda_{max}$ , CI, CR                         |
| Target diharapkan | yang | Menampilkan hasil perhitungan bobot prioritas hingga hasil pengujian konsistensi   |
| Path / jalur      |      | 30-31-32-33-36-37-38-40-38-44-36-46-49-50-51-52-53-54-55-56-57-53-63-64-65-64-68-69-70-71-72-71-50-79-80-81-80-85-86-89-90-(92-99) |
| <b>Jalur 8</b>    |      |  |
| Test Case         |      | Jika berhasil menghitung bobot prioritas masing-masing kriteria, menghitung nilai $\lambda_{max}$ , CI, CR                         |
| Target diharapkan | yang | Menampilkan hasil perhitungan bobot prioritas hingga hasil pengujian konsistensi   |
| Hasil Pengujian   |      | Benar  |
| Path / jalur      |      | 30-31-32-33-36-37-38-40-38-44-36-46-49-50-51-52-53-54-53-63-64-65-64-68-69-70-71-72-71-50-79-80-81-80-85-86-89-90-(92-99)          |
| <b>Jalur 9</b>    |      |  |
| Test Case         |      | Jika berhasil menghitung bobot prioritas masing-masing kriteria, menghitung nilai $\lambda_{max}$ , CI, CR                         |
| Target diharapkan | yang | Menampilkan hasil perhitungan bobot prioritas hingga hasil pengujian konsistensi   |
| Hasil Pengujian   |      | Benar  |

|                        |  |
|------------------------|--|
| Path / jalur           | 30-31-32-33-36-37-38-40-38-44-36-46-49-50-51-52-53-54-55-56-57-53-63-64-65-64-68-69-70-71-50-79-80-81-80-85-86-89-90-(92-99) |
| <b>Jalur 10</b>        |  |
| Test Case              | Jika berhasil menghitung bobot prioritas masing-masing kriteria, menghitung nilai $\lambda_{max}$ , CI, CR                   |
| Target yang diharapkan | Menampilkan hasil perhitungan bobot prioritas hingga hasil pengujian konsistensi   |
| Path / jalur           | 30-31-32-33-36-37-38-40-41-38-44-36-46-49-50-51-52-53-54-53-63-64-65-64-68-69-70-71-50-79-80-81-80-85-86-89-90-(92-99)       |
| <b>Jalur 8</b>         |  |
| Test Case              | Jika berhasil menghitung bobot prioritas masing-masing kriteria, menghitung nilai $\lambda_{max}$ , CI, CR                   |
| Target yang diharapkan | Menampilkan hasil perhitungan bobot prioritas hingga hasil pengujian konsistensi   |
| Hasil Pengujian        | Benar  |
| Path / jalur           | 30-31-32-33-36-37-38-40-38-44-36-46-49-50-51-52-53-54-53-63-64-65-64-68-69-70-71-50-79-80-81-80-85-86-89-90-(92-99)          |

#### 4.4.2 Black Box

*Black box testing* merupakan metode pengujian perangkat lunak yang memeriksa fungsionalitas dari aplikasi yang berkaitan dengan struktur internal atau kerja. Metode ini memfokuskan pada keperluan fungsionalitas dari software. Pengujian *black box* pada sistem penilaian prestasi kerja guru sebagai luaran pemilihan calon kepala sekolah menggunakan metode *Profile Matching* – AHP dilakukan untuk mengetahui apakah masukan dan keluaran dari sistem sesuai dengan kebutuhan fungsional atau tidak. Pengujian dilakukan pada setiap *use case*. Hasil pengujian *black box* dapat dilihat pada Lampiran E (Pengujian *Black Box*).



#### 4.4.3 Uji Validitas

Uji validitas merupakan pengujian yang dilakukan untuk membandingkan hasil metode dengan penerapan pada sistem. Perbandingan metode dan hasil implementasi sistem digunakan untuk mengetahui apakah hasil yang di dapat dalam perhitungan penilaian prestasi kerja guru sesuai, perbandingan tersebut di bahas pada bab 5 hasil dan pembahasan. Pada bab 5 terdapat hasil dari perhitungan metode profile matching, AHP, serta penggabungan perhitungan keduanya yang akan di bahas dengan nilai pada sistem itu sendiri, apakah konsisten atau tidak. Terdapat juga uji validitas secara manual. Uji validitas secara perhitungan ini dilakukan untuk menghitung tingkat kebenaran dari penentuan nilai masing-masing kriteria, apakah sebanding antara yang satu guru dengan guru yang lain sebagai pemilik nilai tiap kriteria penilaian, perhitungan uji validitas manual akan di jelaskan pada rumus perhitungan Product Moment di bawah ini :

$$r_{xy} = \frac{N\sum XY - (\sum X)(\sum Y)}{\sqrt{\{N\sum X^2 - (\sum X)^2\}\{N\sum Y^2 - (\sum Y)^2\}}}$$

Keterangan :

$$\sum X = \dots \sum Y = \dots \sum XY = \dots \sum X^2 = \dots \sum Y^2 = \dots n = \dots$$

X = Skor yang diperoleh subyek dari seluruh item

Y = Skor total yang diperoleh dari seluruh item

$\sum X$  = Jumlah skor dalam distribusi X

$\sum Y$  = Jumlah skor dalam distribusi Y

$\sum X^2$  = Jumlah kuadrat dalam skor distribusi X

$\sum Y^2$  = Jumlah kuadrat dalam skor distribusi Y

$N$  = Banyaknya responden

Berikut merupakan penerapan rumus perhitungan uji validitas pada sistem Penilaian Prestasi Kerja Guru Sebagai Luaran Pemilihan Calon Kepala Sekolah

- Masukkan semua skor yang di dapat masing-masing guru pada kriteria penilaian.
- Cari nilai  $\sum X$

| No | Nama     | Bobot Kualifikasi Akademik | Bobot Kesehatan | Bobot Sertifikat | Bobot Golongan | Bobot Usia (Tanggal Lahir) | Bobot Catatan Hukum | Bobot Pengalaman Mengajar (Tanggal Mulai Kerja) | Bobot Nilai DP3 | Bobot Tugas Tambahan |
|----|----------|----------------------------|-----------------|------------------|----------------|----------------------------|---------------------|---|-----------------|----------------------|
| 1  | Roni     | 3                          | 4               | 2                | 7              | 2                          | 3                   | 2   | 2               | 1                    |
| 2  | Tania    | 2                          | 1               | 2                | 5              | 3                          | 1                   | 1   | 2               | 0                    |
| 3  | Mika     | 2                          | 3               | 1                | 4              | 1                          | 2                   | 1   | 1               | 1                    |
|    | $\sum X$ | 7                          | 8               | 5                | 16             | 6                          | 6                   | 4   | 5               | 2                    |

- Cari nilai  $\sum Y$

| No | Nama  | Bobot Kualifikasi Akademik | Bobot Kesehatan | Bobot Sertifikat | Bobot Golongan | Bobot Usia (Tanggal Lahir) | Bobot Catatan Hukum | Bobot Pengalaman Mengajar (Tanggal Mulai Kerja) | Bobot Nilai DP3 | Bobot Tugas Tambahan | Jumlah |
|----|-------|----------------------------|-----------------|------------------|----------------|----------------------------|---------------------|---|-----------------|----------------------|--------|
| 1  | Roni  | 3                          | 4               | 2                | 7              | 2                          | 3                   | 2   | 2               | 1                    | 26     |
| 2  | Tania | 2                          | 1               | 2                | 5              | 3                          | 1                   | 1   | 2               | 0                    | 17     |
| 3  | Mika  | 2                          | 3               | 1                | 4              | 1                          | 2                   | 1   | 1               | 1                    | 16     |

|  |          |   |   |   |    |   |   |   |   |   |    |
|--|----------|---|---|---|----|---|---|---|---|---|----|
|  |          |   |   |   |    |   |   |   |   |   |    |
|  | $\sum X$ | 7 | 8 | 5 | 16 | 6 | 6 | 4 | 5 | 2 |    |
|  | $\sum Y$ |   |   |   |    |   |   |   |   |   | 59 |

d. Cari nilai  $(\sum X)^2$  yaitu dengan memangkatkan dua dari nilai  $\sum X$

| No | Nama         | Bobot Kualifikasi Akademik | Bobot Kesehatan | Bobot Sertifikat | Bobot Golongan | Bobot Usia (Tanggal Lahir) | Bobot Catatan Hukum | Bobot Pengalaman Mengajar (Tanggal Mulai Kerja) | Bobot Nilai DP3 | Bobot Tugas Tambahan | Jumlah |
|----|--------------|----------------------------|-----------------|------------------|----------------|----------------------------|---------------------|---|-----------------|----------------------|--------|
| 1  | Roni         | 3                          | 4               | 2                | 7              | 2                          | 3                   | 2   | 2               | 1                    | 26     |
| 2  | Tania        | 2                          | 1               | 2                | 5              | 3                          | 1                   | 1   | 2               | 0                    | 17     |
| 3  | Mika         | 2                          | 3               | 1                | 4              | 1                          | 2                   | 1   | 1               | 1                    | 16     |
|    | $\sum X$     | 7                          | 8               | 5                | 16             | 6                          | 6                   | 4   | 5               | 2                    |        |
|    | $\sum Y$     |                            |                 |                  |                |                            |                     |   |                 |                      | 59     |
|    | $(\sum X)^2$ | 49                         | 64              | 25               | 256            | 36                         | 36                  | 16  | 25              | 4                    |        |

e. Cari nilai  $\sum XY$  yaitu dengan mengalikan antara skor dengan jumlah skor

| No | Nama  | Bobot Kualifikasi Akademik | Bobot Kesehatan | Bobot Sertifikat | Bobot Golongan | Bobot Usia (Tanggal Lahir) | Bobot Catatan Hukum | Bobot Pengalaman Mengajar (Tanggal Mulai Kerja) | Bobot Nilai DP3 | Bobot Tugas Tambahan | Jumlah |
|----|-------|----------------------------|-----------------|------------------|----------------|----------------------------|---------------------|---|-----------------|----------------------|--------|
| 1  | Roni  | 3                          | 4               | 2                | 7              | 2                          | 3                   | 2   | 2               | 1                    | 26     |
| 2  | Tania | 2                          | 1               | 2                | 5              | 3                          | 1                   | 1   | 2               | 0                    | 17     |

|   |              |    |    |    |     |    |    |    |    |   |     |
|---|--------------|----|----|----|-----|----|----|----|----|---|-----|
| 3 | Mika         | 2  | 3  | 1  | 4   | 1  | 2  | 1  | 1  | 1 | 16  |
|   | $\sum X$     | 7  | 8  | 5  | 16  | 6  | 6  | 4  | 5  | 2 |     |
|   | $\sum Y$     |    |    |    |     |    |    |    |    |   | 59  |
|   | $(\sum X)^2$ | 49 | 64 | 25 | 256 | 36 | 36 | 16 | 25 | 4 |     |
|   | $\sum XY$    |    |    |    |     |    |    |    |    |   | 511 |

f. Cari nilai  $(\sum X.Y)$  (kalikan skor dengan jumlah skor)

| No | Nama          | Bobot Kualifikasi Akademik | Bobot Kesehatan | Bobot Sertifikat | Bobot Golongan | Bobot Usia (Tanggal Lahir) | Bobot Catatan Hukum | Bobot Pengalaman Mengajar (Tanggal Mulai Kerja) | Bobot Nilai DP3 | Bobot Tugas Tambahan |
|----|---------------|----------------------------|-----------------|------------------|----------------|----------------------------|---------------------|---|-----------------|----------------------|
| 1  | Roni          | 78                         | 104             | 52               | 182            | 52                         | 78                  | 52  | 52              | 26                   |
| 2  | Tania         | 34                         | 17              | 34               | 85             | 51                         | 17                  | 17  | 34              | 0                    |
| 3  | Mika          | 32                         | 48              | 16               | 64             | 16                         | 32                  | 16  | 16              | 16                   |
|    | <b>Jumlah</b> | 144                        | 169             | 102              | 331            | 119                        | 127                 | 85  | 102             | 42                   |

g. Cari nilai  $\sum X^2$

| No | Nama          | Bobot Kualifikasi Akademik | Bobot Kesehatan | Bobot Sertifikat | Bobot Golongan | Bobot Usia (Tanggal Lahir) | Bobot Catatan Hukum | Bobot Pengalaman Mengajar (Tanggal Mulai Kerja) | Bobot Nilai DP3 | Bobot Tugas Tambahan | Jumlah |
|----|---------------|----------------------------|-----------------|------------------|----------------|----------------------------|---------------------|---|-----------------|----------------------|--------|
| 1  | Roni          | 9                          | 16              | 4                | 49             | 4                          | 9                   | 4   | 4               | 1                    | 676    |
| 2  | Tania         | 4                          | 1               | 4                | 25             | 9                          | 1                   | 1   | 4               | 0                    | 289    |
| 3  | Mika          | 4                          | 9               | 1                | 16             | 1                          | 4                   | 1   | 1               | 1                    | 256    |
|    | <b>Jumlah</b> | 17                         | 26              | 9                | 90             | 14                         | 14                  | 6   | 9               | 2                    | 1221   |

h. Cari nilai  $\sum Y^2$

dengan menambahkan semua nilai  $\sum X.Y$ .

$$r_{xy} = \frac{N\sum XY - (\sum X)(\sum Y)}{\sqrt{\{N\sum X^2 - (\sum X)^2\}\{N\sum Y^2 - (N\sum Y)^2\}}}$$

$$\sum X = 7 \quad \sum Y = 59 \quad \sum X.Y = 144 \quad \sum X^2 = 17 \quad \sum Y^2 = 1221$$

$$r_{xy} = \frac{3(144) - (7)(59)}{\sqrt{\{3(17) - (7)^2\}\{3(1221) - (59)^2\}}} = 0,995912$$

Selanjutnya cocokkan antara nilai r hitung dengan nilai r product moment yang terdapat pada tabel 4.14

Tabel 4. 14 Tabel r Product Moment

| N<br>(1) | Interval<br>Kepercayaan |             | N<br>(1) | Interval<br>Kepercayaan |             | N<br>(1) | Interval<br>Kepercayaan |             |
|----------|-------------------------|-------------|----------|-------------------------|-------------|----------|-------------------------|-------------|
|          | 95 %<br>(2)             | 99 %<br>(3) |          | 95 %<br>(2)             | 99 %<br>(3) |          | 95 %<br>(2)             | 99 %<br>(3) |
|          | 3                       | 0,997       |          | 0,999                   | 27          |          | 0,381                   | 0,487       |
| 4        | 0,950                   | 0,990       | 28       | 0,374                   | 0,478       | 60       | 0,254                   | 0,330       |
| 5        | 0,878                   | 0,959       | 29       | 0,367                   | 0,470       | 65       | 0,244                   | 0,317       |
| 6        | 0,811                   | 0,917       | 30       | 0,361                   | 0,463       | 70       | 0,235                   | 0,306       |
| 7        | 0,754                   | 0,874       | 31       | 0,355                   | 0,456       | 75       | 0,227                   | 0,296       |
| 8        | 0,707                   | 0,874       | 32       | 0,349                   | 0,449       | 80       | 0,220                   | 0,286       |
| 9        | 0,666                   | 0,798       | 33       | 0,344                   | 0,442       | 85       | 0,213                   | 0,278       |

nilai r hitung untuk pertanyaan diatas adalah 0,999 dibandingkan dengan nilai r tabel atau nilai r product moment yaitu 3 untuk taraf kesalahan 5% yaitu sebesar 0,997. karena nilai r hitung > nilai r product moment yaitu 0,999 > 0,997 maka pertanyaan tersebut VALID.

## BAB 6. PENUTUP

Bab ini berisi mengenai kesimpulan dan saran dari peneliti tentang penelitian yang telah dilakukan. Kesimpulan dan saran tersebut diharapkan dapat digunakan sebagai acuan pada penelitian selanjutnya.

### 6.1 Kesimpulan

Kesimpulan yang didapat dari penelitian yang telah dilakukan adalah sebagai berikut :

1. Penerapan metode *Profile Matching-AHP* pada sistem penilaian prestasi kerja guru sebagai luaran pemilihan calon kepala sekolah digunakan untuk memberikan nilai bobot masing-masing kriteria penilaian kerja yang digunakan sebagai penentu nilai akhir peringkat hasil nilai prestasi kerja sebagai luaran pemilihan calon kepala sekolah. Langkah awal yakni melakukan penilaian kerja pada masing-masing guru di sekolah. Selanjutnya dari hasil penilaian kerja berdasarkan kriteria, maka akan didapatkan nilai yang nantinya akan digunakan untuk pemetaan gap. Pemetaan gap didapat dari selisih antara nilai ideal dengan nilai kerja yang didapat masing-masing guru berdasarkan kriteria. Langkah selanjutnya pembobotan gap berdasarkan hasil dari pemetaan gap. Setelah didapat pembobotan gap, maka dilanjutkan dengan melakukan pembobotan nilai prioritas tiap kriteria menggunakan metode AHP. Nilai prioritas kriteria dari AHP haruslah mendapat nilai CR di bawah 0,1%, nilai itu sama dengan CR yang di dapat dalam perhitungan di sistem ini, hal itu membuktikan jika perbandingan nilai bobot prioritas antar kriteria konsisten. Hasil dari pembobotan tiap kriteria prioritas pada metode AHP akan dikalikan dengan hasil nilai pembobotan gap yang didapat masing-masing guru sebelumnya di perhitungan *Profile Matching*. Nilai penggabungan ini selanjutnya di total sebagai nilai akhir yang akan berfungsi sebagai acuan dalam seleksi pemilihan

calon kandidat kepala sekolah berdasarkan hasil peringkat nilai tertinggi sampai terendah pada penilaian prestasi kerja guru.

2. Sistem penilaian prestasi kerja guru sebagai luaran pemilihan calon kepala sekolah menggunakan metode *Profile Matching* - AHP (studi kasus : SMA Negeri Grugujan Bondowoso) telah berhasil dibangun menggunakan model pengembangan *waterfall* dan berbasis *website*. Terdapat 3 hak akses dalam sistem ini, yaitu admin, Kepala Sekolah, dan guru. Pengembangan sistem ini menggunakan model *Waterfall* dimulai dengan tahapan analisis pengumpulan data kebutuhan fungsional dan non fungsional secara detail. Tahap selanjutnya adalah desain sistem yang digunakan sebagai acuan penulisan kode program. Setelah melakukan tahap desain, tahap selanjutnya adalah implementasi desain kedalam kode program. Tahap berikutnya yaitu pengujian sistem dilanjutkan yang terakhir adalah tahap pemeliharaan sistem secara berkala. Pada model *Waterfall* jika terjadi perubahan atau penambahan kebutuhan sistem, haruslah kembali ke tahap sebelumnya. Seperti yang terjadi pada saat pengerjaan di tahap pengkodean program, dimana pada saat itu harus menambahkan fitur rekap nilai prestasi kerja guru tiap tahun ajaran yang sebelumnya tidak ada, hal itu mengharuskan proses pengerjaan kembali ke penambahan analisis kebutuhan dan design sistem yang ikut berubah pula.

## 6.2 Saran

Adapun saran yang ditunjukkan untuk memberikan masukan yang lebih baik yaitu :

- a. Sistem Penilaian Prestasi Kerja Guru Sebagai Luarannya Pemilihan Calon Kepala Sekolah yang akan dikembangkan selanjutnya akan lebih baik jika jumlah kriteria bisa ditambahkan melalui sistem agar jika ada peraturan tentang penilaian prestasi guru yang berubah bisa disesuaikan langsung dengan peraturan yang baru.

- b. Sistem diharapkan ada pengembangan selanjutnya dengan memberi fitur yang lebih lengkap berupa fungsi aktor Guru yang dapat memasukkan softfile berupa informasi berkas terkait mengenai kriteria penilaian tiap tahunnya agar rekap data penilaian semakin terlihat objective dan lengkap.





**DAFTAR PUSTAKA**

- Agissa, W. (2013). White Box and Black Box testing. Diambil kembali dari <http://bangwildan.web.id/berita-176-white-box-testing--black-box-testing.html>
- Hanafi, Tyroni, Pranomo. (2011). *Sistem Pendukung Keputusan Penetapan Calon Peserta Sertifikasi Guru Sekolah Dasar Menggunakan Metode ANALYTICAL HIEARCHY PROCESS (AHP)*. Jurnal Teknologi Informasi, dan Komputer.
- Ima, Ermanu, Husniah. (2011). *Aplikasi Pendukung Pengambilan Keputusan Pemilihan Kepala Sekolah Menggunakan Metode Profile Matching*. Seminar Hasil Teknik Informatika UMM.
- Jogiyanto, H. M. (2004). *Pengenalan Komputer: Dasar Ilmu Komputer, Pemrograman, Sistem Informasi dan Intelegensi Buatan*. Yogyakarta: C.V ANDI OFFSET .
- Mufid. (2014). *Sistem Pendukung Keputusan Penilaian Proposal Kegiatan PNPM MPd Menggunakan Metode Profile Matching dan Analytic Hierarchy Process (AHP)*. Jurnal Sistem Informasi Bisnis.
- Peraturan Menteri Pendidikan Nasional. (2010). *Penugasan Guru Sebagai Kepala Sekolah/Madrasah Dengan Rahmat Tugan Yang Maha Esa Manusia*. Materi : Pendidikan Nasional.
- Pressman, R. S. (2002). *Rekayasa Perangkat Lunak Pendekatan Praktisi*. Yogyakarta: Andi.
- Republik Indonesia. (2003). *Undang- Undang Pendidik dan Tenaga Kependidikan Bab XI pasal 39, 40, dan 42*.

## LAMPIRAN

**Lampiran A. Scenario**

## A1. Usecase Manajemen Data User (Update)

Usecase skenario ini menjelaskan alur untuk proses mengupdate status aktif tidak aktifnya user guru login dan mereset password user guru jika mengalami lupa password. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario manajemen data user (update) dapat dilihat pada Tabel A1

Tabel A 1 Tabel Usecase Manajemen Data User (Update)

|                |   |
|----------------|---|
| Nomer Usecase  | USC-03                                    |
| Nama           | Usecase Manajemen Data User (Update)      |
| Aktor          | Admin                                     |
| Pre Condition  | Admin akan melakukan update ke data user  |
| Post Condition | Admin telah melakukan update ke data user |

| <b>SKENARIO DATA NORMAL</b><br><b>“Manajemen Data User (Update) Aktif Status User”</b> |   |
|--|---|
| 1. Klik menu “Data Guru”   |   |
|  | 2. Menampilkan halaman data daftar guru berupa nama pegawai dan status dari tabel capres di database, dengan menggunakan function index() |
| 3. Klik icon “centang √” pada salah satu baris   |   |
|  | 4. Mengubah attribute status_user di tabel capres pada database menggunakan function aktv_datapegawai() berdasarkan id_user yang pilih    |
|  | 5. Menampilkan pesan “User berhasil diaktifkan!”  |

|                             |
|-----------------------------|
| <b>SKENARIO DATA NORMAL</b> |
|-----------------------------|

| <b>“Manajemen Data User (Update) Nonaktif Status User”</b> |  |
|--|--|
| 1. Klik menu “Data Guru”                                   |  |
|  | 2. Menampilkan halaman data daftar guru berupa nama pegawai dan status dari tabel capres di database, dengan menggunakan function index()                          |
| 3. Klik icon “silang X” pada salah satu baris              |  |
|  | 4. Mengubah attribute status_user di tabel capres dan status di tabel login pada database menggunakan function delete_datapegawai() berdasarkan id_user yang pilih |
|  | 5. Menampilkan pesan “User berhasil dinonaktifkan!”  |

| <b>SKENARIO DATA NORMAL</b>                               |   |
|---|---|
| <b>“Manajemen Data User (Update) Reset Password User”</b> |   |
| 1. Klik menu “Data Guru”                                  |   |
|   | 2. Menampilkan halaman data daftar guru berupa nama pegawai dan status dari tabel capres di database, dengan menggunakan function index() |
| 3. Klik icon “refresh” pada salah satu baris              |   |
|   | 4. Menggubah attribute password pada tabel login di database menggunakan function reset_pegawai() berdasarkan id_user yang dipilih        |
|   | 5. Menampilkan pesan “Password Berhasi Direset dengan Hasil = 1234”   |

## A2. Usecase Manajemen Data Pribadi Guru

### a. Usecase Manajemen Data Pribadi Guru (view)

Usecase skenario ini menjelaskan alur untuk proses melihat data pribadi guru pada masing-masing guru. Aktor yang dapat melihat data pribadi ini adalah semua guru yang telah memiliki user login. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario manajemen data pribadi guru (view) dapat dilihat pada tabel A2

Tabel A 2 Usecase Skenario Manajemen Data Pribadi Guru (View)

|                |  |
|----------------|--|
| Nomer Usecase  | USC-04   |
| Nama           | Usecase Manajemen Data Pribadi Guru (View)     |
| Aktor          | Kepala Sekolah dan Guru                        |
| Pre Condition  | Guru akan melakukan view ke data pribadi guru  |
| Post Condition | Guru telah melakukan view ke data pribadi guru |

| <b>SKENARIO DATA NORMAL</b><br><b>“Manajemen Data Pribadi Guru (View)”</b> |   |
|--|---|
| 1. Klik menu “Biodata Pribadi”   |   |
|  | 2. Menampilkan halaman Biodata Pribadi Guru dengan data NIP, nama, jenis kelamin, tempat tanggal lahir, alamat, dan no hp yang berasal dari tabel capres dan login menggunakan function index() |

b. Usecase Manajemen Data Pribadi Guru (Edit)

Usecase skenario ini menjelaskan alur untuk proses edit data pribadi guru pada masing-masing guru jika terjadi kesalahan saat penginputan data guru oleh Admin. Aktor yang dapat mengedit data pribadi ini adalah guru yang memiliki status user aktif. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario manajemen data pribadi guru (edit) dapat dilihat pada tabel A3

Tabel A 3 Usecase Skenario Manajemen Data Pribadi Guru (Edit)

|                |  |
|----------------|--|
| Nomer Usecase  | USC-04   |
| Nama           | Usecase Manajemen Data Pribadi Guru (Edit)     |
| Aktor          | Guru   |
| Pre Condition  | Guru akan melakukan edit ke data pribadi guru  |
| Post Condition | Guru telah melakukan edit ke data pribadi guru |

| <b>SKENARIO DATA NORMAL<br/>“Manajemen Pribadi Guru (Edit)”</b> |   |
|---|---|
| 1. Klik menu “Biodata Pribadi”                                  |   |
|   | 2. Menampilkan halaman Biodata Pribadi Guru dengan data NIP, nama, jenis kelamin, tempat tanggal lahir, alamat, dan no hp yang berasal dari tabel capres dan login menggunakan function index() |
| 3. Klik tombol “Edit Biodata”                                   |   |
|   | 4. Menampilkan form pop up edit biodata berisi NIP, nama, jenis kelamin, tempat tanggal lahir, alamat dan no hp berasal dari tabel capres dan login menggunakan function index()                |
| 5. Menggubah isian form edit biodata                            |   |
| 6. Klik tombol “Save”   |   |
|   | 7. Menggubah data attribute berupa bio, namaPeg, jenisKel, tanggalL, tempatTL, alamatS, dan nope dalam tabel capres di database berdasarkan id_user yang dipilih menggunakan function edit()    |
|   | 8. Menampilkan pesan “Berhasil, Data Berhasil di Update”  |
|   | 9. Menampilkan halaman Biodata Pribadi Guru dengan data NIP, nama, jenis kelamin, tempat tanggal lahir, alamat, dan no hp yang berasal dari tabel capres  |

|   |  |
|---|--|
|   | dan login menggunakan function index()   |
| <b>SKENARIO ALTERNATIF<br/>“Cancel”</b>                   |  |
| 5a. Menggubah isian form edit Biodata                     |  |
| 6a. Klik tombol “cancel”                                  |  |
|   | 7a. Menampilkan halaman Biodata Pribadi Guru dengan data NIP, nama, jenis kelamin, tempat tanggal lahir, alamat, dan no hp yang berasal dari tabel capres dan login menggunakan function index() |
| <b>SKENARIO ALTERNATIF<br/>“Belum Mengisi Field”</b>      |  |
| 6b. Klik tombol “save”                                    |  |
|   | 7b. Menampilkan pesan “Isi isian ini”  |
| <b>SKENARIO ALTERNATIF<br/>“Inputan Tidak Valid”</b>      |  |
| 6c. Klik tombol “save”                                    |  |
|   | 7c. Menampilkan pesan “Gagal, NIP Harus Angka!” (jika NIP salah tipe data) atau pesan “Gagal, Nomer Harus Angka!” (jika nomer HP salah tipe data)  |
| <b>SKENARIO ALTERNATIF<br/>“NIP yang Diinputkan Sama”</b> |  |
| 6d. Klik tombol “save”                                    |  |
|   | 7d. Menampilkan pesan “Gagal, NIP sudah digunakan”   |

### A3. Usecase Update Password Pribadi

Usecase skenario ini menjelaskan alur untuk proses mengedit password pribadi masing-masing guru yang semula diatur default sama oleh admin, menjadi sesuai keinginan pribadi masing-masing guru sehingga tingkat kerahasiaan semakin

aman. Aktor yang data mengupdate password ini adalah guru pada masing-masing akun pribadi mereka. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario update password pribadi dapat dilihat pada tabel A4

Tabel A 4 Usecase Skenario Update Password Pribadi (Edit)

|                |   |
|----------------|---|
| Nomer Usecase  | USC-05  |
| Nama           | Usecase Update Password Pribadi (Edit)            |
| Aktor          | Guru  |
| Pre Condition  | Guru akan melakukan update pada password pribadi  |
| Post Condition | Guru telah melakukan update pada password pribadi |

| <b>SKENARIO DATA NORMAL<br/>“Update Password Pribadi(Edit)”</b> |   |
|---|---|
| 1. Klik menu “Biodata Pribadi”                                  |   |
|   | 2. Menampilkan halaman Biodata Pribadi Guru dengan data NIP, nama, jenis kelamin, tempat tanggal lahir, alamat, dan no hp yang berasal dari tabel capres dan login menggunakan function index() |
| 3. Klik tombol “Ganti Password”                                 |   |
|   | 4. Menampilkan halaman Ganti Password   |
| 5. Mengisi halaman form ganti password                          |   |
| 6. Klik tombol “Save”   |   |
|   | 7. Menampilkan pesan “Password Berhasil Diganti”  |
|   | 8. Menampilkan halaman Biodata Pribadi  |
| <b>SKENARIO ALTERNATIF<br/>“Belum Mengisi Field”</b>            |   |
| 6b. Klik tombol “save”  |   |
|   | 7b. Menampilkan pesan “Isi isian ini”   |
| <b>SKENARIO ALTERNATIF<br/>“Password Lama Salah”</b>            |   |

|   |   |
|---|---|
| 6c. Klik tombol “save”                                    |   |
|   | 7c. Menampilkan pesan “Password Lama Yang Anda Masukkan Salah”      |
| <b>SKENARIO ALTERNATIF<br/>“Password Baru Tidak Sama”</b> |   |
| 6c. Klik tombol “save”                                    |   |
|   | 7c. Menampilkan pesan “Password Baru Yang Anda Masukkan Tidak Sama” |

#### A4. Usecase Melihat Kriteria Penilaian Kinerja Guru

##### a. Usecase Melihat Kriteria Penilaian Kinerja Guru (view)

Usecase skenario ini menjelaskan alur untuk proses melihat data kriteria apa saja yang digunakan dalam penilaian kinerja prestasi guru. Aktor yang dapat melihat ini adalah Kepala Sekolah. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario melihat kriteria penilaian kinerja guru (view) dapat dilihat pada tabel A5

Tabel A 5 Usecase Skenario Melihat Kriteria Penilaian Kinerja Guru (View)

|                |  |
|----------------|--|
| Nomer Usecase  | USC-06   |
| Nama           | Usecase Melihat Kriteria Penilaian Kinerja Guru (View)                 |
| Aktor          | Kepala Sekolah   |
| Pre Condition  | Kepala Sekolah akan melakukan view ke kriteria penilaian kinerja guru  |
| Post Condition | Kepala Sekolah telah melakukan view ke kriteria penilaian kinerja guru |

|   |   |
|---|---|
| <b>SKENARIO DATA NORMAL<br/>“Melihat Kriteria Penilaian Kinerja Guru(View)”</b> |   |
| 1. Klik menu “Kriteria Penilaian”   |   |
|   | 2. Menampilkan halaman Kriteria Penilaian Kinerja |



#### A5. Usecase Melihat Nilai Kinerja Guru

##### a. Usecase Melihat Nilai Kinerja Guru (view)

Usecase skenario ini menjelaskan alur untuk proses melihat data nilai guru pada masing-masing guru yang telah dinilai. Aktor yang dapat melihat nilai kinerja guru ini adalah guru dan Kepala Sekolah. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario melihat nilai kinerja (view) dapat dilihat pada tabel A6

Tabel A 6 Usecase Skenario Melihat Nilai Kinerja (View)

|                |  |
|----------------|--|
| Nomer Usecase  | USC-07   |
| Nama           | Usecase Melihat Nilai Kinerja (View)                               |
| Aktor          | Guru dan Kepala Sekolah  |
| Pre Condition  | Guru dan Kepala Sekolah akan melakukan view ke nilai kinerja guru  |
| Post Condition | Guru dan Kepala Sekolah telah melakukan view ke nilai kinerja guru |

| <b>SKENARIO DATA NORMAL<br/>“Melihat Nilai Kinerja (View)”</b> |   |
|--|---|
| 1. Klik menu “Nilai Kinerja”                                   | 2. Menampilkan halaman Nilai Kinerja Guru |

#### A6. Usecase Manajemen Nilai Ideal Profile Matching

##### a. Usecase Manajemen Nilai Ideal Profile Matching (view)

Usecase skenario ini menjelaskan alur untuk proses melihat nilai ideal profile matching. Aktor yang dapat melihat nilai ideal Profile Matching ini adalah Admin. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario manajemen nilai ideal profile matching (view) dapat dilihat pada tabel A7

Tabel A 7 Usecase Skenario Manajemen Nilai Ideal Profile Matching (View)

|                |  |
|----------------|--|
| Nomer Usecase  | USC-8  |
| Nama           | Usecase Manajemen Nilai Ideal Profile Matching (View)      |
| Aktor          | Admin  |
| Pre Condition  | Admin akan melakukan view ke nilai ideal profile matching  |
| Post Condition | Admin telah melakukan view ke nilai ideal profile matching |

| <b>SKENARIO DATA NORMAL<br/>“Manajemen Nilai Ideal Profile Matching (View)”</b> |  |
|---|--|
| 1. Klik menu “Perhitungan Profile”  |  |
|   | 2. Menampilkan halaman Data Profile Matching |
| 3. Klik tombol “Update Nilai Ideal”   |  |
|   | 4. Menampilkan halaman nilai kinerja ideal   |

b. Usecase Manajemen Nilai Ideal Profile Matching (Edit)

Usecase skenario ini menjelaskan alur untuk proses mengedit nilai ideal profil matching jika ada perubahan pada kebijakan penilaian kinerja guru. Aktor yang dapat mengedit nilai ideal Profil Matching ini adalah Admin. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario manajemen nilai ideal profile matching (edit) dapat dilihat pada tabel A8

Tabel A 8 Usecase Skenario Manajemen Nilai Ideal Profile Matching (Edit)

|               |   |
|---------------|---|
| Nomer Usecase | USC-8   |
| Nama          | Usecase Manajemen Nilai Ideal Profile Matching (Edit)     |
| Aktor         | Admin   |
| Pre Condition | Admin akan melakukan edit ke nilai ideal profile matching |

|                |  |
|----------------|--|
| Post Condition | Admin telah melakukan edit ke nilai ideal profile matching |
|----------------|--|

| <b>SKENARIO DATA NORMAL</b><br><b>“Manajemen Nilai Ideal Profile Matching (Edit)”</b> |   |
|---|---|
| 1. Klik menu “Perhitungan Profile”  |   |
|   | 2. Menampilkan halaman Data Profile Matching            |
| 3. Klik tombol “Update Nilai Ideal”   |   |
|   | 4. Menampilkan halaman nilai kinerja ideal              |
| 5. Klik tombol “Edit” pada salah satu baris   |   |
|   | 6. Menampilkan halaman edit nilai ideal                 |
| 7. Menganti isian form edit nilai ideal   |   |
| 8. Klik tombol save   |   |
|   | 9. Menampilkan pesan “Berhasil, Data Berhasil Disimpan” |
| <b>SKENARIO ALTERNATIF</b><br><b>“Cancel”</b>   |   |
| 7a. Menganti isian form nilai ideal   |   |
| 8a. Klik tombol “Batal”   |   |
|   | 7a. Menampilkan halaman nilai kinerja Ideal             |
| <b>SKENARIO ALTERNATIF</b><br><b>“Belum Mengisi Field”</b>                            |   |
| 8b. Kik tombol “save”   |   |
|   | 9b. Menampilkan pesan “Isi Isian ini”                   |
| <b>SKENARIO ALTERNATIF</b><br><b>“Salah Tipe Data”</b>                                |   |
| 8c. Kik tombol “save”   |   |
|   | 9c. Menampilkan pesan “GAGAL! Nilai Ideal Harus Angka”  |

## A7. Usecase Manajemen Bobot Kriteria AHP

## a. Usecase Manajemen Bobot Kriteria AHP(View)

Usecase skenario ini menjelaskan alur untuk proses melihat bobot kriteria AHP. Aktor yang dapat melihat bobot kriteria AHP ini adalah Admin. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario manajemen bobot kriteria AHP(view) dapat dilihat pada tabel A9

Tabel A 9 Usecase Skenario Manajemen Bobot Kriteria AHP (View)

|                |  |
|----------------|--|
| Nomer Usecase  | USC-10   |
| Nama           | Usecase Manajemen Bobot Kriteria AHP (View)      |
| Aktor          | Admin  |
| Pre Condition  | Admin akan melakukan view ke Bobot Kriteria AHP  |
| Post Condition | Admin telah melakukan view ke Bobot Kriteria AHP |

| <b>SKENARIO DATA NORMAL</b><br><b>“Manajemen Bobot Kriteria AHP (View)”</b> |  |
|---|--|
| 1. Klik menu “Perhitungan AHP”  |  |
|   | 2. Menampilkan halaman Perhitungan AHP |

## b. Usecase Manajemen Bobot Kriteria AHP (Edit)

Usecase skenario ini menjelaskan alur untuk proses mengedit bobot kriteria AHP. Aktor yang dapat mengedit bobot kriteria AHP ini adalah Admin. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario manajemen bobot kriteria AHP (edit) dapat dilihat pada tabel A10

Tabel A 10 Usecase Skenario Manajemen Nilai Kinerja Profile Matching (Edit)

|               |   |
|---------------|---|
| Nomer Usecase | USC-10  |
| Nama          | Usecase Manajemen Bobot Kriteria AHP (Edit)     |
| Aktor         | Admin   |
| Pre Condition | Admin akan melakukan edit ke Bobot Kriteria AHP |

|                |  |
|----------------|--|
| Post Condition | Admin telah melakukan edit ke Bobot Kriteria AHP |
|----------------|--|

| <b>SKENARIO DATA NORMAL<br/>“Manajemen Bobot Kriteria AHP (Edit)”</b> |   |
|---|---|
| 1. Klik menu “Perhitungan AHP”  |   |
|   | 2. Menampilkan halaman Perhitungan AHP                  |
| 3. Klik tombol “Update”   |   |
|   | 4. Menampilkan halaman Update Bobot Kriteria            |
| 5. Mengisi form Update Bobot Nilai AHP                                |   |
| 6. Klik tombol “Save”   |   |
|   | 7. Menampilkan pesan “Berhasil, Data Berhasil Diupdate” |
| <b>SKENARIO ALTERNATIF<br/>“Cancel”</b>                               |   |
| 5a. Menganti isian form update bobot Kriteria                         |   |
| 6a. Klik tombol “Batal”   |   |
|   | 7a. Menampilkan halaman Perhitungan AHP                 |

#### A8. Usecase Melihat Nilai Kinerja Profile Matching (View)

##### a. Usecase Melihat Nilai Kinerja Profile Matching (View)

Usecase skenario ini menjelaskan alur untuk proses melihat nilai kinerja dari hasil perhitungan profile mathcing. Aktor yang dapat melihat nilai kinerja profile matching ini adalah Admin dan Kepala Sekolah. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario melihat nilai kinerja profile matching (view) dapat dilihat pada tabel A11

Tabel A 11 Usecase Skenario Melihat Nilai Kinerja Profile Matching (View)

|               |        |
|---------------|--------|
| Nomer Usecase | USC-10 |
|---------------|--------|

|                |  |
|----------------|--|
| Nama           | Usecase Melihat Nilai Kinerja Profile Matching (View)                          |
| Aktor          | Admin dan Kepala Sekolah   |
| Pre Condition  | Admin dan Kepala Sekolah akan melakukan view ke nilai kinerja Profile Matching |
| Post Condition | Admin telah melakukan view ke nilai kinerja Profile Matching                   |

| <b>SKENARIO DATA NORMAL</b><br><b>“Melihat Nilai Kinerja Profile Matching (View)”</b> |  |
|---|--|
| 1. Klik menu “Perhitungan Profile”  |  |
|   | 2. Menampilkan halaman Data Profile Matching |

#### A9. Usecase Melihat Bobot Kriteria AHP

##### a. Usecase Melihat Bobot Kriteria AHP(View)

Usecase skenario ini menjelaskan alur untuk proses melihat bobot kriteria AHP. Aktor yang dapat melihat bobot kriteria AHP ini adalah Kepala Sekolah. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario melihat bobot kriteria AHP(view) dapat dilihat pada tabel A12

Tabel A 12 Usecase Skenario Melihat Bobot Kriteria AHP (View)

|                |   |
|----------------|---|
| Nomer Usecase  | USC-11  |
| Nama           | Usecase Melihat Bobot Kriteria AHP (View)                 |
| Aktor          | Kepala Sekolah  |
| Pre Condition  | Kepala Sekolah akan melakukan view ke Bobot Kriteria AHP  |
| Post Condition | Kepala Sekolah telah melakukan view ke Bobot Kriteria AHP |

| <b>SKENARIO DATA NORMAL</b><br><b>“Melihat Bobot Kriteria AHP (View)”</b> |  |
|---|--|
|---|--|

|                                |  |
|--------------------------------|--|
| 1. Klik menu “Perhitungan AHP” |  |
|                                | 2. Menampilkan halaman Perhitungan AHP |

#### A10. Usecase Melihat Peringkat Nilai Kinerja

Usecase skenario ini menjelaskan alur untuk proses melihat hasil peringkat nilai kinerja guru dari nilai tertinggi sampai terendah urut berdasarkan pengabungan perhitungan profile matching dan AHP. Aktor yang dapat melihat hasil penentuan peringkat ini adalah Admin. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario Melihat Peringkat Nilai Kinerja (view) dapat dilihat pada tabel A13

Tabel A 13 Usecase Skenario A14 Melihat Peringkat Nilai Kinerja (View)

|                |   |
|----------------|---|
| Nomer Usecase  | USC-12  |
| Nama           | Usecase Melihat Peringkat Nilai Kinerja (View)        |
| Aktor          | Admin   |
| Pre Condition  | Admin akan melakukan view ke peringkat nilai kinerja  |
| Post Condition | Admin telah melakukan view ke peringkat nilai kinerja |

| <b>SKENARIO DATA NORMAL<br/>“Melihat Peringkat Nilai Kinerja(View)”</b> |  |
|---|--|
| 1. Klik menu “Hasil Peringkat”  |  |
|   | 2. Menampilkan Halaman Hasil Penentuan Peringkat |

#### A11. Usecase Manajemen Verifikasi Peringkat Nilai Kinerja

##### a. Usecase Manajemen Verifikasi Peringkat Nilai Kinerja (view)

Usecase skenario ini menjelaskan alur untuk proses melihat peringkat nilai kinerja semua yang nanti akan dilakukan verifikasi disana. Aktor yang dapat melihat peringkat nilai kinerja ini adalah kepala sekolah. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario

manajemen verifikasi peringkat nilai kinerja (view) dapat dilihat pada tabel A14

Tabel A 14 Usecase Skenario Menejaman Verifikasi Peringkat Nilai Kinerja (View)

|                |   |
|----------------|---|
| Nomer Usecase  | USC-13  |
| Nama           | Usecase Menejaman Verifikasi Peringkat Nilai Kinerja (View)               |
| Aktor          | Kepala Sekolah  |
| Pre Condition  | Kepala Sekolah akan melakukan view ke verifikasi peringkat nilai kinerja  |
| Post Condition | Kepala Sekolah telah melakukan view ke verifikasi peringkat nilai kinerja |

| <b>SKENARIO DATA NORMAL</b><br><b>“Manajemen Verifikasi Peringkat Nilai Kinerja (View)”</b> |   |
|---|---|
| 1. Klik menu “Nilai Kinerja”  |   |
|   | 2. Menampilkan halaman Nilai Kinerja Guru           |
| 3. Klik tombol “Peringkat Nilai Kinerja Guru”   |   |
|   | 4. Menampilkan halaman peringkat nilai kinerja guru |

b. Usecase Menejaman Verifikasi Peringkat Nilai Kinerja (Update)

Usecase skenario ini menjelaskan alur untuk proses melakukan update verifikasi peringkat nilai kinerja guru yang telah dikerjakan oleh Admin sehingga akan muncul dihalaman login masing-masing user pengguna guru. Aktor yang dapat melakukan update verifikasi peringkat nilai kinerja ini adalah kepala sekolah. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario manajemen verifikasi peringkat nilai kinerja (update) dapat dilihat pada tabel A15

Tabel A 15 Menejaman Verifikasi Peringkat Nilai Kinerja (Update)

|               |   |
|---------------|---|
| Nomer Usecase | USC-13  |
| Nama          | Usecase Menejaman Verifikasi Peringkat Nilai Kinerja (Update) |



|                |   |
|----------------|---|
| Aktor          | Kepala Sekolah  |
| Pre Condition  | Kepala Sekolah akan melakukan update ke verifikasi peringkat nilai kinerja  |
| Post Condition | Kepala Sekolah telah melakukan update ke verifikasi peringkat nilai kinerja |

| <b>SKENARIO DATA NORMAL</b><br><b>“Manajemen Verifikasi Peringkat Nilai Kinerja (View)”</b> |   |
|---|---|
| 1. Klik menu “Nilai Kinerja”  |   |
|   | 2. Menampilkan halaman Nilai Kinerja Guru                   |
| 3. Klik tombol “Peringkat Nilai Kinerja Guru”   |   |
|   | 4. Menampilkan halaman peringkat nilai kinerja guru         |
| 5. Klik tombol “Verifikasi”   |   |
|   | 6. Menampilkan pesan “Berhasil, Data Berhasil Diverifikasi” |

#### A12. Usecase Melihat Peringkat Nilai Kinerja Terverifikasi

##### a. Usecase Melihat Peringkat Nilai Kinerja Terverifikasi (view)

Usecase skenario ini menjelaskan alur untuk proses melihat peringkat nilai kinerja semua guru yang telah terverifikasi oleh kepala sekolah. Aktor yang dapat melihat peringkat nilai kinerja terverifikasi ini adalah guru. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario melihat peringkat nilai kinerja terverifikasi (view) dapat dilihat pada tabel A16

Tabel A 16 Usecase Skenario Melihat Peringkat Nilai Kinerja Terverifikasi (View)

|               |   |
|---------------|---|
| Nomer Usecase | USC-14  |
| Nama          | Usecase Melihat Peringkat Nilai Kinerja Terverifikasi (View)      |
| Aktor         | Guru  |
| Pre Condition | Guru akan melakukan view ke peringkat nilai kinerja terverifikasi |

|                |  |
|----------------|--|
| Post Condition | Guru telah melakukan view ke peringkat nilai kinerja terverifikasi |
|----------------|--|

| <b>SKENARIO DATA NORMAL</b><br><b>“Melihat Peringkat Nilai Kinerja Terverifikasi (View)”</b> |   |
|--|---|
| 1. Klik menu “Nilai Kinerja”   |   |
|  | 2. Menampilkan halaman Nilai Kinerja Guru           |
| 3. Klik tombol “Peringkat Nilai Kinerja Guru”  |   |
|  | 4. Menampilkan halaman peringkat nilai kinerja guru |

#### A13. Usecase Manajemen Update Rekap Hasil Peringkat

##### a. Usecase Manajemen Update Rekap Hasil Peringkat (View)

Usecase skenario ini menjelaskan alur untuk proses melihat hasil peringkat yang nanti akan di update menjadi menjadi hasil rekap. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario manajemen update rekap hasil peringkat (view) dapat dilihat pada tabel A17

Tabel A 17 Usecase Skenario Melihat Update Rekap Hasil Peringkat (View)

|                |   |
|----------------|---|
| Nomer Usecase  | USC-15  |
| Nama           | Usecase Manajemen Update Rekap Hasil Peringkat (View) |
| Aktor          | Admin   |
| Pre Condition  | Admin akan melihat ke Update Rekap Hasil Peringkat    |
| Post Condition | Admin telah melihat Update Rekap Hasil Peringkat      |

| <b>SKENARIO DATA NORMAL</b><br><b>“Manajemen Manajemen Update Rekap Hasil Peringkat (View)”</b> |  |
|---|--|
| 1. Klik menu “Hasil Peringkat”  |  |
|   | 2. Menampilkan halaman Hasil Penentuan Peringkat |

##### b. Usecase Manajemen Update Rekap Hasil Peringkat (update)

Usecase skenario ini menjelaskan alur untuk proses melakukan update hasil penentuan peringkat sehingga tersimpan ke database menu rekap nilai tiap tahun ajaran. Aktor yang dapat melakukan update rekap hasil peringkat ini adalah admin. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario manajemen update rekap hasil peringkat (update) dapat dilihat pada tabel A18

Tabel A 18 Usecase Skenario Manajemen Update Rekap Hasil Peringkat (Update)

|                |  |
|----------------|--|
| Nomer Usecase  | USC-15   |
| Nama           | Usecase Manajemen Update Rekap Hasil Peringkat (Update)      |
| Aktor          | Admin  |
| Pre Condition  | Admin akan melakukan update ke Update Rekap Hasil Peringkat  |
| Post Condition | Admin telah melakukan update ke Update Rekap Hasil Peringkat |

| <b>SKENARIO DATA NORMAL</b><br><b>“Manajemen Update Rekap Hasil Peringkat (Update)”</b> |   |
|---|---|
| 1. Klik menu “Hasil Peringkat”  |   |
|   | 2. Menampilkan halaman Hasil Penentuan Peringkat            |
| 3. Klik tombol “Selesai”  |   |
|   | 4. Menyimpan data dalam database                            |
|   | 5. Menampilkan pesan “Berhasil, Rekap Data Telah Dilakukan” |
| \   | 6. Menampilkan halaman Data Rekap                           |

A14. Usecase Melihat Rekap Penilaian  
Usecase Melihat Rekap Penilaian (View)

Usecase skenario ini menjelaskan alur untuk proses melihat rekap peringkat penilaian kinerja guru pada tahun ajaran sebelumnya. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario melihat rekap penilaian (view) dapat dilihat pada tabel A19

Tabel A 19 Usecase Skenario Melihat Rekap Penilaian (View)

|                |   |
|----------------|---|
| Nomer Usecase  | USC-15  |
| Nama           | Usecase Melihat Rekap Penilaian (View)        |
| Aktor          | Admin   |
| Pre Condition  | Admin akan melihat ke Rekap Penilaian (View)  |
| Post Condition | Admin telah melihat ke Rekap Penilaian (View) |

| <b>SKENARIO DATA NORMAL<br/>“Melihat Rekap Penilaian (View)”</b> |   |
|--|---|
| 1. Klik menu “Rekap”   |   |
|  | 2. Menampilkan halaman Data Rekap         |
| 3. Klik salah satu baris tahun ajaran dikolom tabel              |   |
|  | 4. Menampilkan tabel peringkat nilai guru |
| 5. Klik salah satu baris nama di kolom tabel                     |   |
|  | 6. Menampilkan halaman data rekap user    |

#### A15. Logout

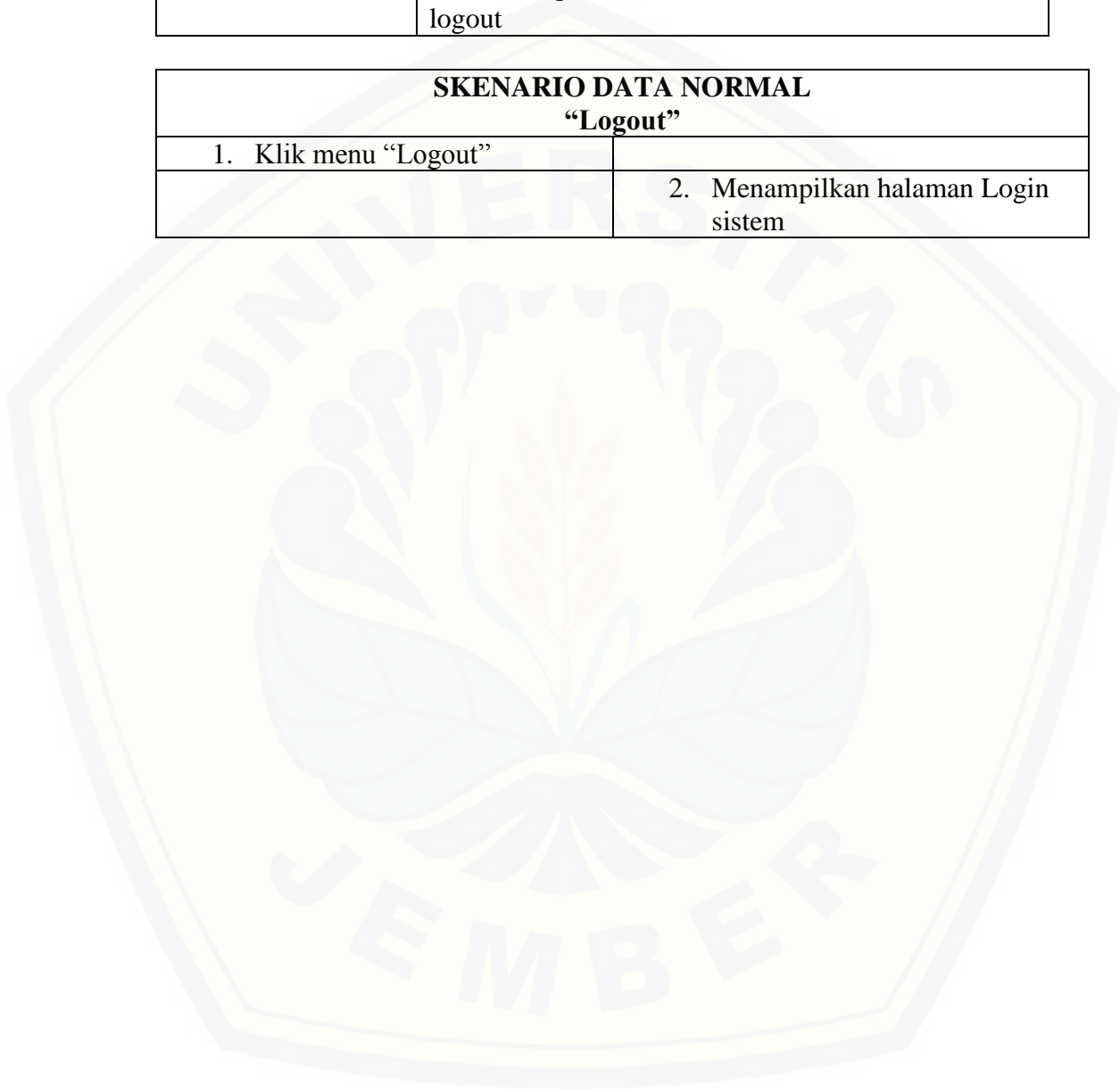
Usecase logout ini menjelaskan proses untuk keluar dari sistem. Aktor yang dapat melakukan proses logout adalah guru, kepala sekolah, dan admin. Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario logout dapat dilihat pada tabel A20

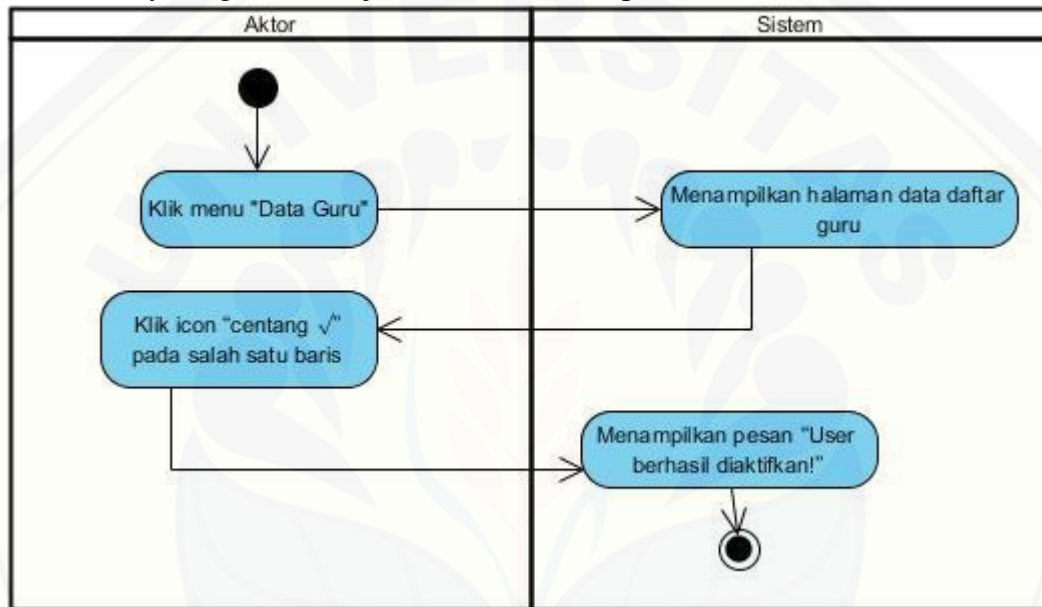
Tabel A 20 Usecase Skenario Logout

|               |                |
|---------------|----------------|
| Nomer Usecase | USC-16         |
| Nama          | Usecase Logout |

|                |  |
|----------------|--|
| Aktor          | Admin, Kepala Sekolah, dan Guru                        |
| Pre Condition  | Admin, Kepala Sekolah, dan Guru akan melakukan logout  |
| Post Condition | Admin, Kepala Sekolah, dan Guru telah melakukan logout |

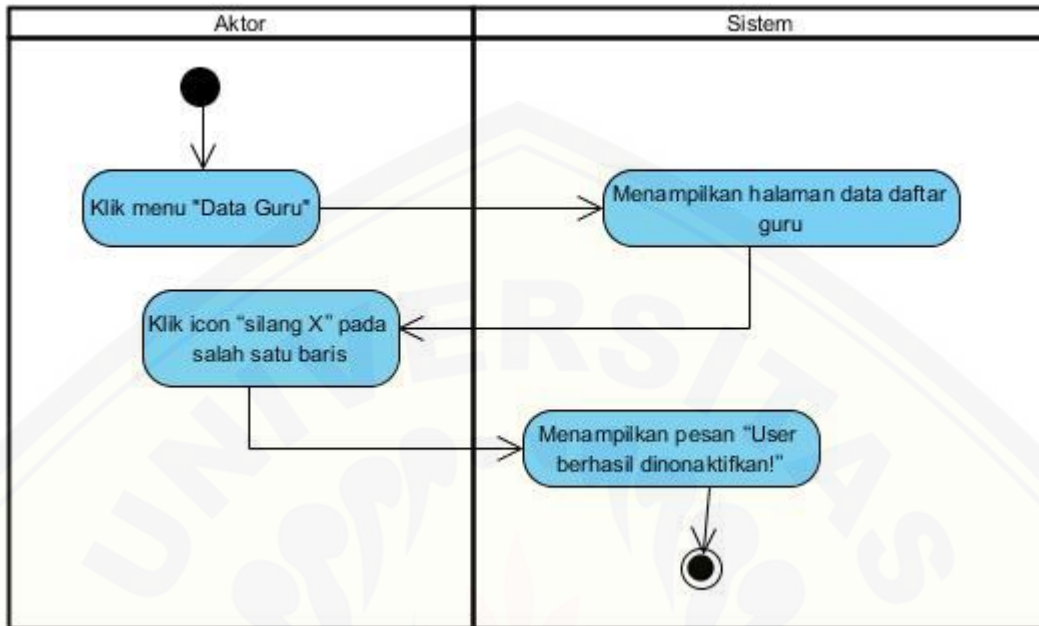
|  |                                     |
|--|-------------------------------------|
| <b>SKENARIO DATA NORMAL</b><br><b>“Logout”</b> |                                     |
| 1. Klik menu “Logout”                          |                                     |
|  | 2. Menampilkan halaman Login sistem |



**Lampiran B. Activity Diagram****B1. Activity Diagram Manajemen Data User (Update) Status Aktif User**

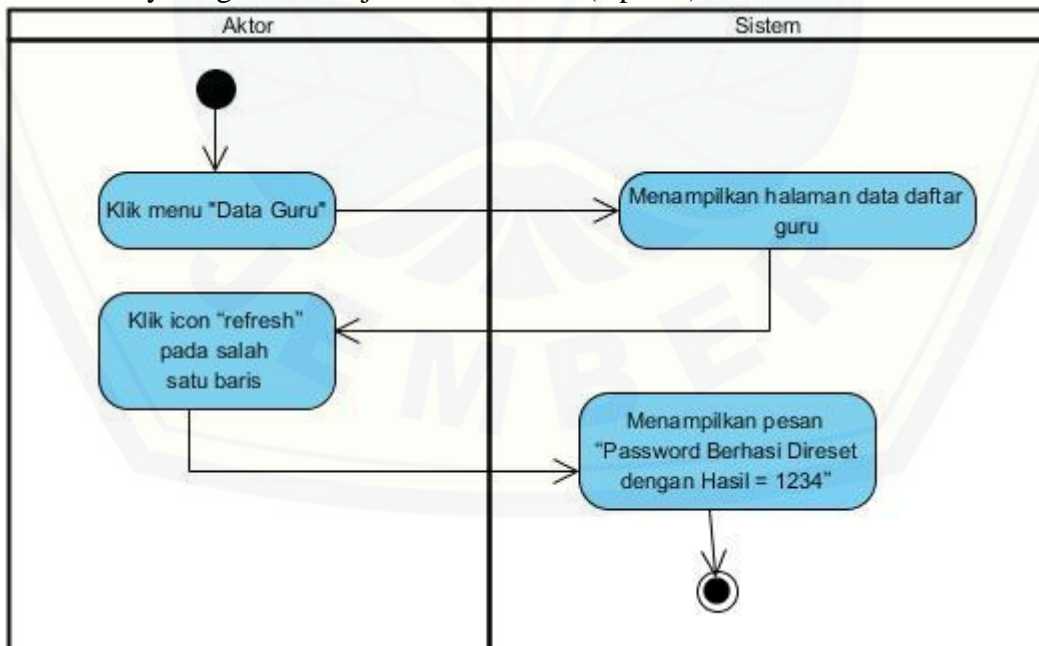
Gambar B 1 Activity Diagram Manajemen Data User (Update) Status Aktif User

## B2. Activity Diagram Manajemen Data User (Update) Status Nonaktif User



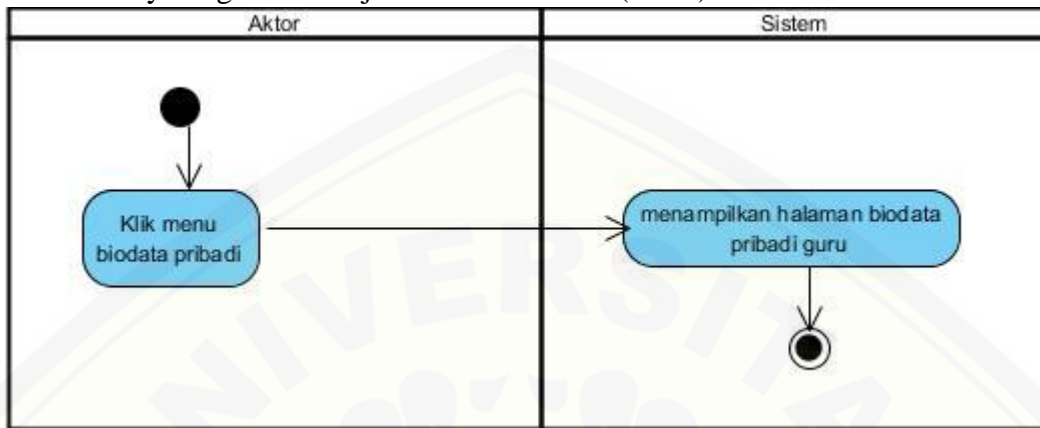
Gambar B.2 Activity Diagram Manajemen Data User (Update) Status Nonaktif User

## B3. Activity Diagram Manajemen Data User (Update) Reset Password User



Gambar B 3 Activity Diagram Manajemen Data User (Update) Reset Password User

## B4. Activity Diagram Manajemen Data Pribadi (View)

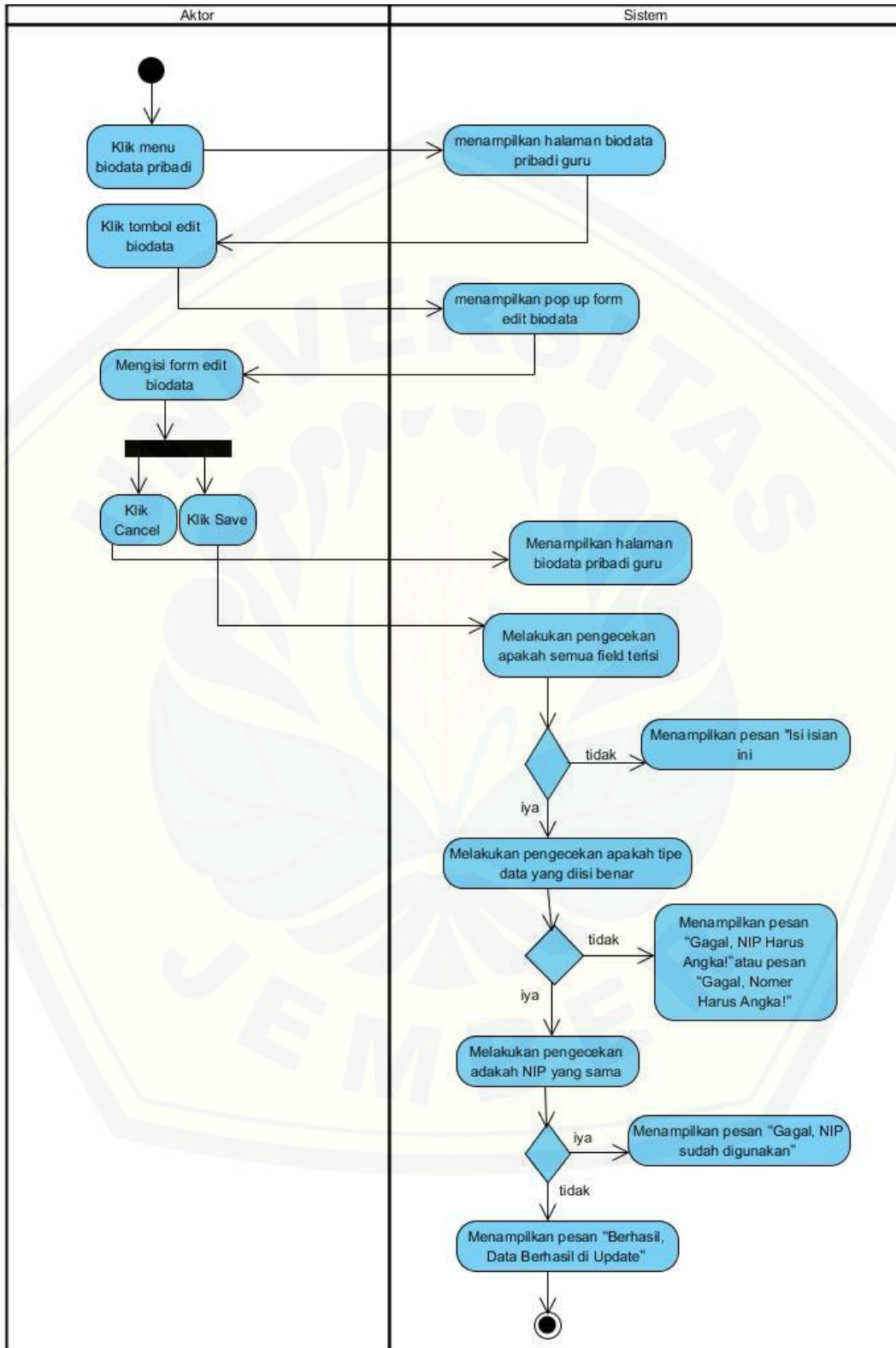


Gambar B 4 Activity Diagram Manajemen Data Pribadi (View)



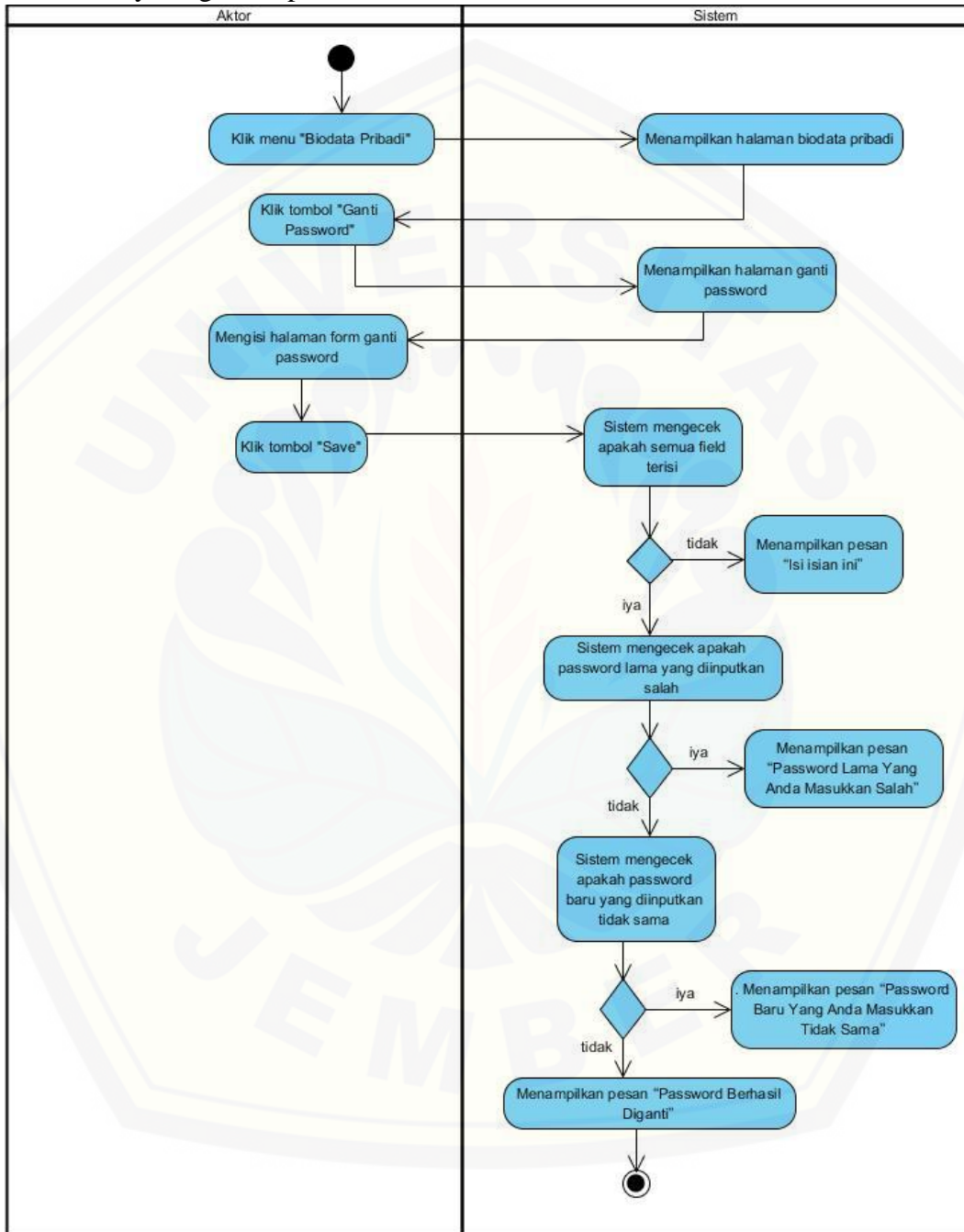
B5. Activity Diagram Manajemen Data Pribadi (Edit)





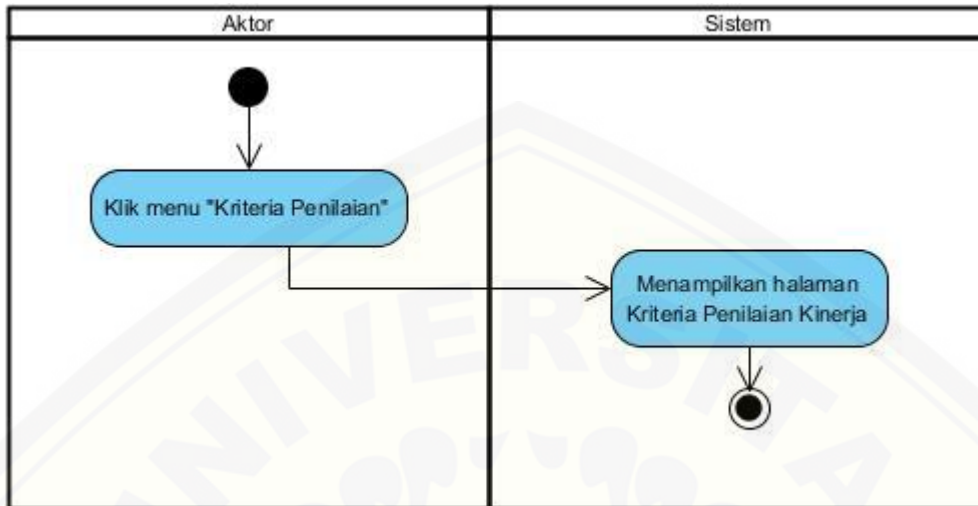
Gambar B 5 Activity Diagram Manajemen Data Pribadi (Edit)

B6. Activity Diagram Update Password Pribadi



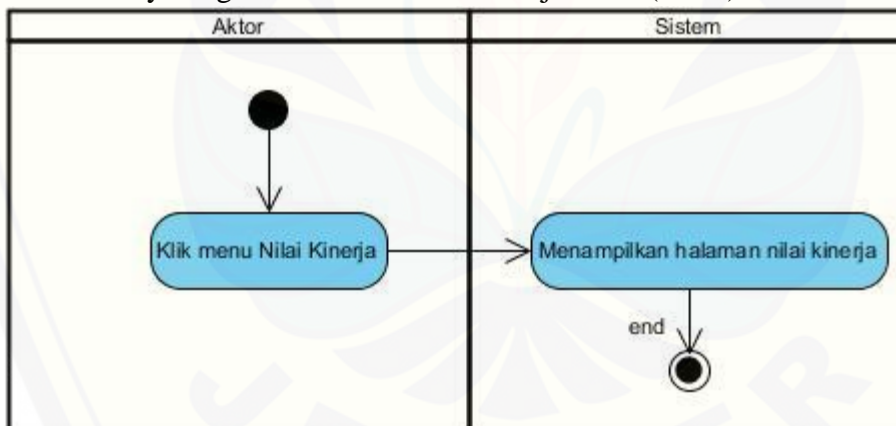
Gambar B 6 Activity Diagram Update Password Pribadi

## B7. Activity Diagram Melihat Kriteria Penilaian Kinerja Guru (View)



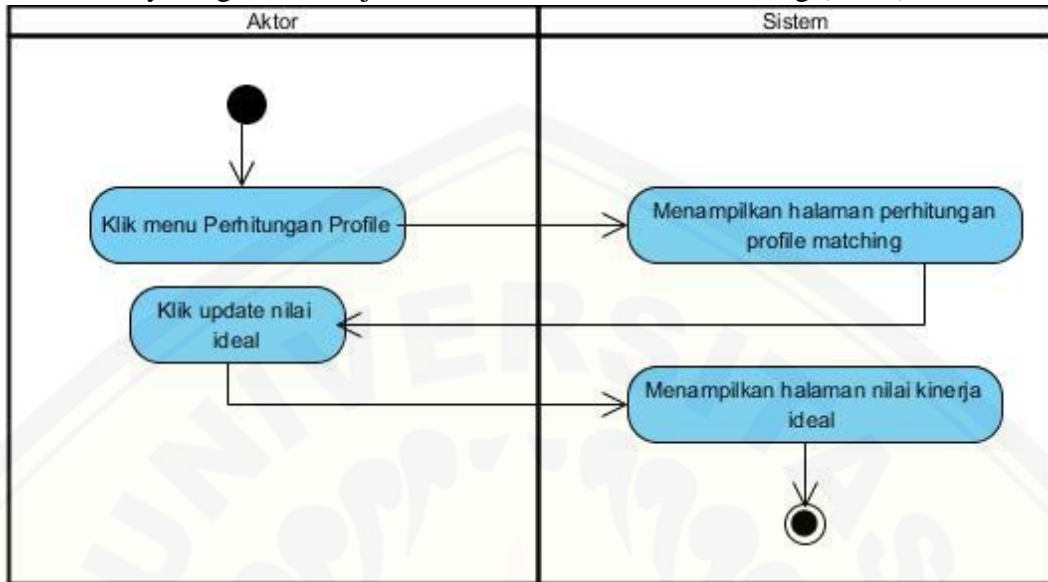
Gambar B 7 Activity Diagram Melihat Kriteria Penilaian Kinerja Guru (View)

## B8. Activity Diagram Melihat Nilai Kinerja Guru (View)



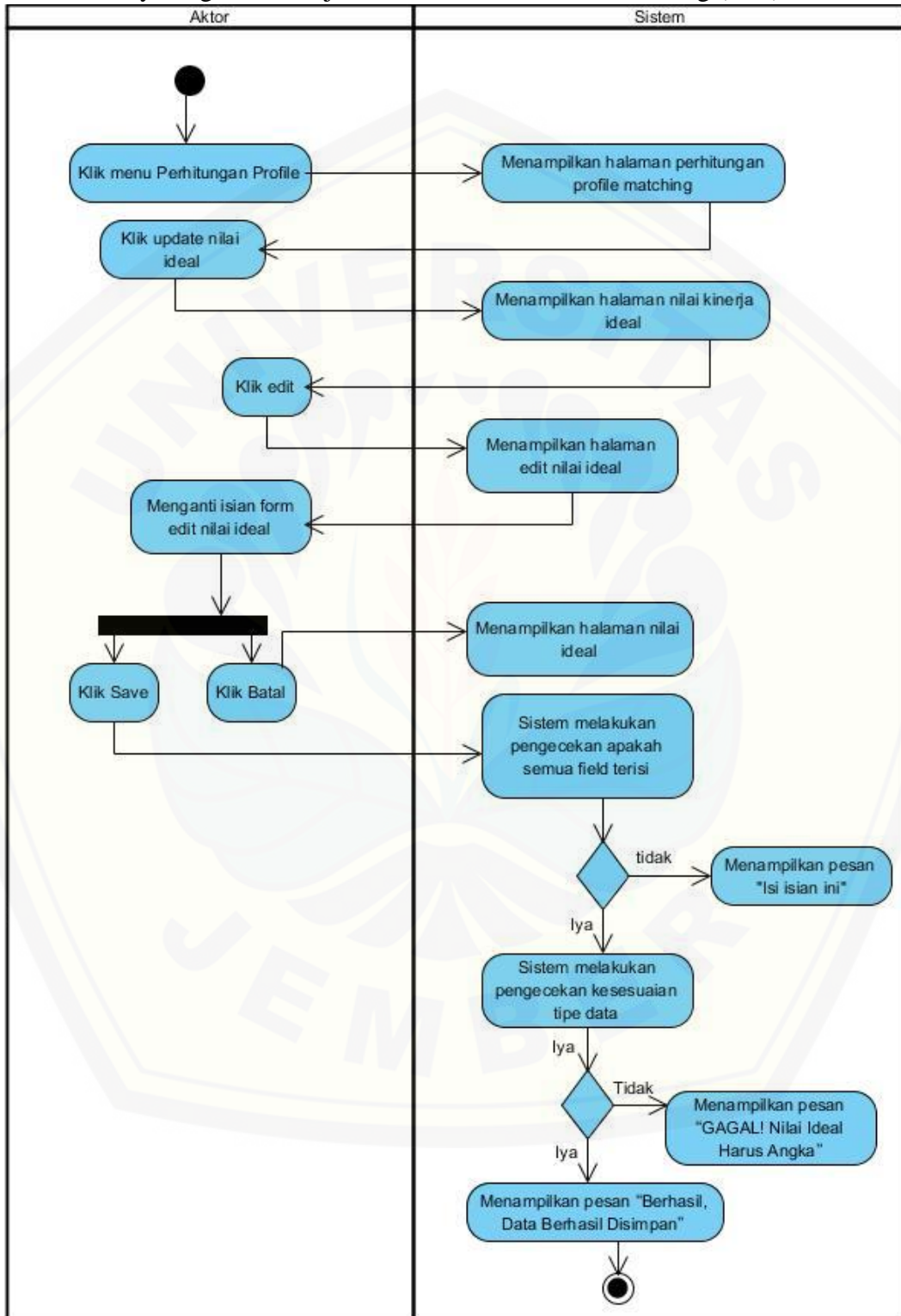
Gambar B 8 Activity Diagram Melihat Nilai Kinerja Guru (View)

B9. Activity Diagram Manajemen Nilai Ideal Profile Matching (View)



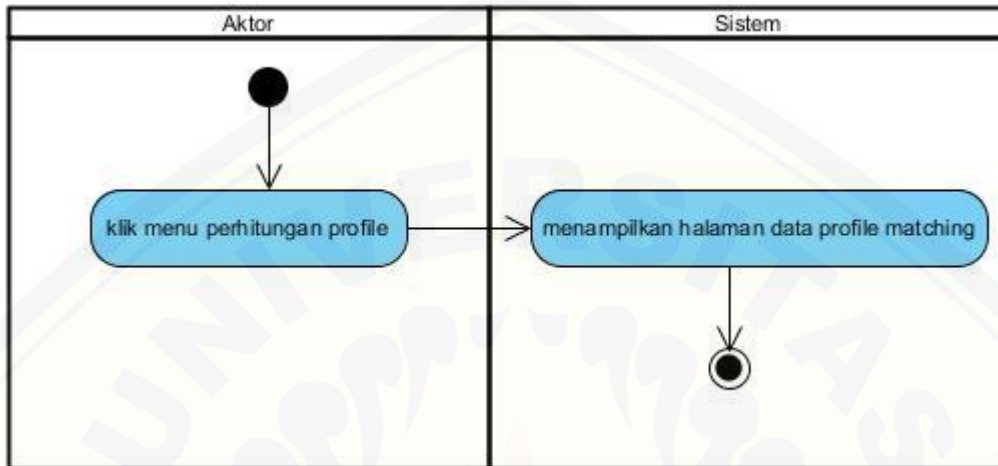
Gambar B 9 Activity Diagram Manajemen Nilai Ideal Profile Matching (View)

B10. Activity Diagram Manajemen Nilai Ideal Profile Matching (Edit)



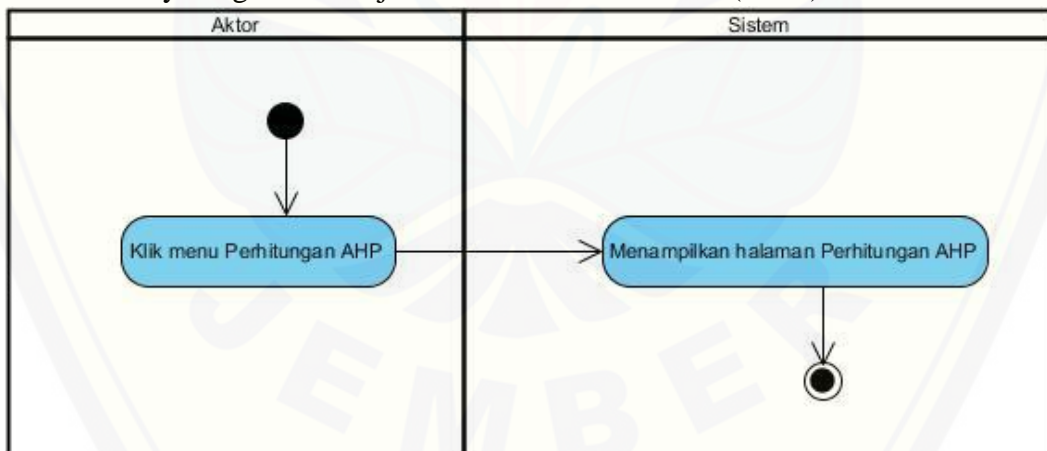
Gambar B 10 Activity Diagram Manajemen Nilai Ideal Profile Matching (Edit)

B11. Activity Diagram Melihat Nilai Kinerja Profile Matching (View)



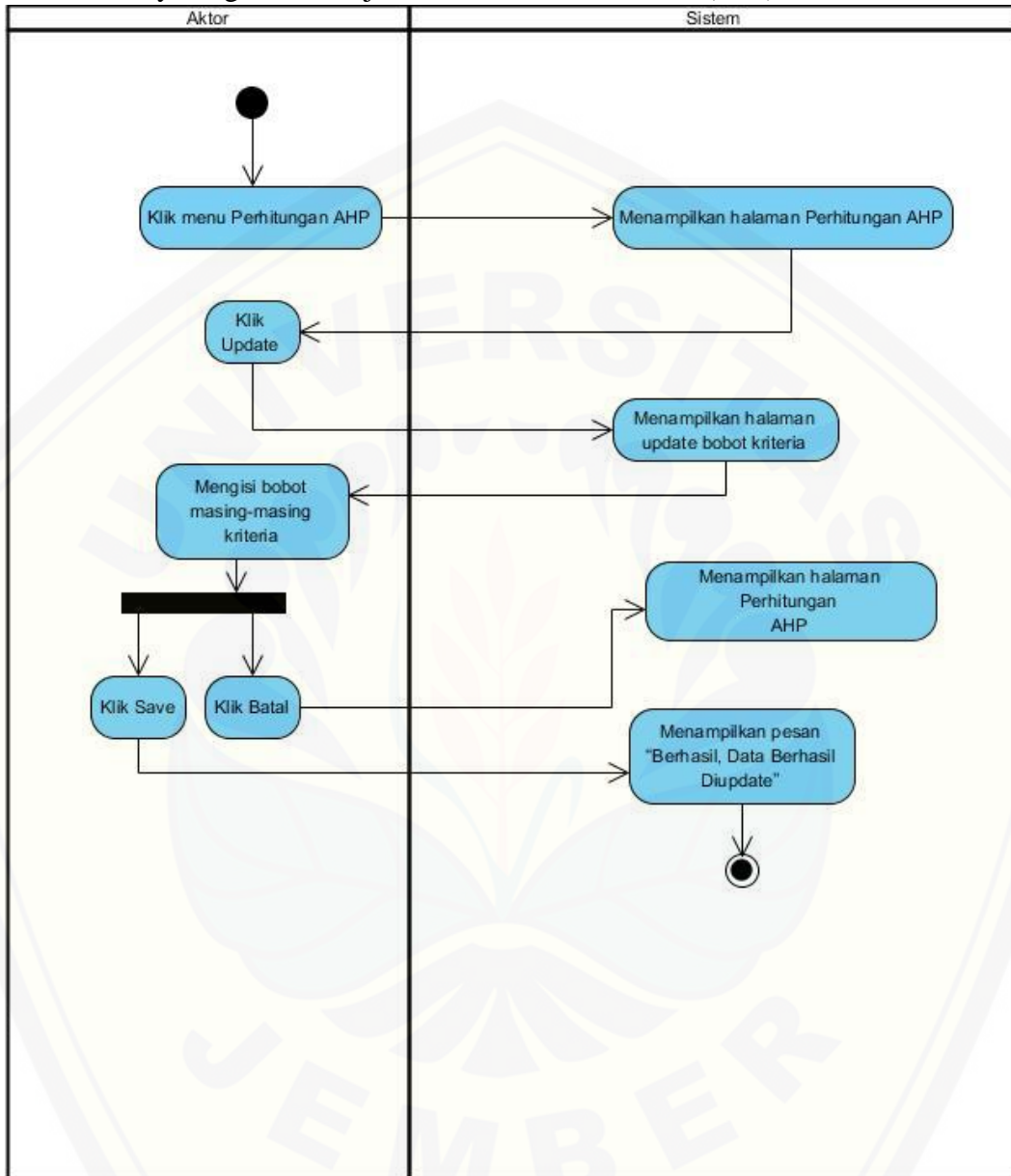
Gambar B 11 Activity Diagram Manajemen Nilai Ideal Profile Matching (View)

B12. Activity Diagram Manajemen Bobot Kriteria AHP (View)



Gambar B 12 Activity Diagram Manajemen Bobot Kriteria AHP (View)

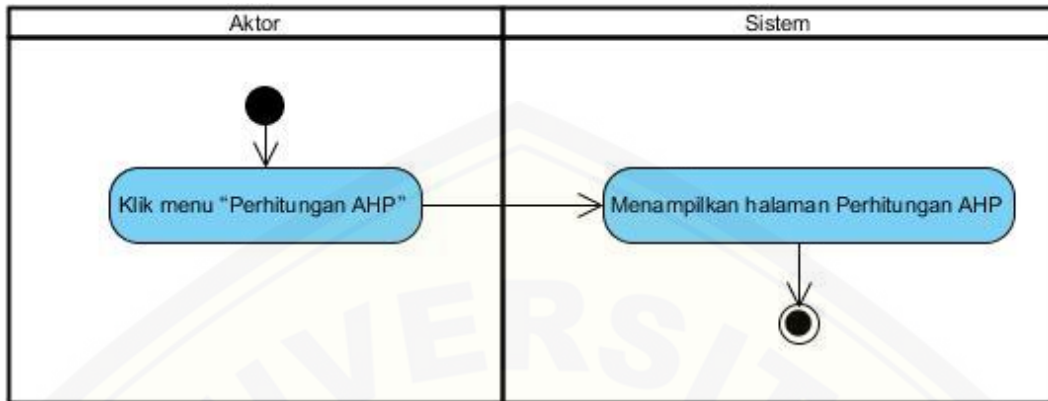
B13. Activity Diagram Manajemen Bobot Kriteria AHP (Edit)



Gambar B 13 Activity Diagram Manajemen Bobot Kriteria AHP (Edit)

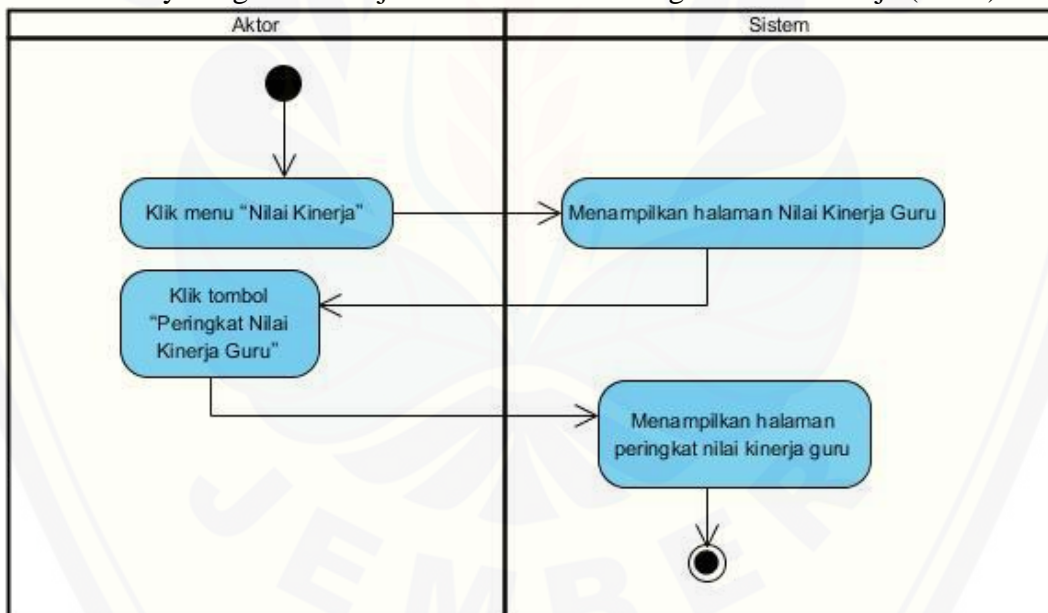


B14. Activity Diagram Manajemen Bobot Kriteria AHP (View)



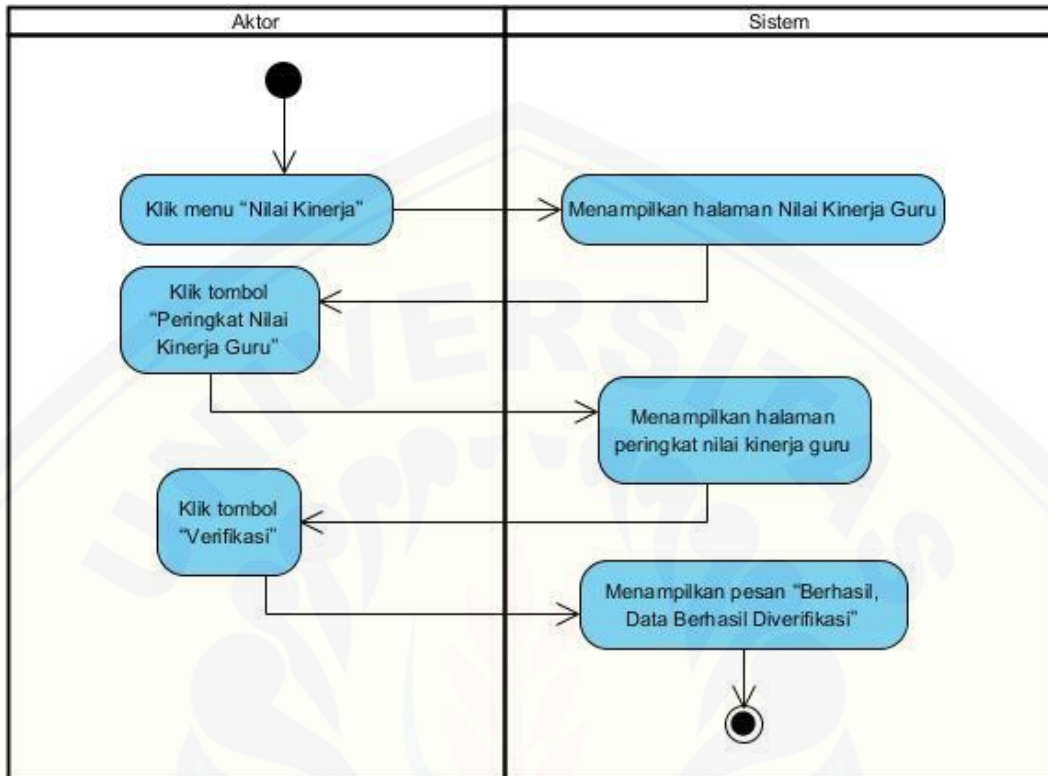
Gambar B 14 Activity Diagram Manajemen Bobot Kriteria AHP (View)

B15. Activity Diagram Manajemen Verifikasi Peringkat Nilai Kinerja (View)



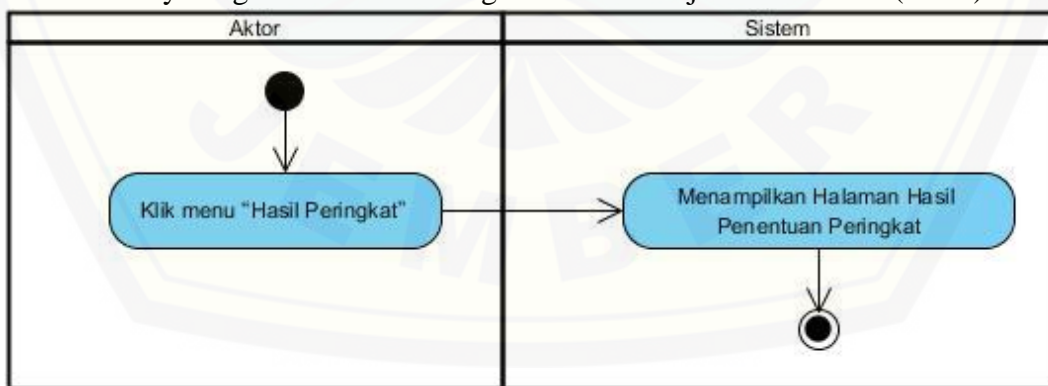
Gambar B 15 Activity Diagram Manajemen Verifikasi Peringkat Nilai Kinerja (View)

B16. Activity Diagram Manajemen Verifikasi Peringkat Nilai Kinerja (Update)



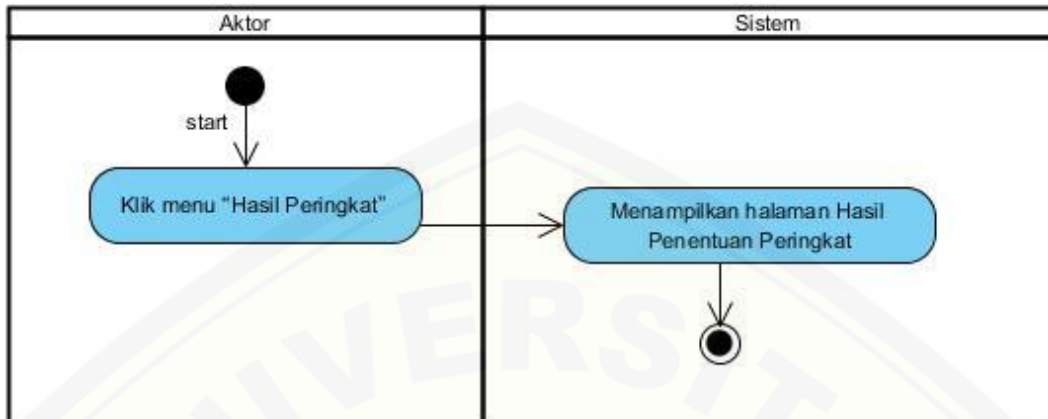
Gambar B 16 Activity Diagram Manajemen Verifikasi Peringkat Nilai Kinerja (Update)

B17. Activity Diagram Melihat Peringkat Nilai Kinerja Terverifikasi (View)



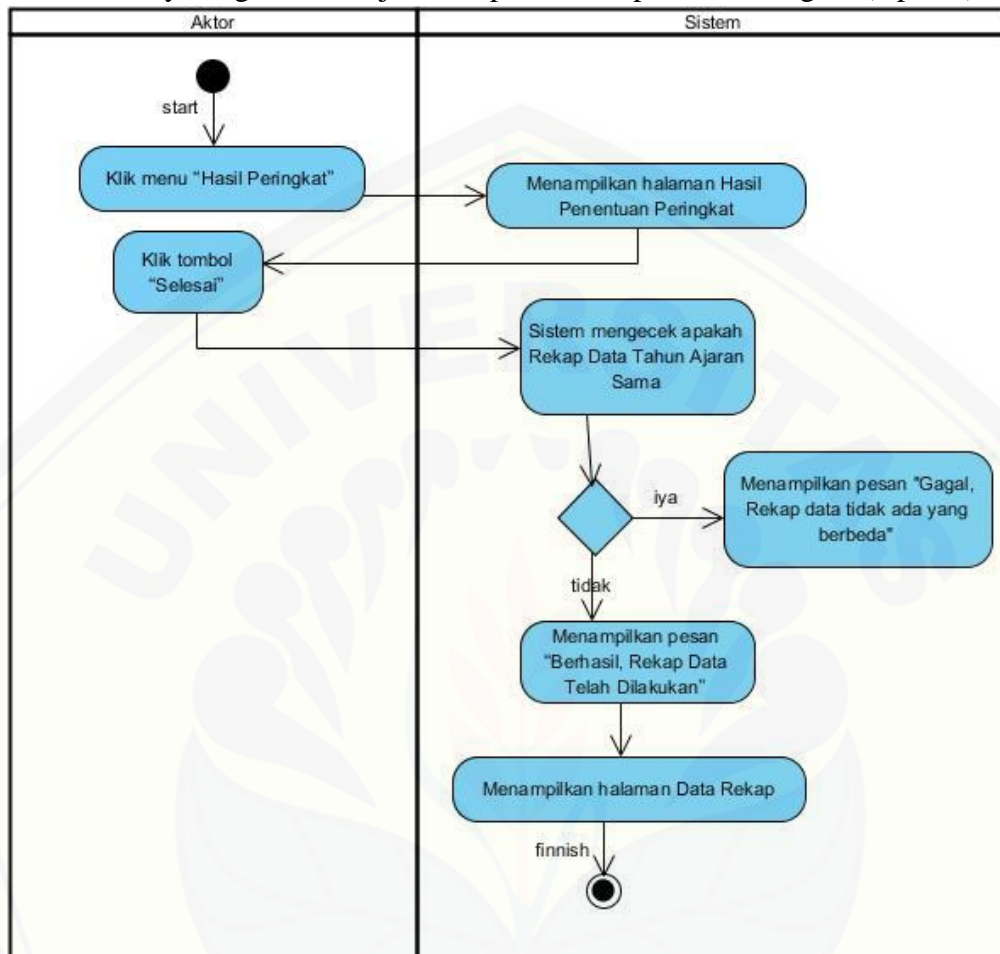
Gambar B 17 Activity Diagram Melihat Peringkat Nilai Kinerja Terverifikasi (View)

B18. Activity Diagram Manajemen Update Rekap Hasil Peringkat (View)



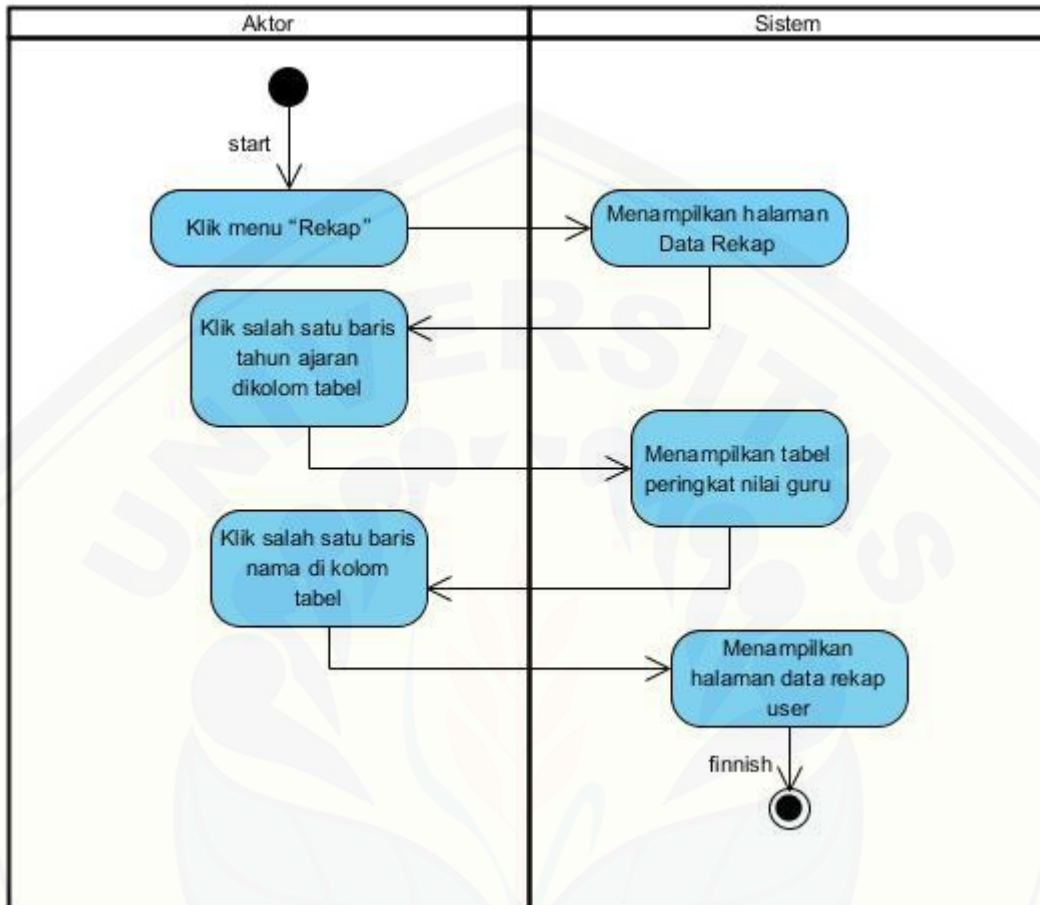
Gambar B 18 Activity Diagram Manajemen Update Rekap Hasil Peringkat (View)

B19. Activity Diagram Manajemen Update Rekap Hasil Peringkat (Update)



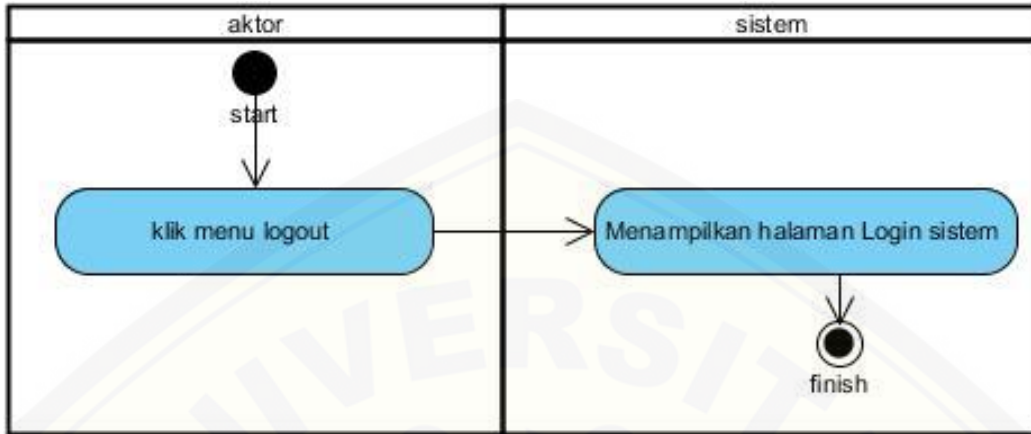
Gambar B 19 . Activity Diagram Manajemen Update Rekap Hasil Peringkat (Update)

## B20. Activity Diagram Melihat Rekap Penilaian (View)



Gambar B 20 Activity Diagram Melihat Rekap Penilaian (View)

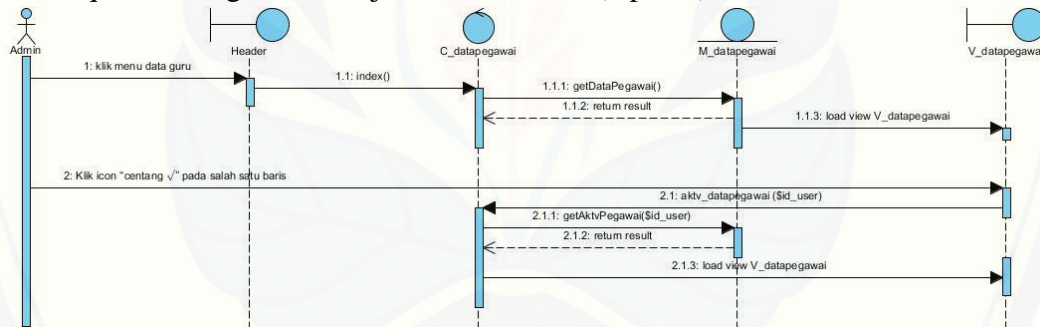
B21. Activity Diagram Logout



Gambar B 21 Activity Diagram Logout

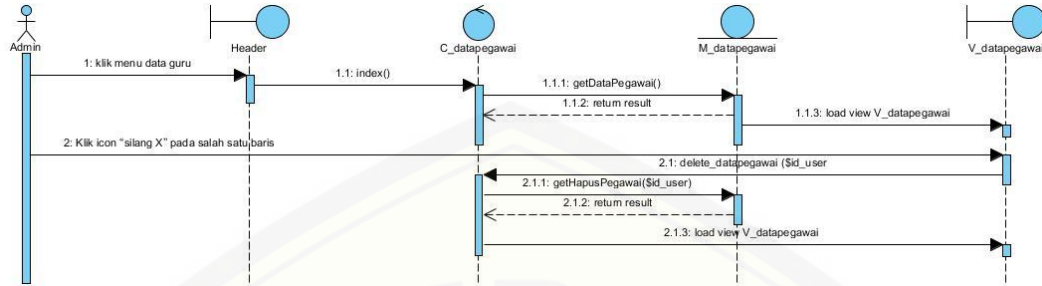
Lampiran C. Sequence Diagram

C1. Sequence Diagram Manajemen Data User (Update) Status Aktif User



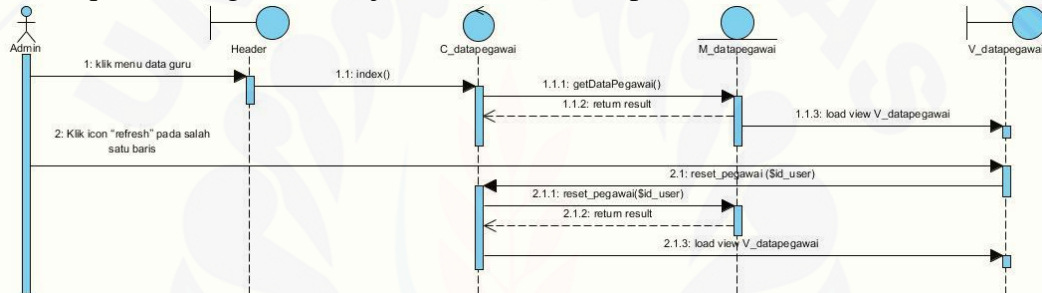
Gambar C 1 Sequence Diagram Manajemen Data User (Update) Status Aktif User

C2. Sequence Diagram Manajemen Data User (Update) Nonaktif Aktif User



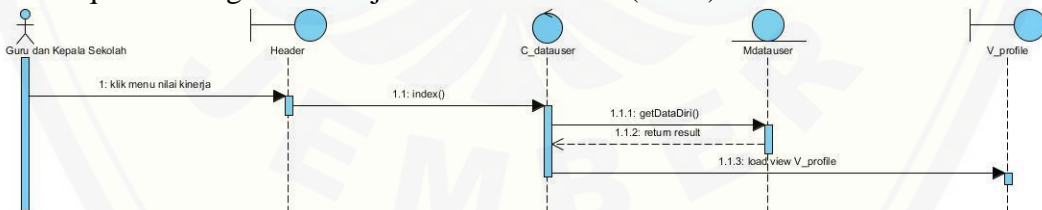
Gambar C 2 Sequence Diagram Manajemen Data User (Update) Nonaktif Aktif User

C3. Sequence Diagram Manajemen Data User (Update) Reset Password User



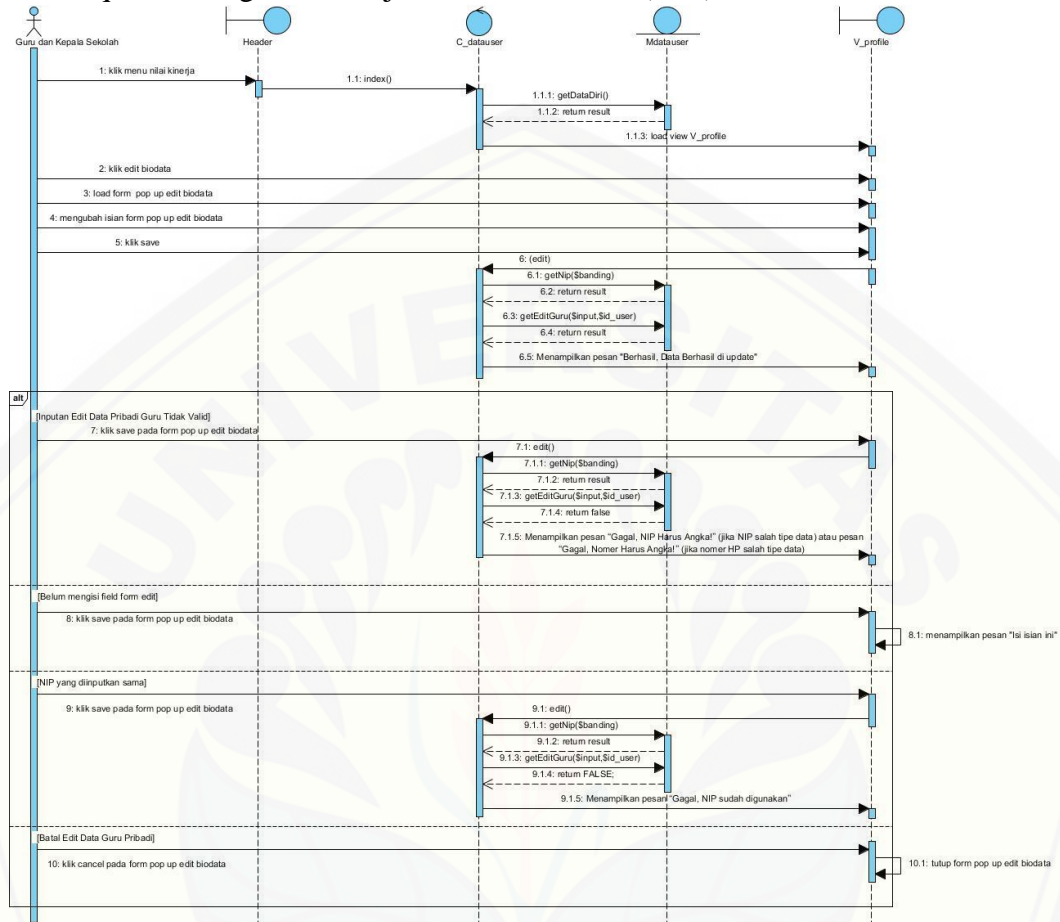
Gambar C 3 Sequence Diagram Manajemen Data User (Update) Reset Password User

C4. Sequence Diagram Manajemen Data Pribadi (View)



Gambar C 4 Sequence Diagram Manajemen Data Pribadi (View)

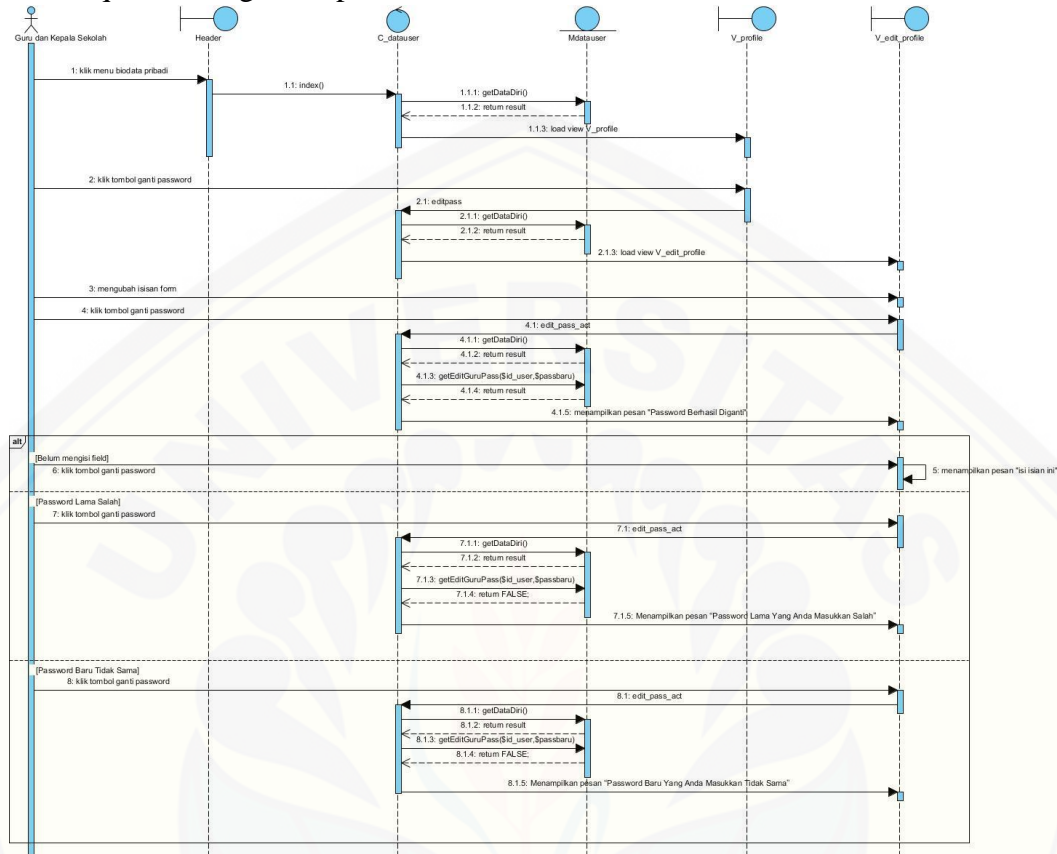
C5. Sequence Diagram Manajemen Data Pribadi (Edit)



Gambar C 5 Sequence Diagram Manajemen Data Pribadi (Edit)

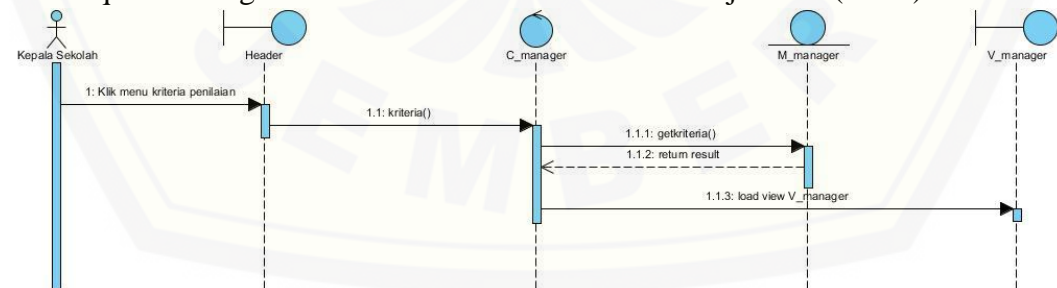


C6. Sequence Diagram Update Password Pribadi



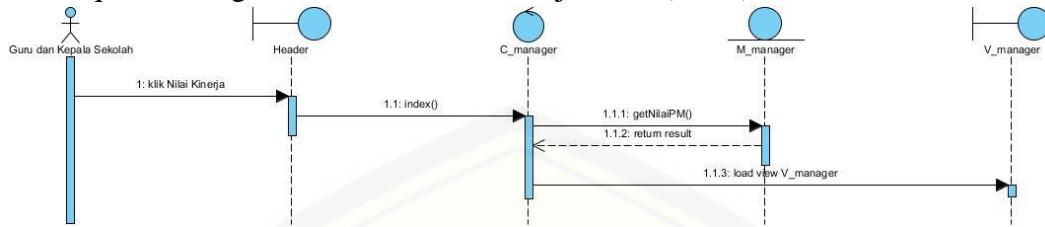
Gambar C 6 Sequence Diagram Update Password Pribadi

C7. Sequence Diagram Melihat Kriteria Penilaian Kinerja Guru (View)



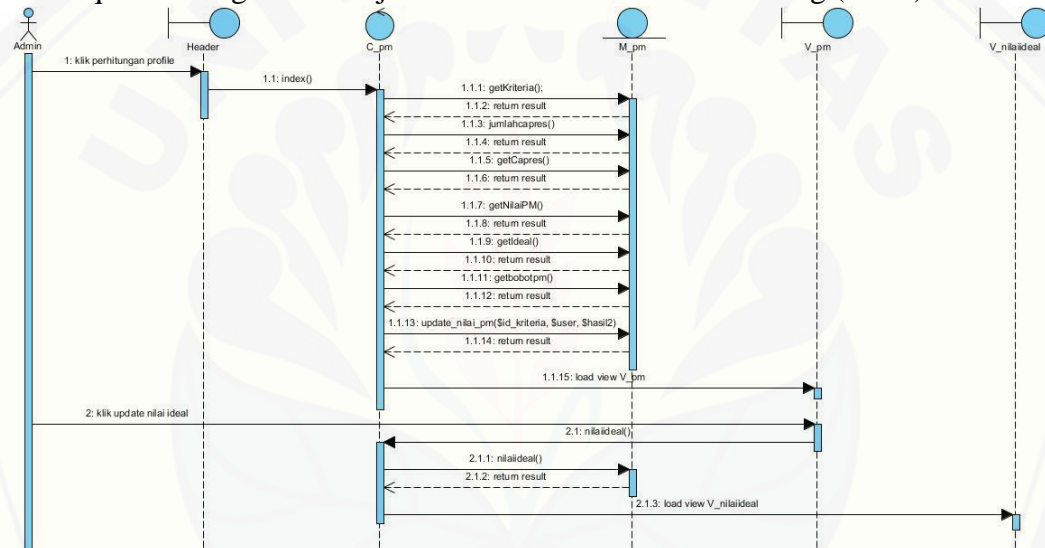
Gambar C 7 Sequence Diagram Melihat Kriteria Penilaian Kinerja Guru (View)

C8. Sequence Diagram Melihat Nilai Kinerja Guru (View)



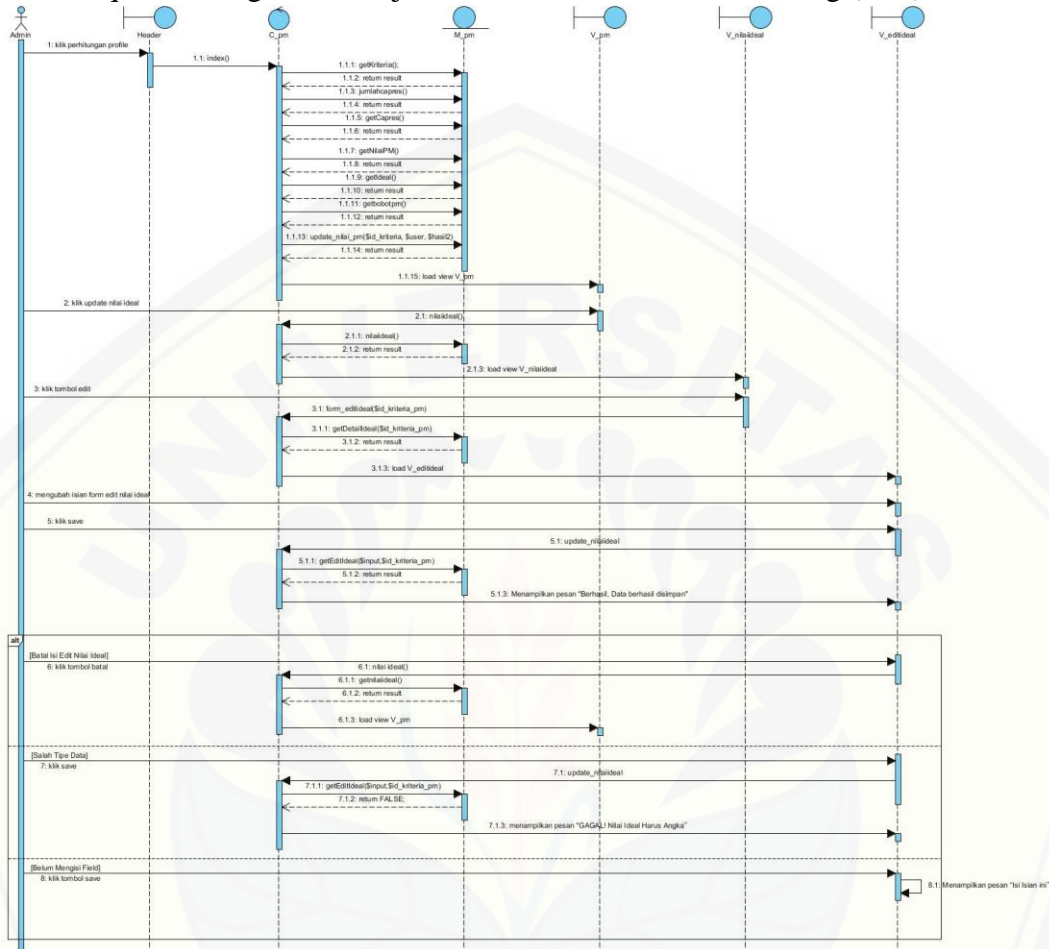
Gambar C 8 Sequence Diagram Melihat Nilai Kinerja Guru (View)

C9. Sequence Diagram Manajemen Nilai Ideal Profile Matching (View)



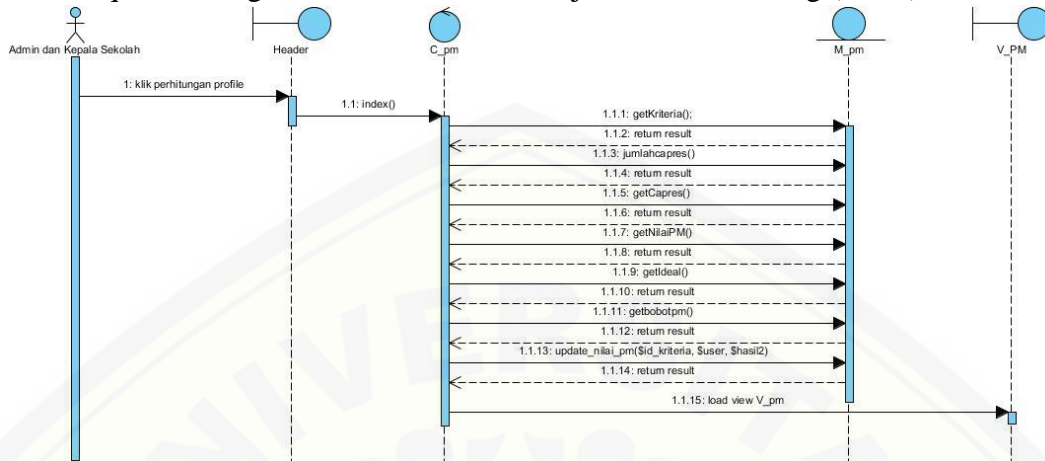
Gambar C 9 Sequence Diagram Manajemen Nilai Ideal Profile Matching (View)

C10. Sequence Diagram Manajemen Nilai Ideal Profile Matching (Edit)



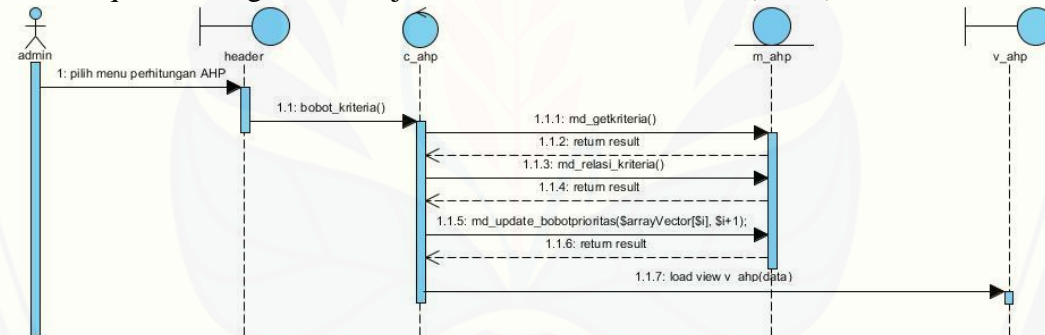
Gambar C 10 Sequence Diagram Manajemen Nilai Ideal Profile Matching (Edit)

C11. Sequence Diagram Melihat Nilai Kinerja Profile Matching (View)



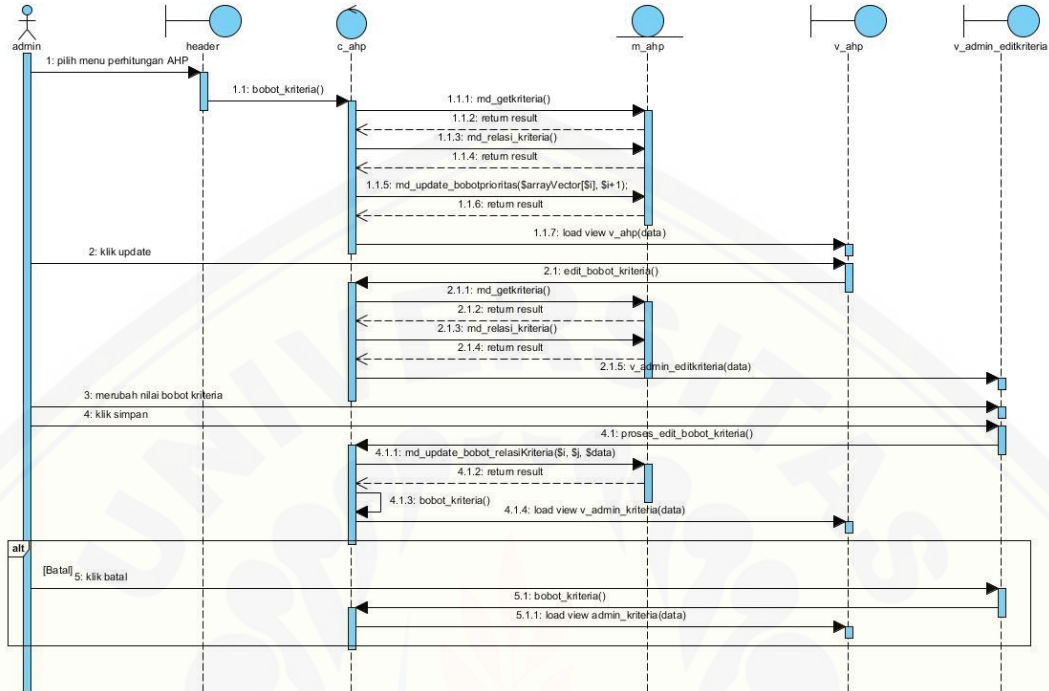
Gambar C 11 Sequence Diagram Melihat Nilai Kinerja Profile Matching (View)

C12. Sequence Diagram Manajemen Bobot Kriteria AHP (View)



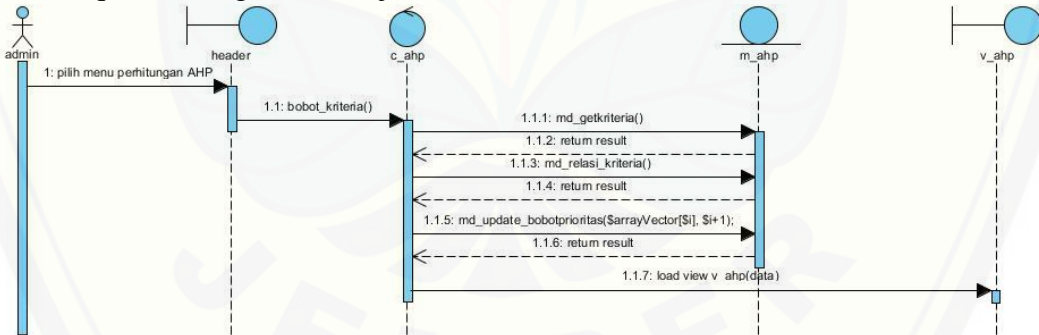
Gambar C 12 Sequence Diagram Manajemen Bobot Kriteria AHP (View)

C13. Sequence Diagram Manajemen Bobot Kriteria AHP (Edit)



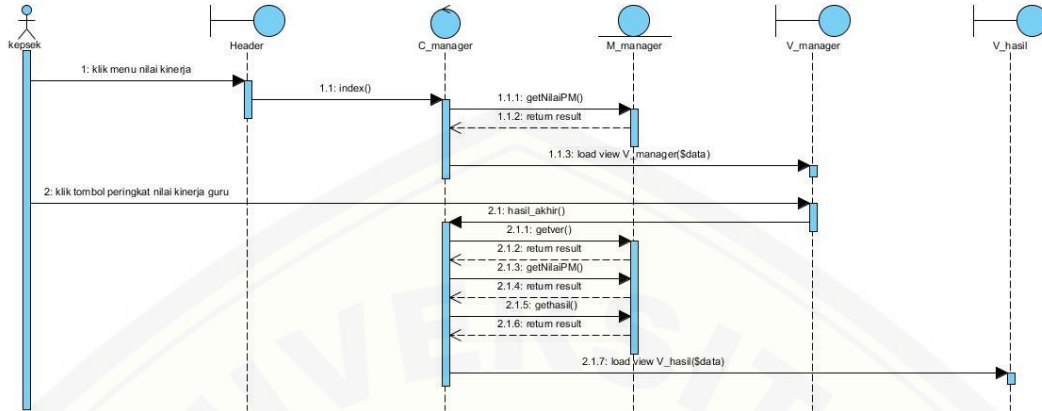
Gambar C 13 Sequence Diagram Manajemen Bobot Kriteria AHP (Edit)

C14. Sequence Diagram Manajemen Bobot Kriteria AHP (View)



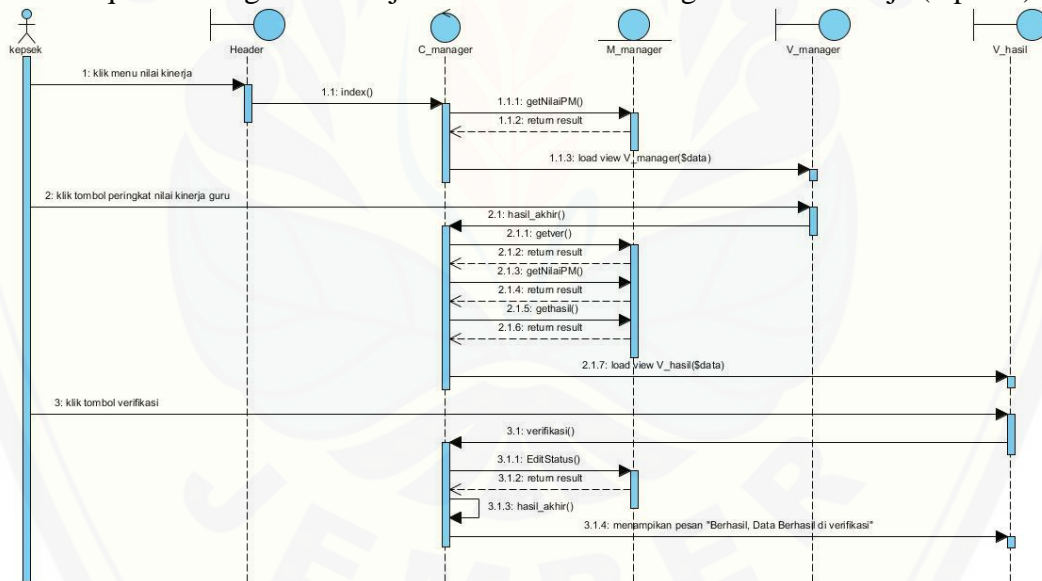
Gambar C 14 Sequence Diagram Manajemen Bobot Kriteria AHP (View)

C15. Sequence Diagram Manajemen Verifikasi Peringkat Nilai Kinerja (View)



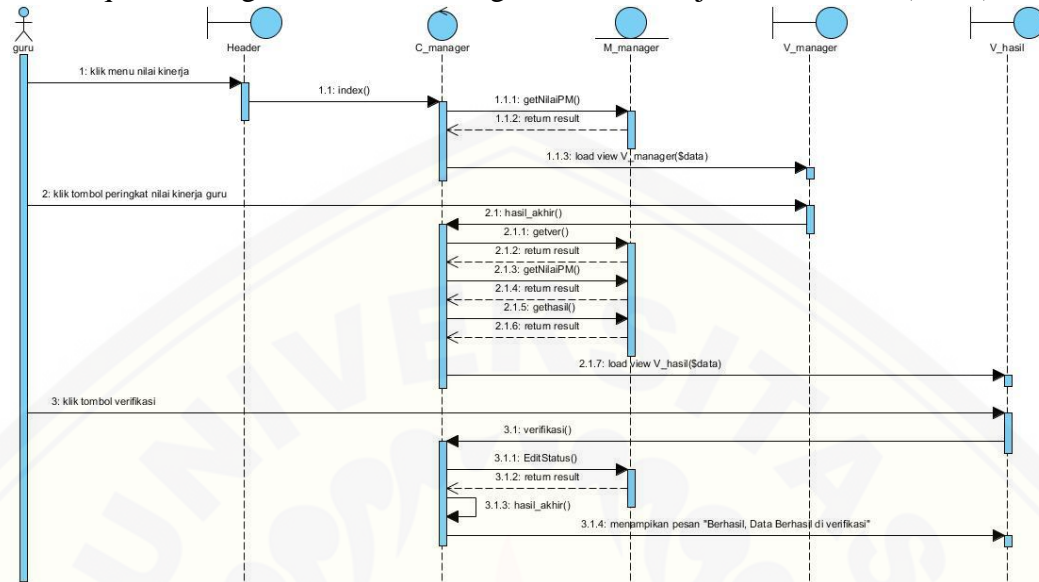
Gambar C 15 Sequence Diagram Manajemen Verifikasi Peringkat Nilai Kinerja (View)

C16. Sequence Diagram Manajemen Verifikasi Peringkat Nilai Kinerja (Update)



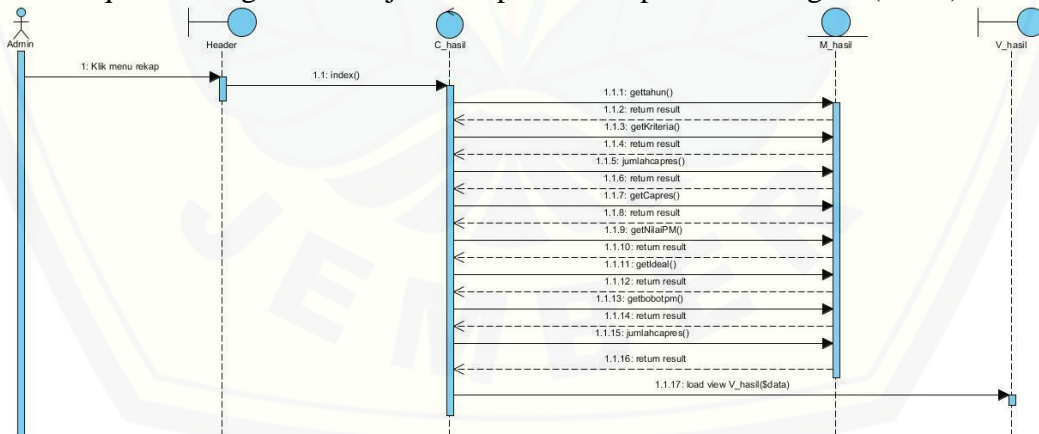
Gambar C 16 Sequence Diagram Manajemen Verifikasi Peringkat Nilai Kinerja (Update)

C17. Sequence Diagram Melihat Peringkat Nilai Kinerja Terverifikasi (View)



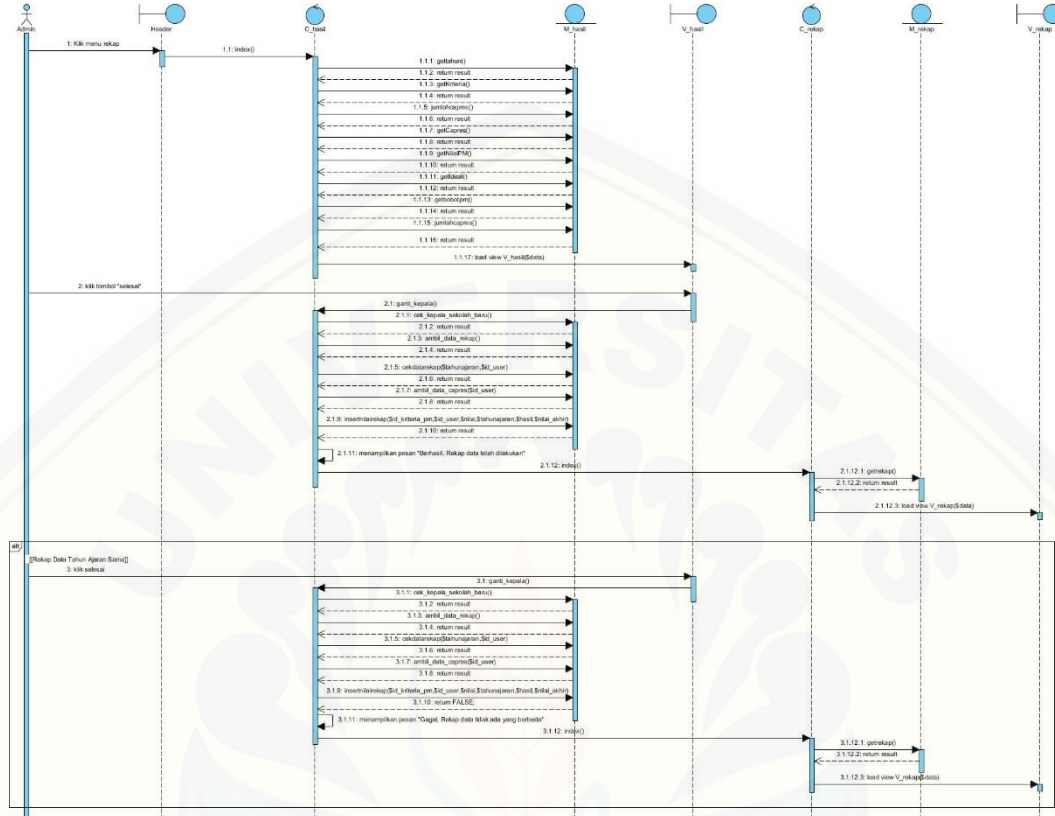
Gambar C 17 Sequence Diagram Melihat Peringkat Nilai Kinerja Terverifikasi (View)

C18. Sequence Diagram Manajemen Update Rekap Hasil Peringkat (View)



Gambar C 18 Sequence Diagram Manajemen Update Rekap Hasil Peringkat (View)

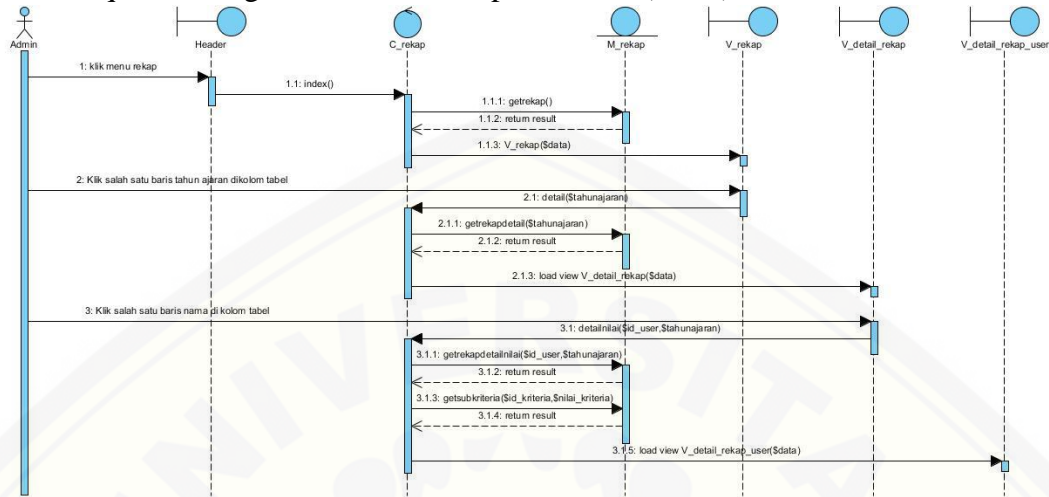
C19. Sequence Diagram Manajemen Update Rekap Hasil Peringkat (Update)



Gambar C 19 Sequence Diagram Manajemen Update Rekap Hasil Peringkat (Update)

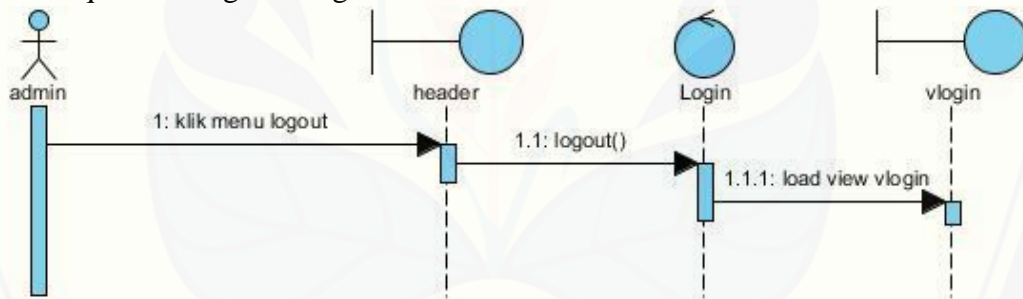


C20. Sequence Diagram Melihat Rekap Penilaian (View)



Gambar C 20 Sequence Diagram Melihat Rekap Penilaian (View)

C21. Sequence Diagram Logout



Gambar C 21 Sequence Diagram Logout

## Lampiran D. Penulisan Kode Program

### D1. Penulisan Kode Program class controllers/C\_ahp

Tabel D 1 Penulisan Kode Program class controllers/C\_ahp

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class C_ahp extends CI_Controller{

    /*
    * KONSTRUKTOR
    * Bertujuan untuk memuat model transaksi
    */
    public function __construct(){
        parent::__construct();
        if(!$this->session->userdata('user_on')){
            redirect ('login');
        }else{
            $this->load->model('M_ahp');
        }
    }

    /*
    Fungsi index
    Bertujuan untuk load data ketika kontroller pertama kali dipanggil
    */
    public function index(){
        $this->load->view('admin/header');
        // $this->load->view('admin/V_ahp',$data);
        $this->load->view('footer');}

    //=====Menghitung Jumlah Per Kolom=====
    public function bobot_kriteria() {
        $data['daftar_kriteria'] = $this->M_ahp->md_getkriteria()->result();
        $data['relasi_kriteria'] = $this->M_ahp->md_relasi_kriteria()->result();
        $data['jumlahKriteria'] = count($data['daftar_kriteria']);
    }
}

```

```

$jumlah = array();
//jumlah itu nama untuk data kriterianya
//id kriteria dari itu (yg kanan) dan id kriteria tujuan (yg atas)
foreach ($data['daftar_kriteria'] as $kriteria) {
    $jumlah1 = 0;
    foreach ($data['relasi_kriteria'] as $relasi) {
        //memfilter kriteria per kolom
        if($relasi->id_kriteria_tujuan == $kriteria->id_kriteria){
            $jumlah1 = $jumlah1 + $relasi->bobot;
        }
    }
    array_push($jumlah, $jumlah1);
}
$data['jumlah_per_kolom'] = $jumlah;

//=====Menghitung nilai per cell (nilai cell / jumlah)=====
$arrayVector = array();
foreach ($data['daftar_kriteria'] as $kriteria) {
    $nilai = array();
    $a = 0;
    foreach ($data['relasi_kriteria'] as $relasi) {
        if($relasi->id_kriteria_dari == $kriteria->id_kriteria){
            $nilai1 = $relasi->bobot / $jumlah[$a];
            array_push($nilai, $nilai1);
            $a++;
        }
    }
}

//=====Menghitung nilai vector=====
//BOBOT SINTESA
$vector = 0;
for($i = 0; $i < count($nilai); $i++){
    $vector = $vector + $nilai[$i];
}
//EIGEN VECTOR
$vector = $vector / $data['jumlahKriteria'];
$vector = number_format($vector, 3, '.', '');
array_push($arrayVector, $vector);
for ($i=0; $i < 9 ; $i++) {
    if(isset($arrayVector[$i])){
        $this->M_ahp->md_update_bobotprioritas($arrayVector[$i], $i+1);
    }
}

```

```

    }
}

//=====Menghitung Nilai Lamda=====
$lamda = 0;
for($i = 0; $i < count($arrayVector); $i++){
    $lamda = $lamda + ($arrayVector[$i] * $jumlah[$i]);
}

//=====Menghitung Nilai CI=====
$sci = ($lamda - $data['jumlahKriteria']) / ($data['jumlahKriteria'] - 1);
$sci = number_format($sci, 3, '.', '');

//=====Menghitung Nilai CR=====
$scr = $sci / $this->nilaiRI($data['jumlahKriteria']);
$scr = number_format($scr, 3, '.', '');

$data['nilai_lamda'] = $lamda;
$data['nilai_ci'] = $sci;
$data['nilai_cr'] = $scr;
$data['nilai_vector'] = $arrayVector;
$data['kriteria']=$this->M_ahp->md_getkriteria();
$this->load->view('admin/header');
$this->load->view('admin/V_ahp',$data);
$this->load->view('footer');
}

//=====EDIT pembobotan KRITERIA=====
public function edit_bobot_kriteria(){
    $data['daftar_kriteria'] = $this->M_ahp->md_getkriteria()->result();
    $data['relasi_kriteria'] = $this->M_ahp->md_relasi_kriteria()->result();
    $this->load->view('admin/header');
    $data['jumlah'] = count($data['daftar_kriteria']);
    $this->load->view('admin/V_admin_editkriteria', $data);

    //$this->load->view('admin/V_ahp',$data);

    $this->load->view('footer');
}

public function proses_edit_bobot_kriteria(){
    // i mewakili id kriteria dari

```

```
// j mewakili id kriteria tujuan
for($i = 1; $i <= 9; $i++){
    for($j = 1; $j <= 9; $j++){
        $data['bobot'] = $this->input->post($i.$j);
        $this->M_ahp->md_update_bobot_relasiKriteria($i, $j, $data);
    }
}
redirect(site_url()!/C_ahp/bobot_kriteria?suksesupdate');
}

//=====KETETAPAN NILAI RI=====
public function nilaiRI($jumlah){
    $nilai2;
    if($jumlah == '1'){
        $nilai2 = 0;
    } else if($jumlah == '2'){
        $nilai2 = 0;
    } else if($jumlah == '3'){
        $nilai2 = 0.58;
    } else if($jumlah == '4'){
        $nilai2 = 0.9;
    } else if($jumlah == '5'){
        $nilai2 = 1.12;
    } else if($jumlah == '6'){
        $nilai2 = 1.24;
    } else if($jumlah == '7'){
        $nilai2 = 1.32;
    } else if($jumlah == '8'){
        $nilai2 = 1.41;
    } else if($jumlah == '9'){
        $nilai2 = 1.45;
    } else if($jumlah == '10'){
        $nilai2 = 1.49;
    } else if($jumlah == '11'){
        $nilai2 = 1.51;
    } else if($jumlah == '12'){
        $nilai2 = 1.48;
    } else if($jumlah == '13'){
        $nilai2 = 1.56;
    } else if($jumlah == '14'){
        $nilai2 = 1.57;
    } else if($jumlah == '15'){
```

```
        $nilai2 = 1.59;
    }

    return $nilai2;
}

    }

?>
```

## D2. Penulisan Kode Program class controllers/C\_AHPguru

Tabel D 2 Penulisan Kode Program class controllers/C\_AHPguru

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class C_AHPguru extends CI_Controller{

    /*
    * KONSTRUKTOR
    * Bertujuan untuk memuat model transaksi
    */
    public function __construct(){
        parent::__construct();
        if(!$this->session->userdata('user_on')){
            redirect ('login');
        }else{
            $this->load->model('M_ahp');
        }
    }

    /*
    Fungsi index
    Bertujuan untuk load data ketika kontroller pertama kali dipanggil
    */
    public function index(){
        $this->load->view('guru/header');
        // $this->load->view('admin/V_ahp',$data);
        $this->load->view('footer');}
}
```

```

//=====Menghitung Jumlah Per Kolom=====
public function bobot_kriteria() {
$data['daftar_kriteria'] = $this->M_ahp->md_getkriteria()->result();
$data['relasi_kriteria'] = $this->M_ahp->md_relasi_kriteria()->result();
$data['jumlahKriteria'] = count($data['daftar_kriteria']);
$jumlah = array();
//jumlah itu nama untuk data kriterianya
//id kriteria dari itu (yg kanan) dan id kriteria tujuan (yg atas)
foreach ($data['daftar_kriteria'] as $kriteria) {
    $jumlah1 = 0;
    foreach ($data['relasi_kriteria'] as $relasi) {
        //memfilter kriteria per kolom
        if($relasi->id_kriteria_tujuan == $kriteria->id_kriteria){
            $jumlah1 = $jumlah1 + $relasi->bobot;
        }
    }
    array_push($jumlah, $jumlah1);
}
$data['jumlah_per_kolom'] = $jumlah;

//=====
==

//=====Menghitung nilai per cell (nilai cell / jumlah)=====
$arrayVector = array();
foreach ($data['daftar_kriteria'] as $kriteria) {
    $nilai = array();
    $a = 0;
    foreach ($data['relasi_kriteria'] as $relasi) {
        if($relasi->id_kriteria_dari == $kriteria->id_kriteria){
            $nilai1 = $relasi->bobot / $jumlah[$a];
            array_push($nilai, $nilai1);
            $a++;
        }
    }
}

//=====Menghitung nilai vector=====
//BOBOT SINTESA
$vector = 0;
for($i = 0; $i < count($nilai); $i++){
    $vector = $vector + $nilai[$i];
}

```

```
}
//EIGEN VECTOR
$vector = $vector / $data['jumlahKriteria'];
$vector = number_format($vector, 3, '.', '');
array_push($arrayVector, $vector);
for ($i=0; $i < 9 ; $i++) {
    if(isset($arrayVector[$i])){
        $this->M_ahp->md_update_bobotprioritas($arrayVector[$i], $i+1);
    }
}

//=====
==

}

//=====
=

//=====Menghitung Nilai Lamda=====
$lamda = 0;
for($i = 0; $i < count($arrayVector); $i++){
    $lamda = $lamda + ($arrayVector[$i] * $jumlah[$i]);
}

//=====
=

//=====Menghitung Nilai CI=====
$sci = ($lamda - $data['jumlahKriteria']) / ($data['jumlahKriteria'] - 1);
$sci = number_format($sci, 3, '.', '');

//=====
=

//=====Menghitung Nilai CR=====
$scr = $sci / $this->nilaiRI($data['jumlahKriteria']);
$scr = number_format($scr, 3, '.', '');

//=====
```



```
$data['nilai_lamda'] = $lamda;
$data['nilai_ci'] = $ci;
$data['nilai_cr'] = $cr;
$data['nilai_vector'] = $arrayVector;
//$id_kriteria = $id_kriteria;

$data['kriteria']=$this->M_ahp->md_getkriteria();
$this->load->view('guru/header');
$this->load->view('guru/V_AHPguru',$data);

$this->load->view('footer');
}

//=====KETETAPAN NILAI RI=====
public function nilaiRI($jumlah){
    $nilai2;
    if($jumlah == '1'){
        $nilai2 = 0;
    } else if($jumlah == '2'){
        $nilai2 = 0;
    } else if($jumlah == '3'){
        $nilai2 = 0.58;
    } else if($jumlah == '4'){
        $nilai2 = 0.9;
    } else if($jumlah == '5'){
        $nilai2 = 1.12;
    } else if($jumlah == '6'){
        $nilai2 = 1.24;
    } else if($jumlah == '7'){
        $nilai2 = 1.32;
    } else if($jumlah == '8'){
        $nilai2 = 1.41;
    } else if($jumlah == '9'){
        $nilai2 = 1.45;
    } else if($jumlah == '10'){
        $nilai2 = 1.49;
    } else if($jumlah == '11'){
        $nilai2 = 1.51;
    } else if($jumlah == '12'){
        $nilai2 = 1.48;
    }
}
```

```
} else if($jumlah == '13'){
    $nilai2 = 1.56;
} else if($jumlah == '14'){
    $nilai2 = 1.57;
} else if($jumlah == '15'){
    $nilai2 = 1.59;
}

return $nilai2;
}

?>
```

### D3. Penulisan Kode Program class controllers/C\_datapegawai

Tabel D 3 Penulisan Kode Program class controllers/C\_datapegawai

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class C_datapegawai extends CI_Controller{

    /*
    * KONSTRUKTOR
    * Bertujuan untuk memuat model transaksi
    */
    public function __construct(){

        parent::__construct();
        if(!$this->session->userdata('user_on')){
            redirect ('login');
        }else{
            $this->load->model('M_datapegawai');
        }
    }

    /*
    Fungsi index
    Bertujuan untuk load data ketika kontroller pertama kali dipanggil
```

```
*/
public function index(){
    $data = array(
'pelajaran' => $this->M_datapegawai-> getDataPegawai()
);
        $this->load->view('admin/header');
        $this->load->view('admin/V_datapegawai',$data);
        $this->load->view('footer');
    }
public function getubah(){
$this->M_datapegawai->ubah();
$this->load->view('admin/header');
        $this->load->view('admin/V_datapegawai');
        $this->load->view('footer');}

public function gettambahpegawai(){

    $data["kualifikasi"]=$this->M_datapegawai->get_kualifikasi();
    $data["kesehatan"]=$this->M_datapegawai->get_kesehatan();
    $data["catathukum"]=$this->M_datapegawai->get_catathukum();
    $data["sertifikat"]=$this->M_datapegawai->get_sertifikat();
    $data["golongan"]=$this->M_datapegawai->get_golongan();
    $data["DP3"]=$this->M_datapegawai->get_DP3();
    $data["tugastambahan"]=$this->M_datapegawai->get_tugastambahan();

    $this->load->view('admin/header');
        $this->load->view('admin/V_inputPegawai',$data);
        $this->load->view('footer');}

public function gettambahnilai ($id_user){
    $data=array('editpegawaix'=>$this->M_datapegawai-
>getDetailPegawai($id_user));

    $nilai = array();//nilai per kriteria
    $namakriteria = array();

    foreach ($data['editpegawaix'] as $row){
    array_push($nilai, $row['nilai']);
    }
}
```

```
for ($i = 1; $i <= 9; $i++) {
    $ambil = $nilai[$i - 1]; // bobot kriteria sub
    $data['namasub'] = $this->M_datapegawai->getDetailSub($ambil, $i);
    foreach ($data['namasub'] as $key) {
        array_push($namakriteria, $key['nama_subkriteria']);
    }
}

$data['nilai_user'] = $nilai;
$data['cetak'] = $namakriteria;

$data["kualifikasi"] = $this->M_datapegawai->get_kualifikasi();
$data["kesehatan"] = $this->M_datapegawai->get_kesehatan();
$data["catathukum"] = $this->M_datapegawai->get_catathukum();
$data["sertifikat"] = $this->M_datapegawai->get_sertifikat();
$data["golongan"] = $this->M_datapegawai->get_golongan();
$data["DP3"] = $this->M_datapegawai->get_DP3();
$data["tugastambahan"] = $this->M_datapegawai->get_tugastambahan();

$this->load->view('admin/header');
$this->load->view('admin/V_inputnilai', $data);
$this->load->view('footer');

public function submit(){
    $data['nilai'] = $this->M_datapegawai->masuk();
    foreach ($data['nilai'] as $key) { // perulangan database
        $sidnew = $key['id_user']; // id user terakhir
    }
    $id_user = $sidnew + 1;
    $banding = $this->input->post('nip'); // deklarasi
    $tahunajaran = $this->input->post('tahunajaran');
    $nope = $this->input->post('nope');
    $row['cek'] = $this->M_datapegawai->getNip($banding);
    $ckifok = count($row['cek']); // dijumlah ada brp nip yg sama
    // $data['ambil'] = $banding;

    // $nilai5 = usia
    $birthday = $this->input->post('tanggalL');
    // Convert Ke Date Time
    $bday = new DateTime($birthday);
```

```
$today = new DateTime();
$diff = $today->diff($biday);//pengurangan hr ini-ultah
if ($diff->y > '41' and $diff->y < '56'){
    $nilai5 = '3' ;
}elseif ($diff->y >= '30' and $diff->y <= '40') {
    $nilai5 = '2' ;
}else{
    $nilai5 = '1' ;
}
//nilai7 = pengalaman
$kerja = $this->input->post('tanggalMKerja');

// Convert Ke Date Time
$work = new DateTime($kerja);
$now = new DateTime();
$pengalaman_y = $now->diff($work);

if ($pengalaman_y->y > '25'){
    $nilai7 = '3' ;
}elseif ($pengalaman_y->y >= '11' and $pengalaman_y->y <= '25') {
    $nilai7 = '2' ;
}else{
    $nilai7 = '1' ;
}

//masuk ke capres
if (is_numeric($nope)){//jika nope numerik

if ($ckifok < 1){
    $result = $this->M_datapegawai->
    insert( $id_user,
    // $this->input->post('tahunajaran'),

    $this->input->post('nip'),
    $this->input->post('namaPeg'),
    $this->input->post('jenisKel'),
    $this->input->post('tanggalL'),
    $this->input->post('tempatTL'),
    $this->input->post('alamatS'),
    $this->input->post('nope'),
    $this->input->post('tanggalMKerja'),
```

```
$this->input->post('level')

);
//end deklarasi (buat input di db nilaipm)
for($i = 1; $i <= 9; $i++){

    if($i == '5'){
        $nilai=$nilai5;
    }
    elseif($i == '7'){
        $nilai=$nilai7;
    }else{
        $nilai = $this->input-
>post('nilai'.$i);
    }
    $id_kriteria_pm = $i;
    $hasil = 0;//
    //$this->M_datapegawai-
>insertnilai($id_kriteria_pm, $id_user, $nilai,$hasil);
    // $this->M_datapegawai-
>updatenilaipm($id_kriteria_pm, $id_user, $nilai,$tahunajaran)
    $this->M_datapegawai-
>insertnilai($id_kriteria_pm, $id_user, $nilai,$hasil,$tahunajaran);
    //$this->M_datapegawai->EditStatus();
    }

    if($result){
        $this->session->set_flashdata('message','SUKSES ENTRY
DATA');
        redirect('C_datapegawai');
    }else{
        $this->session->set_flashdata('message','SUKSES ENTRY
DATA');
        redirect('C_datapegawai');
    }
}

    $this->session-
>set_flashdata('message','GAGAL, Data Sudah Ada');
    redirect('C_datapegawai/gettambahpegawai');
}
```

```
}else{
    $this->session->set_flashdata('message','GAGAL, Nomor Hp harus
number');
    redirect('C_datapegawai/gettambahpegawai');
}
}
public function submitnilai(){

    $tahunajaran = $this->input->post('tahunajaran');
    $id_user = $this->input->post('id_user');
    //$nilai5 = usia
    $birthday = $this->input->post('tanggalL');
    // Convert Ke Date Time
    $biday = new DateTime($birthday);
    $today = new DateTime();
    $diff = $today->diff($biday);//pengurangan hr ini-ultah
    if ($diff->y > '41' and $diff->y < '56'){
        $nilai5 = '3' ;
    }elseif ($diff->y >= '30' and $diff->y <= '40') {
        $nilai5 = '2' ;
    }else{
        $nilai5 = '1' ;
    }

    //nilai7 = pengalaman
    $kerja = $this->input->post('tanggalMKerja');

    // Convert Ke Date Time
    $work = new DateTime($kerja);
    $now = new DateTime();
    $pengalaman_y = $now->diff($work);

    if ($pengalaman_y->y > '25'){
        $nilai7 = '3' ;
    }elseif ($pengalaman_y->y >= '11' and $pengalaman_y->y <= '25') {
        $nilai7 = '2' ;
    }else{
        $nilai7 = '1' ;
    }
    //cek nilai di tb_nilai_pm
```

```

    $data['ceknilaipm'] = $this->M_datapegawai-
>ceknilaipm($tahunajaran,$sid_user);
    $data['cekrekap'] = $this->M_datapegawai->cekrekap($tahunajaran);
    $cekrekap = count($data['cekrekap']);
    $ceknilaipm = count($data['ceknilaipm']);
    if ($ceknilaipm > 0 || $cekrekap >0){
        $this->session->set_flashdata('message','GAGAL!,DATA Sudah
ADA');
        redirect('C_datapegawai');
    }else{
//end deklarasi (buat input di db nilaipm)
for($i = 1; $i <= 9; $i++){
        if($i == '5'){
            $nilai=$nilai5;
        }
        elseif($i == '7'){
            $nilai=$nilai7;
        }else{
            $nilai = $this->input-
>post('nilai'.$i);
        }
        $id_kriteria_pm = $i;
        $hasil = 0;//
        $result = $this->M_datapegawai-
>updatenilaipm($id_kriteria_pm, $sid_user, $nilai,$tahunajaran);
        // $this->M_datapegawai->EditStatus();
        }
        $this->session-
>set_flashdata('message','Berhasil Entry Data');
        redirect('C_datapegawai');
    }
}

public function edit(){
    $data['nilai'] = $this->M_datapegawai->masuk();
    $id_user=$this->input->post('id');
    $banding= $this->input->post('nip');
    $banding2= $this->input->post('nip2');
    $nope = $this->input->post('nope');
    $jumKel = $this->input->post('jumKel');
    $row['cek'] = $this->M_datapegawai->getNip($banding);

```



```
$tahunajaran = $this->input->post('tahunajaran');
$skifok = count($row['cek']);
//$data['ambil']= $banding;

//$nilai5 = usia
$birthday = $this->input->post('tanggalL');
// Convert Ke Date Time
$biday = new DateTime($birthday);
$today = new DateTime();
$diff = $today->diff($biday);
if ($diff->y > '51' and $diff->y < '56'){
$nilai5 = '3' ;
}elseif ($diff->y >= '40' and $diff->y <= '50') {
    $nilai5 = '2' ;
}else{
    $nilai5 = '1' ;
}
//nilai7 = pengalaman
$kerja = $this->input->post('tanggalMKerja');

// Convert Ke Date Time
$work = new DateTime($kerja);
$now = new DateTime();
$pengalaman_y = $now->diff($work);

if ($pengalaman_y->y > '25'){
    $nilai7 = '3' ;
}elseif ($pengalaman_y->y >= '11' and $pengalaman_y->y <= '25') {
    $nilai7 = '2' ;
}else{
    $nilai7 = '1' ;
}

//end deklarasi
echo $this->input->post('id');
echo $this->input->post('nip');
echo $this->input->post('namaPeg');
echo $this->input->post('jenisKel');
```



```
        $result = $this->M_datapegawai->updateuser( $id_user,
            $this->input->post('id_level'),
            $this->input->post('nip'),
            $this->input->post('namaPeg'),
            $this->input->post('jenisKel'),
            $this->input->post('tanggalL'),
            $this->input->post('tempatTL'),
            $this->input->post('alamatS'),
            $this->input->post('nope'),
            $this->input->post('tanggalMKerja'));

        $this->session->set_flashdata('message','SUKSES EDIT
DATA');

        redirect('C_datapegawai');
    }else{
        $this->session->set_flashdata('message','GAGAL, NIP SUDAH
DIGUNAKAN');

        redirect('C_datapegawai');
    }else{
        $this->session->set_flashdata('message','GAGAL, NIP HARUS ANGKA');

        redirect('C_datapegawai');
    }
}

public function detail_pegawai($id_user){
    $data=array('username'=>$this->M_datapegawai->getDetailPegawai($id_user));
```

```
$nilai = array();
$namakriteria = array();
foreach ($data['username'] as $row){
array_push($nilai, $row['nilai']);
}
for ($i = 1; $i <= 9; $i++) {
    $ambil = $nilai[$i - 1];
    $data['namasub']=$this->M_datapegawai->getDetailSub($ambil,$i);
    foreach ($data['namasub'] as $key) {
        array_push($namakriteria,$key['nama_subkriteria']);
    }
}

$data['nilai_user']=$nilai;
$data['cetak']=$namakriteria;
$nilai = array();
foreach ($data['username'] as $row){
array_push($nilai, $row['nilai']);
}

    $data['nilai_user']=$nilai;
    $this->load->view('admin/header');
    $this->load->view('admin/V_detailPegawai',$data);
    $this->load->view('footer');}

        public function form_editpegawai ($id_user){
    $data=array('editpegawai'=>$this->M_datapegawai-
>getDetailPegawai($id_user));
    $nilai = array();//nilai per kriteria
    $namakriteria = array();
    foreach ($data['editpegawai'] as $row){
array_push($nilai, $row['nilai']);
}
for ($i = 1; $i <= 9; $i++) {
    $ambil = $nilai[$i - 1];//bobot kriteria sub
    $data['namasub']=$this->M_datapegawai->getDetailSub($ambil,$i);
    foreach ($data['namasub'] as $key) {
        array_push($namakriteria,$key['nama_subkriteria']);
    }
}

    $data['nilai_user']=$nilai;
```

```

$data['cetak']=$namakriteria;

$data["kualifikasi"]=$this->M_datapegawai->get_kualifikasi();
$data["kesehatan"]=$this->M_datapegawai->get_kesehatan();
$data["catathukum"]=$this->M_datapegawai->get_catathukum();
$data["sertifikat"]=$this->M_datapegawai->get_sertifikat();
$data["golongan"]=$this->M_datapegawai->get_golongan();
$data["DP3"]=$this->M_datapegawai->get_DP3();
$data["tugastambahan"]=$this->M_datapegawai->get_tugastambahan();
        $this->load->view('admin/header');
    $this->load->view('admin/V_editPegawai',$data);
    $this->load->view('footer');}

        public function cari(){
            $key = $this->input->post('cari');
            $data['cari'] = $this->M_datapegawai-
>getCari($key);
                $this->load->view('admin/header');
            $this->load->view('admin/V_caripegawai',$data);
            $this->load->view('footer');
        }

        public function caritahun(){
            $key = $this->input->post('caritahun');
            $data['cari'] = $this->M_datapegawai-
>getCaritahun($key);
                $this->load->view('admin/header');
            $this->load->view('admin/V_caripegawai',$data);
            $this->load->view('footer');
        }
    }

        public function update_datapegawai (){
            $id_user = $this->input->get('id_user');
            $input['id_level'] = $this->input->post('id_level');
            $input['nip'] = $this->input->post('nip');
            $input['namaPeg'] = $this->input->post('namaPeg');
            $input['jenisKel'] = $this->input->post('jenisKel');
            $input['tanggalL'] = $this->input->post('tanggalL');
            $input['tempatTL'] = $this->input->post('tempatTL');
            $input['alamatS'] = $this->input->post('alamatS');
            $input['nope'] = $this->input->post('nope');
            $input['tanggalMKerja'] = $this->input->post('tanggalMKerja');
            if (is_numeric($input['nope'])){

```

```
$insert = $this->M_datapegawai->getEditPegawai($input,$id_user);
$this->session->set_flashdata('message','Berhasil, Data Berhasil diupdate');
redirect('C_datapegawai');

        }else{
            $this->session->set_flashdata('message','GAGAL, Nomor
Hp harus number');
            redirect('C_datapegawai');
        }
    }

    public function delete_datapegawai ($id_user){
        $data=array('username'=>$this->M_datapegawai->getHapusPegawai($id_user));
        $this->load->view('admin/header');
        $this->load->view('admin/V_datapegawai',$data);
        $this->load->view('footer');
        //redirect('C_datapegawai');
        // echo "<script>
        // alert ('User Barhasil Dinonaktifkan!!!');
        // </script>";
        // echo "<META HTTP-EQUIV='Refresh' CONTENT='0;
URL=http://localhost/kepsek2/index.php/C_datapegawai'>";
        $this->session->set_flashdata('message','User Berhasil
Dinonaktifkan');
        redirect('C_datapegawai');
    }

    public function aktv_datapegawai ($id_user){
        $data=array('username'=>$this->M_datapegawai-
>getAktvPegawai($id_user));
        $this->load->view('admin/header');
        $this->load->view('admin/V_datapegawai',$data);
        $this->load->view('footer');
        $this->session->set_flashdata('message','User Berhasil
Diaktifkan');
        redirect('C_datapegawai');
        //redirect('C_datapegawai');
    }

    public function reset_pegawai ($id_user){
        $data=array('username'=>$this->M_datapegawai-
>reset_pegawai($id_user));
        $this->load->view('admin/header');
```

```

        $this->load->view('admin/V_datapegawai',$data);
        $this->load->view('footer');
        $this->session->set_flashdata('message','Password Berhasil
Direset dengan Hasil = 1234');
        redirect('C_datapegawai');
        //redirect('C_datapegawai');
    }
}
?>

```

#### D4. Penulisan Kode Program class controllers/C\_datauser

Tabel D 4 Penulisan Kode Program class controllers/C\_datauser

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class C_datauser extends CI_Controller{

    /*
    * KONSTRUKTOR
    * Bertujuan untuk memuat model transaksi
    */
    public function __construct(){
        parent::__construct();
        if(!$this->session->userdata('user_on')){
            redirect ('login');
        }else{
            $this->load->model('Mdatauser');
        }
    }

    /*
    Fungsi index
    Bertujuan untuk load data ketika kontroller pertama kali dipanggil
    */
    public function index(){
        $data["diri"]=$this->Mdatauser->getDataDiri();
        $this->load->view('guru/header');
        $this->load->view('guru/V_profile',$data);
    }
}

```

```
        $this->load->view('footer');
    }

    public function edit(){
        $id_user = $this->session->userdata('id');
        $input['nip'] = $this->input->post('nip');
        $input['nip2'] = $this->input->post('nip2');
        $input['namaPeg'] = $this->input->post('nama');
        $input['jenisKel'] = $this->input->post('jenis_kelamin');
        $input['tanggalL'] = $this->input->post('tanggal_lahir');
        $input['tempatTL'] = $this->input->post('tempat');
        $input['alamatS'] = $this->input->post('alamat');
        $input['nope'] = $this->input->post('no_hp');
        $banding = $input['nip'];
        $row['cek'] = $this->Mdatauser->getNip($banding);
        $ckifok = count($row['cek']);

        if ($ckifok < 1 or $input['nip'] == $input['nip2']){
            if (is_numeric($input['nip'])){
                if (is_numeric($input['nope'])){
                    $insert = $this->Mdatauser->getEditGuru($input,$id_user);
                    $this->session->set_flashdata('message','Berhasil, Data Berhasil
di update');
                    redirect('C_datauser');
                }else{
                    $this->session->set_flashdata('message','Gagal, Nomer HP harus
angka');
                    redirect('C_datauser');
                }
            }else{
                $this->session->set_flashdata('message','Gagal, NIP harus angka');
                redirect('C_datauser');
            }
        }else{
            $this->session->set_flashdata('message','Gagal, NIP sudah
digunakan');
            redirect('C_datauser');
        }
    }

    public function editpass(){
        $data["diri"]=$this->Mdatauser->getDataDiri();
        $this->load->view('guru/header');
        $this->load->view('guru/V_edit_profile',$data);
    }
}
```



```

        $this->load->view('footer');
    }
    public function edit_pass_act(){
        $data["diri"]=$this->Mdatauser->getDataDiri();
        $sid_user = $this->session->userdata('id');
        $passlama = $this->input->post('passlama');
        $passbaru = $this->input->post('passbaru');
        $passbaru2 = $this->input->post('passbaru2');
        foreach ($data['diri'] as $row) {
        }
        if ($passlama == $row['Password']){
            if ($passbaru == $passbaru2){
                $insert = $this->Mdatauser-
>getEditGuruPass($sid_user,$passbaru);
                $this->session-
>set_flashdata('message','Password Berhasil Diganti');
                redirect('C_datauser/editpass');
            }else{
                $this->session-
>set_flashdata('message','Password Baru Yang Anda Masukkan Tidak Sama');
                redirect('C_datauser/editpass');
            }
        }else{
            $this->session->set_flashdata('message','Password Lama
Yang Anda Masukkan Salah');
            redirect('C_datauser/editpass');
        }
    }
}
?>

```

#### D5. Penulisan Kode Program class controllers/C\_hasil

Tabel D 5 Penulisan Kode Program class controllers/C\_hasil

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class C_hasil extends CI_Controller{

```

```
/*
 * KONSTRUKTOR
 * Bertujuan untuk memuat model transaksi
 */
public function __construct(){
    parent::__construct();
    if(!$this->session->userdata('user_on')){
        redirect ('login');
    }else{
        $this->load->model('M_hasil');
    }
}

/*
 Fungsi index
 Bertujuan untuk load data ketika kontroller pertama kali dipanggil
 */
public function index(){
    $data['tahun']= $this->M_hasil->gettahun();
    $data['kriteria'] = $this->M_hasil->getKriteria();
    $data['jum_capres'] = $this->M_hasil->jumlahcapres();
    $data['jumlah_capres']=count($data['jum_capres']);
    $data['data_capres'] = $this->M_hasil->getCapres();
    $data['pm'] = $this->M_hasil->getNilaiPM();
    $data['ideal'] = $this->M_hasil->getIdeal();
    $jum_kri = count($data['kriteria']);
    //$data['bobot'] = $this->M_hasil->getbobotpm();
    $data['jum_capres'] = $this->M_hasil->jumlahcapres();
    $capres=count($data['jum_capres']);
    $data['reset']=$this->M_hasil->getreset();

    $gap=0.5;

    $hasil_akhir = array();
    foreach ($data['pm'] as $nilai){
        foreach ($data['kriteria'] as $kriteria){
            if($nilai['id_kriteria_pm'] == $kriteria['id_kriteria']){
                //$skurang=0;
                $penentuan=$nilai['hasil'] * $kriteria['bobot_prioritas'];
                array_push($hasil_akhir, $penentuan);
            }
        }
    }
}
```

```
    }
  }
  $nilai_akhir = 0;
  $hasil_penentuan = array();
  $n = 0;
  foreach ($data['data_capres'] as $user){
    for($j = 1; $j <=$jum_kri ; $j++){
      $nilai_akhir=$nilai_akhir + $hasil_akhir[$n];
      $n++;
    }
    $nilai = $nilai_akhir;
    array_push($hasil_penentuan,$nilai_akhir);
    $id_user = $user['id_user'];
    $this->M_hasil->update_nilai($id_user, $nilai);
    $nilai_akhir = 0;
  }
  $data['capres'] = $capres;
  $data['yeay'] = $hasil_penentuan;
  $data['cetak_pm'] = $hasil_akhir;

  $data['cetak_yeay'] = $this->M_hasil->gethasil();

  $this->load->view('admin/header');
  $this->load->view('admin/V_hasil',$data);
  $this->load->view('footer');
}

public function ganti_kepala(){
    $data['kepala_baru']=$this->M_hasil-
>cek_kepala_sekolah_baru();
    foreach ($data['kepala_baru'] as $id) {
      $id_ganti = $id['id_user'];
    }
    $data['rekap']=$this->M_hasil->ambil_data_rekap();
    //cek data rekap kosong atau tidak
    foreach ($data['rekap'] as $upload_rekap) {
      $id_user=$upload_rekap['id_user'];
      $tahunajaran=$upload_rekap['tahun_ajaran'];
    }
}
```

```

        $data['cekdatarekap']=$this->M_hasil-
>cekdatarekap($tahunajaran,$sid_user);
        $jumlahdatarekap = count($data['cekdatarekap']);
        if ($jumlahdatarekap < 1){
            foreach ($data['rekap'] as $upload_rekap) {
                $id_user=$upload_rekap['id_user'];
                $id_kriteria_pm=$upload_rekap['id_kriteria_pm'];
                $nilai=$upload_rekap['nilai'];
                $tahunajaran=$upload_rekap['tahun_ajaran'];
                $hasil=$upload_rekap['hasil'];
                $data['nilai_akhir']=$this->M_hasil-
>ambil_data_capres($sid_user);
                foreach ($data['nilai_akhir'] as $key) {
                    $nilai_akhir = $key['hasil'];
                }

                $this->M_hasil-
>insertnilairekap($id_kriteria_pm,$sid_user,$nilai,$tahunajaran,$hasil,$nilai_akhir)
;

                    //echo $sid_user;
                }
                //$this->M_hasil->change_head($sid_ganti);
                $this->session->set_flashdata('message','Berhasil, Rekap
data telah dilakukan');
                redirect('C_rekap');
            }else{
                $this->session->set_flashdata('message','Gagal, Rekap data
tidak ada yang berbeda');
                redirect('C_rekap');
            }
        }
    // public function simpan_next(){
    // $this->M_hasil->delete_next();
    // $data['jum_capres'] = $this->M_hasil->jumlahcapres();
    // $capres=count($data['jum_capres']);
    // for($i = 1; $i <=$capres ; $i++){
    //     $nilai = $this->input->post('total'.$i);
    //     $id_user = $this->input->post('id_user'.$i);
    //     $this->M_hasil->insert_next($id_user, $nilai);
    // }
    // redirect(site_url().'/C_profile');
    // }

```

```

    }
?>

```

#### D6. Penulisan Kode Program class controllers/c\_manager

Tabel D 6 . Penulisan Kode Program class controllers/c\_manager

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class c_manager extends CI_Controller{

    /*
    * KONSTRUKTOR
    * Bertujuan untuk memuat model transaksi
    */
    public function __construct(){
        parent::__construct();
        if(!$this->session->userdata('user_on')){
            redirect ('login');
        }else{
            $this->load->model('M_manager');
        }
    }

    public function dashboard(){
        // $id_user = $this->session->userdata('id');
        // $data['user'] = $this->M_manager->
        >getlogin($id_user);
        $this->load->view('guru/header');
        $this->load->view('guru/V_dash');
        $this->load->view('footer');
    }

    public function index(){
        $data['nilai'] = $this->M_manager->getNilaiPM();
        $this->load->view('guru/header');
        $this->load->view('V_manager',$data);
        $this->load->view('footer');
    }

    public function detailnilai(){
        $id_user = $this->session->userdata('id');

```

```
$data['rekap']=$this->M_manager->
getrekapdetailnilai($id_user);
//foreach ($data['rekap'] as $key){ echo $key['nilai']; }
$nilai = array();
foreach ($data['rekap'] as $key) {
    $data['nama']=$key['namaPeg'];
    //echo $key['namaPeg'];
    $id_kriteria = $key['id_kriteria_pm'];
    $nilai_kriteria = $key['nilai'];
    $data['nilai']=$this->M_manager-
>getsubkriteria($id_kriteria,$nilai_kriteria);
    foreach ($data['nilai'] as $row){
        array_push($nilai,$row['nama_subkriteria']);
    }
}
//echo $data['nama']."<br>";
$data['nilai']=$nilai;
$this->load->view('guru/header');
$this->load->view('guru/v_detail_rekap_user',$data);
$this->load->view('footer');
}

public function hasil_akhir(){
    $data['tahun']= $this->M_manager->gettahun();
    $data['ver'] = $this->M_manager->getver();
    $data['pm'] = $this->M_manager->getNilaiPM();
    $data['cetak_yeay'] = $this->M_manager->gethasil();

    $this->load->view('guru/header');
    $this->load->view('guru/V_hasil',$data);
    $this->load->view('footer');
}
public function detailnilaiuser($id_user,$tahunajaran){
    $data['rekap']=$this->M_manager->
getrekapdetailnilai2($id_user,$tahunajaran);
    $nilai = array();
    foreach ($data['rekap'] as $key) {
        $data['nama']=$key['namaPeg'];
```

```

$key['id_kriteria_pm'];
>getsubkriteria($id_kriteria,$nilai_kriteria);
    $sid_kriteria =
    $nilai_kriteria = $key['nilai'];
    $data['nilai']=$this->M_manager-
    foreach ($data['nilai'] as $row){
        array_push($nilai,$row['nama_subkriteria']);
        }
        }
        //echo $data['nama']." </br>";
        $data['nilai']=$nilai;
        $this->load->view('guru/header');
        $this->load-
>view('guru/V_detailnilaipribadi2',$data);
        $this->load->view('footer');
    }

    public function verifikasi(){
        $insert = $this->M_manager->EditStatus();
        $this->session->set_flashdata('message','Berhasil, Data Berhasil
di verifikasi');
        redirect('c_manager/hasil_akhir');
    }

    public function kriteria(){
        $data = array(
        'pelajaran' => $this ->M_manager-> getkriteria()
        );

        $this->load->view('guru/header');
        $this->load->view('guru/V_kriteria',$data);
        $this->load->view('footer');
    }
}
?>

```

Tabel D 7 Penulisan Kode Program class controllers/C\_pm

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class C_pm extends CI_Controller{

    /*
    * KONSTRUKTOR
    * Bertujuan untuk memuat model transaksi
    */
    public function __construct(){
        parent::__construct();
        if(!$this->session->userdata('user_on')){
            redirect ('login');
        }else{
            $this->load->model('M_pm');
        }
    }

    /*
    Fungsi index
    Bertujuan untuk load data ketika kontroller pertama kali dipanggil
    */
    public function index(){
        $data['kriteria'] = $this->M_pm->getKriteria();
            $data['tahun']= $this->M_pm->gettahun();
        $data['jum_capres'] = $this->M_pm->jumlahcapres();
        $data['jumlah_capres']=count($data['jum_capres']);
        $data['data_capres'] = $this->M_pm->getCapres();
        $data['pm'] = $this->M_pm->getNilaiPM();
        $data['ideal'] = $this->M_pm->getIdeal();
        $data['jumlahkriteriapm'] = count($data['kriteria']);
        $data['calon']=count($data['pm'])/$data['jumlahkriteriapm'];
        // $data['bobot'] = $this->M_pm->getbobotpm();
        //foreach ($data['bobot'] as $gap){ }

        $hasil = array();
        $hasil_bobot = array();
        foreach ($data['kriteria'] as $ideal){
            foreach ($data['pm'] as $nilai){
                if($nilai['id_kriteria_pm'] == $ideal['id_kriteria_pm']){
                    //$kurang=0;
```



```
$kurang=$nilai['nilai']-$ideal['ideal'];
if ($kurang != 0){
    $hasil_nilai_pm= 3 + ($kurang*0.5);//gap nilai 1/2
}else{
    $hasil_nilai_pm= 3;
}
array_push($hasil, $kurang);
array_push($hasil_bobot, $hasil_nilai_pm);
    $user = $nilai['id_user'];
    $id_kriteria = $nilai['id_kriteria_pm'];
    $hasil2 = $hasil_nilai_pm;
    $this->M_pm->update_nilai_pm($id_kriteria, $user, $hasil2);
}
}
}

$data['nilai_bobot'] = $hasil_bobot;
$data['jumlahbobot'] = count($data['nilai_bobot']);
$data['pengurangan']=$hasil;
$data['cetak_pm']=$hasil_bobot;

$this->load->view('admin/header');
$this->load->view('admin/V_pm',$data);
$this->load->view('footer');
}
public function nilaiideal(){
    $data = array(
        'nilaiidealx' => $this->M_pm->getnilaiideal()
    );
    //$data['ideal'] = $this->M_pm->getIdeal();

    $this->load->view('admin/header');
    $this->load->view('admin/V_nilaiideal',$data);
    $this->load->view('footer');
}

//form_editideal

public function form_editideal($id_kriteria_pm){
    $data=array('editidealx'=>$this->M_pm->getDetailIdeal($id_kriteria_pm));
    $this->load->view('admin/header');
```

```

$this->load->view('admin/V_editideal',$data);
$this->load->view('footer');}

public function update_nilaiideal(){
$id_kriteria_pm = $this->input->get('id_kriteria_pm');
$input['nama_kriteria_pm'] = $this->input->post('nama_kriteria_pm');
$input['ideal'] = $this->input->post('ideal');
if (is_numeric($input['ideal'])) {
$insert = $this->M_pm->getEditIdeal($input,$id_kriteria_pm);
$this->session->set_flashdata('message','Berhasil, Data berhasil disimpan');
redirect('C_pm/nilaiideal');
} else {
$this->session->set_flashdata('message','GAGAL, Nilai ideal harus
angka);
redirect('C_pm/form_editideal/'.$id_kriteria_pm);
}
}

public function simpan_next(){
$this->M_pm->delete_next();
$data['jum_capres'] = $this->M_pm->jumlahcapres();
$capres=count($data['jum_capres']);
for($i = 1; $i <=$capres ; $i++){
$nilai = $this->input->post('total'.$i);
$id_user = $this->input->post('id_user'.$i);
$this->M_pm->insert_next($id_user, $nilai);
}
redirect(site_url().'/C_profile');
}

}

?>

```

#### D8. Penulisan Kode Program class controllers/C\_PMguru

Tabel D 8 Penulisan Kode Program class controllers/C\_Pmguru

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class C_PMguru extends CI_Controller{

/*

```

```
* KONSTRUKTOR
* Bertujuan untuk memuat model transaksi
*/
public function __construct(){
    parent::__construct();
    if(!$this->session->userdata('user_on')){
        redirect ('login');
    }else{
        $this->load->model('M_pm');
    }
}

/*
Fungsi index
Bertujuan untuk load data ketika kontroller pertama kali dipanggil
*/
public function index(){

    $data['kriteria'] = $this->M_pm->getKriteria();
        $data['tahun'] = $this->M_pm->gettahun();
    $data['jum_capres'] = $this->M_pm->jumlahcapres();
    $data['jumlah_capres'] = count($data['jum_capres']);
    $data['data_capres'] = $this->M_pm->getCapres();
    $data['pm'] = $this->M_pm->getNilaiPM();
    $data['ideal'] = $this->M_pm->getIdeal();
    $data['jumlahkriteriapm'] = count($data['kriteria']);
    $data['calon'] = count($data['pm'])/$data['jumlahkriteriapm'];
    $gap = 0.5;
    $hasil = array();
    $hasil_bobot = array();
    foreach ($data['kriteria'] as $ideal){
        foreach ($data['pm'] as $nilai){
            if($nilai['id_kriteria_pm'] == $ideal['id_kriteria_pm']){//itu dikurangnya
kriteranya sama
                //$kurang=0;
                $kurang=$nilai['nilai']-$ideal['ideal'];
                if ($kurang != 0){
                    $hasil_nilai_pm= 3 + ($kurang*$gap['nilai']);//gap nilai 1/2
                }else{
                    $hasil_nilai_pm= 3;
                }
            }
        }
    }
}
```

```
array_push($hasil, $kurang);//
array_push($hasil_bobot, $hasil_nilai_pm);
    $user = $nilai['id_user'];//tanya
    $id_kriteria = $nilai['id_kriteria_pm'];
    $hasil2 = $hasil_nilai_pm;
    $this->M_pm->update_nilai_pm($id_kriteria, $user, $hasil2);
}
}
}

$data['nilai_bobot'] = $hasil_bobot;
$data['jumlahbobot'] = count($data['nilai_bobot']);
$data['pengurangan']=$hasil;
$data['cetak_pm']=$hasil_bobot;

$this->load->view('guru/header');
$this->load->view('admin/V_pm',$data);
$this->load->view('footer');
}
public function nilaiideal(){
    $data = array(
        'nilaiidealx' => $this ->M_pm-> getnilaiideal()
    );
    //$data['ideal'] = $this->M_pm->getIdeal();

    $this->load->view('guru/header');
    $this->load->view('guru/V_nilaiideal',$data);
    $this->load->view('footer');
}

//form_editideal

public function form_editideal($id_kriteria_pm){
    $data=array('editidealx'=>$this->M_pm->getDetailIdeal($id_kriteria_pm));
    $this->load->view('admin/header');
    $this->load->view('admin/V_editideal',$data);
    $this->load->view('footer');}

public function update_nilaiideal(){
    $id_kriteria_pm = $this->input->get('id_kriteria_pm');
    $input['nama_kriteria_pm'] = $this->input->post('nama_kriteria_pm');
```

```

$input['ideal'] = $this->input->post('ideal');
if (is_numeric($input['ideal'])) {
    $insert = $this->M_pm->getEditIdeal($input,$id_kriteria_pm);
    $this->session->set_flashdata('message','Berhasil, Data berhasil disimpan');
    redirect('C_pm/nilaiideal');
    }else{
        $this->session->set_flashdata('message','GAGAL, Nilai ideal harus
angka');
        redirect('C_pm/form_editideal/'.$id_kriteria_pm);
    }
}
public function simpan_next(){
    $this->M_pm->delete_next();
    $data['jum_capres'] = $this->M_pm->jumlahcapres();
    $capres=count($data['jum_capres']);
    for($i = 1; $i <=$capres ; $i++){
        $nilai = $this->input->post('total'.$i);
        $id_user = $this->input->post('id_user'.$i);
        $this->M_pm->insert_next($id_user, $nilai);
    }
    redirect(site_url().'/C_profile');
}
}
?>

```

#### D9. Penulisan Kode Program class controllers/C\_profile

Tabel D 9 Penulisan Kode Program class controllers/C\_profile

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class C_profile extends CI_Controller{

    /*
    * KONSTRUKTOR
    * Bertujuan untuk memuat model transaksi
    */
    public function __construct(){
        parent::__construct();
        if(!$this->session->userdata('user_on')){

```

```
        redirect ('login');
    }else{
        $this->load->model('Mprofile');
    }
}

/*
Fungsi index
Bertujuan untuk load data ketika kontroller pertama kali dipanggil
*/
public function index(){
    $ambil = $_GET['reference'];
    $data["namaPeg"]=$this->Mprofile->get_namaPeg($ambil);
    $data["kualifikasi"]=$this->Mprofile->get_kualifikasi();
    $data["usia"]=$this->Mprofile->get_usia($ambil);
    $data["kesehatan"]=$this->Mprofile->get_kesehatan();
    $data["catathukum"]=$this->Mprofile->get_catathukum();
    $data["sertifikat"]=$this->Mprofile->get_sertifikat();
    $data["pengalaman"]=$this->Mprofile->get_pengalaman($ambil);
    $data["golongan"]=$this->Mprofile->get_golongan();
    $data["DP3"]=$this->Mprofile->get_DP3();
    $data["tugastambahan"]=$this->Mprofile->get_tugastambahan();
    $this->load->view('admin/header');
    $this->load->view('admin/V_profile',$data);
    $this->load->view('footer');
}

public function cari(){
    $key = $this->input->post('cari');
    $data['krit']=$this->Mprofile->getKrit();
    $data['data_capres'] = $this->Mprofile->getCapres2($key);
    $data['pm'] = $this->Mprofile->getNilaiPM();
    //$data['cari'] = $this->Mprofile->getCari($key);
    $this->load->view('admin/header');
    $this->load->view('admin/V_carinilai',$data);
    $this->load->view('footer');
}

public function penilaian(){
    $data['krit']=$this->Mprofile->getKrit();
    $data['data_capres'] = $this->Mprofile->getCapres();
```

```
$data['pm'] = $this->Mprofile->getNilaiPM();
$this->load->view('admin/header');
$this->load->view('admin/penilaian',$data);
$this->load->view('footer');}
/*
Fungsi submit
Bertujuan untuk memasukkan data kedalam database
*/
public function edit(){

    $compare=array();
    $compare_nama=array();
    $i=0;
    $id_user = $_GET['id'];
    $data["data"]=$this->Mprofile->get_cek2($id_user);
    foreach ($data["data"] as $row) {
        array_push($compare, $row['nilai']);
        $i++;
    }
    foreach ($data["data"] as $row) {
        array_push($compare_nama, $row['nama_subkriteria']);
        $i++;
    }
    $data["nama"]=$this->Mprofile->get_namaPeg2($id_user);
    $data["namaPeg"]=$this->Mprofile->get_namaPeg3();
    $data['compare']=$compare;
    $data['compare_nama']=$compare_nama;
    $data["kualifikasi"]=$this->Mprofile->get_kualifikasi2();
    $data["usia"]=$this->Mprofile->get_usia2();
    $data["kesehatan"]=$this->Mprofile->get_kesehatan2();
    $data["catathukum"]=$this->Mprofile->get_catathukum2();
    $data["sertifikat"]=$this->Mprofile->get_sertifikat2();
    $data["pengalaman"]=$this->Mprofile->get_pengalaman2();
    $data["golongan"]=$this->Mprofile->get_golongan2();
    $data["DP3"]=$this->Mprofile->get_DP32();
    $data["tugastambahan"]=$this->Mprofile->get_tugastambahan2();
    $this->load->view('admin/header');
    $this->load->view('admin/edit',$data);
    $this->load->view('footer');
}
public function delete(){
    $id_user = $_GET['id'];
```

```
        //echo $id_user;
        $this->Mprofile->delete2($id_user);
        $this->session->set_flashdata('message','Berhasil Dihapus');
        redirect('C_profile/penilaian');
    }
    public function submit(){
        $id_user = $this->input->post('id_user');
        $data['cek']=$this->Mprofile->get_cek($id_user); //tanya
        $cek = count($data['cek']);
echo $id_user;
echo $cek;
//echo $cek['id_user'];
        if ($cek < 1){
        for($i = 1; $i <= 9; $i++){
            $nilai = $this->input->post('nilai'.$i);
            $id_kriteria_pm = $i;
            $hasil = 0;
            $this->Mprofile->insert($id_kriteria_pm, $id_user, $nilai,$hasil);
        }
            $this->session->set_flashdata('message','Berhasil, Data Nilai
Berhasil DiMasukkan');
            redirect('C_pm');
        }else{
            $this->session->set_flashdata('message','GAGAL, Data Sudah Ada');
            redirect('C_pm');
        }
    }

    public function editpenilaian(){
        $id_user = $_GET['id'];
        $kualifikasi = $_POST['nilai1'];
        $usia = $this->input->post('nilai5');
        $kesehatan = $this->input->post('nilai2');
        $sertifikat = $this->input->post('nilai3');
        $golongan = $this->input->post('nilai4');
        $catathukum = $this->input->post('nilai6');
        $pengalaman = $this->input->post('nilai7');
        $dp3 = $this->input->post('nilai8');
        $tugastambahan = $this->input->post('nilai9');

        //if (isset($id_user)){
        for($i = 1; $i <= 9; $i++){
```



```

        $nilai = $this->input->post('nilai'.$i);
        $id_kriteria_pm = $i;
        $hasil = 5;
        $this->Mprofile-
>updatenilai($id_kriteria_pm,$id_user,$nilai,$hasil);

    }
        $this->session->set_flashdata('message','Berhasil
Mengubah Penilaian Guru');
        redirect('C_profile/penilaian');
    }
}
?>

```

#### D10. Penulisan Kode Program class controllers/C\_rekap

Tabel D 10 Penulisan Kode Program class controllers/C\_rekap

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class C_rekap extends CI_Controller{

    /*
    * KONSTRUKTOR
    * Bertujuan untuk memuat model transaksi
    */
    public function __construct(){

        parent::__construct();
        if(!$this->session->userdata('user_on')){
            redirect ('login');
        }else{
            $this->load->model('M_rekap');
        }
    }

    /*
    Fungsi index
    Bertujuan untuk load data ketika kontroller pertama kali dipanggil
    */
}

```

```
public function index(){
    $data = array(
        'pelajaran' => $this->M_rekap-> getrekap()
    );
    $data['tahuncari'] = 0;
        $this->load->view('admin/header');
        $this->load->view('admin/V_rekap',$data);
        $this->load->view('footer');
    }
public function detail($tahunajaran){
    $data['rekap']=$this->M_rekap-> getrekapdetail($tahunajaran);
    foreach ($data['rekap'] as $key) {
        //echo $key['hasil'];
    }
    $this->load->view('admin/header');
    $this->load->view('admin/V_detail_rekap',$data);
    $this->load->view('footer');
}
public function detailnilai($id_user,$tahunajaran){
    $data['rekap']=$this->M_rekap->
getrekapdetailnilai($id_user,$tahunajaran);
    $nilai = array();
    foreach ($data['rekap'] as $key) {
        $data['nama']=$key['namaPeg'];
        $id_kriteria = $key['id_kriteria_pm'];
        $nilai_kriteria = $key['nilai'];
        $data['nilai']=$this->M_rekap-
>getsubkriteria($id_kriteria,$nilai_kriteria);
        foreach ($data['nilai'] as $row){
            array_push($nilai,$row['nama_subkriteria']);
        }
    }
    //echo $data['nama']."<br>";
    $data['nilai']=$nilai;
    $this->load->view('admin/header');
    $this->load->view('admin/V_detail_rekap_user',$data);
    $this->load->view('footer');
}
public function cari(){
    $key = $this->input->post('cari');
    $data['tahuncari'] = $key;
```

```

                                $data['pelajaran']      =      $this->M_rekap-
>getCari($key);
                                $this->load->view('admin/header');
                                $this->load->view('admin/V_rekap',$data);
                                $this->load->view('footer');
                                }
public function detail_pegawai($id_user,$tahunajaran,$tahunnext){
    $data=array('username'=>$this->M_rekap-
>getDetailPegawai($id_user,$tahunajaran,$tahunnext));
    //echo $id_user;
    //echo $tahunajaran;
    //echo $tahunnext;
    $nilai = array();
    $namakriteria = array();
    foreach ($data['username'] as $row){
array_push($nilai, $row['nilai']);
    }
    for ($i = 1; $i <= 9; $i++) {
        $sambil = $nilai[$i - 1];
        $data['namasub']=$this->M_rekap->getDetailSub($sambil,$i);
        foreach ($data['namasub'] as $key) {
            array_push($namakriteria,$key['nama_subkriteria']);
        }
    }

    $data['nilai_user']=$nilai;
    $data['cetak']=$namakriteria;
    $nilai = array();
    foreach ($data['username'] as $row){
array_push($nilai, $row['nilai']);
    }
    $data['nilai_user']=$nilai;
    $this->load->view('admin/header');
    $this->load->view('admin/V_detailrekap',$data);
    $this->load->view('footer');
}
}

?>

```

#### D11. Penulisan Kode Program class controllers/dashboard

Tabel D 11 Penulisan Kode Program class controllers/dashboard

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Dashboard extends CI_Controller {

    /*
    * KONSTRUKTOR
    * Bertujuan untuk memuat model dashboard
    */
    public function __construct(){
        parent::__construct();
        if(!$this->session->userdata('user_on') || $this->session-
>userdata('level') != 'admin'){
            redirect("login");
        }
    }

    /*
    Fungsi index
    Bertujuan untuk load data ketika controller pertama kali dipanggil
    */
    public function index(){
        $this->load->view('admin/header');
        $this->load->view('admin/vhome');
        $this->load->view('footer');
    }

    public function logout(){
        $this->session->sess_destroy();
        redirect('Login');
    }
}
?>

```

## D12. Penulisan Kode Program class controllers/Login

Tabel D 12 Penulisan Kode Program class controllers/Login

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

```

```
class Login extends CI_Controller {

    /*
    * KONSTRUKTOR
    * Bertujuan untuk memuat model LOGIN
    */
    public function __construct(){
        parent::__construct();
        $this->load->model('Mlogin');
    }
    public function index() {

        $this->load->view('vlogin');
    }

    public function proses() {
        if(isset($_POST)){
            $u=$this->input->post('username');
            $p=$this->input->post('password');
            if($this->Mlogin->md_login($u,$p) == TRUE){
                $level =$this->session->userdata('level');
                $NamaUser
                =$this->session->userdata('NamaUser');
                if($level == 'admin'){
                    redirect(site_url() . '/dashboard');
                } elseif ($level == 'guru' ) {
                    redirect(site_url() . '/c_manager/dashboard');
                } elseif ($level == 'kepsek' ) {
                    redirect(site_url() . '/c_manager/dashboard');
                } else { // incorrect username or password
                    echo '<script>alert("Username atau Password yang Anda Masukkan
                    Salah!")</script>';
                    redirect(base_url(), 'refresh');
                }
            } else {

                $this->logout();
            }
        }
    }

    public function logout() {
        $this->session->sess_destroy(); //untuk mendestroy session login
    }
}
```

```
        echo '<script>alert("Username dan Password Salah")</script>'; //untuk
menampilkan alert
        redirect(base_url(), 'refresh'); //untuk mengembalikan ke tampilan login
awal
    }
}
```

### D13. Penulisan Kode Program class models/M\_ahp

Tabel D 13 Penulisan Kode Program class models/M\_ahp

```
<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

Class M_ahp extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    public function get_namakriteria(){
        $this->db->from('tb_kriteria');
        return $this->db->get();
    }

    public function getKriteria(){
        return $query = $this->db->query(" SELECT * from tb_kriteria") -
>result_array();
    }

    public function md_getsubkriteria_id($id_kriteria) {
        $query = $this->db->query("SELECT *FROM tb_kriteria a join
tb_subkriteria b on a.id_kriteria=b.id_kriteria WHERE
a.id_kriteria='$id_kriteria'");
        return $query;
    }

    public function getHapusPegawai($id_user) {
        $query = $this->db->query("UPDATE `capres`
        SET `status_user` ='1'
        WHERE `id_user` ='$id_user'"
    }
}
```

```
);
return $query;
}

public function getDetailPegawai($id_user) {
return $query = $this->db->query("SELECT *
FROM capres
where `id_user`=' $id_user'")->result_array();

}

public function getEditPegawai($input,$id_user) {
return $query = $this->db->query("UPDATE capres
SET nip =' $input[nip]', namaPeg = '$input[namaPeg]', jenisKel
= '$input[jenisKel]', tanggalL= '$input[tanggalL]', tempatTL
= '$input[tempatTL]', alamatS = '$input[alamatS]', nope = '$input[nope]',
statusHub = '$input[statusHub]', jumKel = '$input[jumKel]', alumniPen =
'$input[alumniPen]', gelar = '$input[gelar]', jabatan = '$input[jabatan]',
tanggalMKerja = '$input[tanggalMKerja]'
WHERE id_user= $id_user");

}

//===== data KRITERIA=====
function md_getkriteria() {
$this->db->select('*');
$this->db->from('tb_kriteria');
return $this->db->get();
}

//=====PEMBOBOTAN KRITERIA=====
function md_relasi_kriteria(){
$this->db->select('*');
$this->db->from('tb_relasi_kriteria');
return $this->db->get();
}

function md_update_bobot_relasiKriteria($id_kriteria_dari,
$Sid_kriteria_tujuan, $data){
$this->db->where('id_kriteria_dari', $id_kriteria_dari);
$this->db->where('id_kriteria_tujuan', $id_kriteria_tujuan);
$this->db->update('tb_relasi_kriteria', $data);
}
}
```

```
function md_update_bobotprioritas($vector, $i){
    $query = $this->db->query("UPDATE `tb_kriteria`
        SET `bobot_prioritas` = '$vector'
        where id_kriteria = '$i'
        ");
    return $query;
}
?>
```

#### D14. Penulisan Kode Program class models/M\_datapegawai

Tabel D 14 Penulisan Kode Program class models/M\_datapgawai

```
<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

Class M_datapegawai extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    function get_kualifikasi(){
        $this->db->select('*');
        $this->db->from('tb_subkriteria');
        $this->db->where('id_kriteria', 1);
        return $this->db->get();
    }
    function get_kesehatan(){
        $this->db->select('*');
        $this->db->from('tb_subkriteria');
        $this->db->where('id_kriteria', 3);
        return $this->db->get();
    }
    function get_catathukum(){
        $this->db->select('*');
        $this->db->from('tb_subkriteria');
        $this->db->where('id_kriteria', 4);
```



```
return $this->db->get();
}
function get_sertifikat(){
    $this->db->select('*');
    $this->db->from('tb_subkriteria');
    $this->db->where('id_kriteria', 5);
    return $this->db->get();
}
function get_golongan(){
    $this->db->select('*');
    $this->db->from('tb_subkriteria');
    $this->db->where('id_kriteria', 7);
    return $this->db->get();
}
function get_DP3(){
    $this->db->select('*');
    $this->db->from('tb_subkriteria');
    $this->db->where('id_kriteria', 8);
    return $this->db->get();
}
function get_tugastambahan(){
    $this->db->select('*');
    $this->db->from('tb_subkriteria');
    $this->db->where('id_kriteria', 9);
    return $this->db->get();
}

public function getDataPegawai(){
    return $query = $this->db->query(" SELECT * from capres join login on
capres.id_user = login.id_user where capres.id_user != 0") ->result_array();
}
public function getNip($banding){
    return $query = $this->db->query(" SELECT * from capres where
nip='$banding' ") ->result_array();
}

public function getCari($key){
    return $query = $this->db->query(" SELECT * from capres where (nip LIKE
'%"$key%" OR namaPeg LIKE '%"$key%') AND (id_user != '0') ") ->result_array();
}
public function getCaritahun($id_user,$tahunajaran){
```

```

    return $query = $this->db->query(" SELECT * from tb_rekapnilai where
(tahunajaran LIKE '%$tahunajaran%' ) AND (id_user != '0') ") ->result_array();
}
public function masuk(){
    return $query = $this->db->query(" SELECT id_user from capres order by
id_user DESC LIMIT 1") ->result_array();
}

public function ubah(){
    return $query = $this->db->query("UPDATE tb_nilai_pm, tb_rekapnilai SET
tb_nilai_pm.nilai = tb_rekapnilai.nilai, status = '1' where (tb_nilai_pm.id_user =
tb_rekapnilai.id_user) and (tb_nilai_pm.id_kriteria_pm =
tb_rekapnilai.id_kriteria_pm)");
}

//public function ubah(){
//    return $query = $this->db->query("UPDATE tb_nilai_pm
//SET tb_nilai_pm.nilai = (
//SELECT nilai FROM tb_rekapnilai
//WHERE tb_nilai_pm.id_user = tb_rekapnilai.id_user
//)
//WHERE tb_nilai_pm.id_user in (SELECT id_user FROM tb_rekapnilai)");}

function
insert($id_user,$nip,$namaPeg,$jenisKel,$tanggalL,$tempatTL,$alamatS,$nope,$t
anggalMKerja,$level){
    $insert=$this->db->query("INSERT INTO login
(id_user,Username,id_level>Password,status)
values
('".$id_user."','".$nip."','".$level."','".$1234','1')");
    $insert=$this->db->query("INSERT INTO capres
(id_user,nip,namaPeg,jenisKel,tanggalL,tempatTL,alamatS,nope,tanggalMKerja,st
atus_user)
values
('".$id_user."','".$nip."','".$namaPeg."','".$jenisKel."','".$tanggalL."','".$tempatTL.'"
, '".$alamatS."','".$nope."','".$tanggalMKerja."','".$1')");

    // $insert=$this->db->query("INSERT INTO tb_nilai_pm
(id_kriteria_pm,id_user,nilai,hasil,tahun_ajaran)
values
('".$id_kriteria_pm."','".$id_user."','".$nilai."','".$hasil."','".$tahunajaran.'"");
    // $insert=$this->db->query("INSERT INTO verifikasi (id_user,Username)");
}

```

```
function
updateuser($sid_user,$sid_level,$nip,$namaPeg,$jenisKel,$tanggalL,$tempatTL,$sala
matS,$nope,$tanggalMKerja){
    $updateuser=$this->db->query("UPDATE capres set
        nip='$nip',
        namaPeg='$namaPeg',
        jenisKel='$jenisKel',
        tanggalL='$tanggalL',
        tempatTL='$tempatTL',
        alamatS='$alamatS',
        nope='$nope',
        tanggalMKerja='$tanggalMKerja'
        where id_user='$sid_user'");
    $updateuser=$this->db->query("UPDATE login set
        id_level='$sid_level'
        where id_user='$sid_user'");
}

public function EditStatus() {
return $query = $this->db->query("UPDATE verifikasi
SET status ='0'
WHERE status='1'");
}

function insertnilai($id_kriteria_pm,$sid_user,$nilai,$hasil,$tahunajaran){
    $insert=$this->db->query("INSERT INTO tb_nilai_pm
(id_kriteria_pm,id_user,nilai,hasil,tahun_ajaran) values
('.$id_kriteria_pm.', '$sid_user.', '$nilai.', '$hasil.', '$tahunajaran.'");

    $update=$this->db->query("UPDATE tb_nilai_pm set status='0' where
status='1'");
}

function insertnilairekap($id_kriteria_pm,$sid_user,$nilai,$tahunajaran,$hasil){
    //$insert=$this->db->query("INSERT INTO tb_nilai_pm
(id_kriteria_pm,id_user,nilai,hasil) values
('.$id_kriteria_pm.', '$sid_user.', '$nilai.', '$hasil.'");
    $insert=$this->db->query("INSERT INTO tb_nilai_pm
(id_kriteria_pm,id_user,nilai,hasil,tahun_ajaran) values
('.$id_kriteria_pm.', '$sid_user.', '$nilai.', '$hasil.', '$tahunajaran.'");
}
```

```

}

function updatenilaipm($id_kriteria_pm, $id_user, $nilai,$tahunajaran){
    $update=$this->db->query("UPDATE  tb_nilai_pm  set      nilai='$nilai'  ,
tahun_ajaran='$tahunajaran'
        where id_kriteria_pm='$id_kriteria_pm' AND
                id_user='$id_user'");
    $update=$this->db->query("UPDATE  tb_nilai_pm  set      status='0'  where
status='1'");
}

function updatenilai($id_kriteria_pm, $id_user, $nilai,$tahunajaran){
    $update=$this->db->query("UPDATE tb_nilai_pm set  nilai='$nilai',
tahun_ajaran='$tahunajaran'
        where id_kriteria_pm='$id_kriteria_pm' AND
                id_user='$id_user'");
    }

//===== DATA CAPRES
=====//

function get_datacapres(){
    $this->db->from("capres");
    return $this->db->get();
}

function get_level(){
    $this->db->from("tb_level");
    return $this->db->get();
}

function delete($no_id){
    $delete=$this->db->query("DELETE FROM capres WHERE no_id
=".$no_id."");
}

function get_ranking(){
    $this->db->from("capres");
    $this->db->order_by("nilai",'desc');
    return $this->db->get();
}

public function getHapusPegawai($id_user) {

```

```
$query = $this->db->query("UPDATE `capres`  
    SET `status_user` = '0'  
    WHERE `id_user` = '$id_user'"  
);  
$query = $this->db->query("UPDATE `login`  
    SET `status` = '0'  
    WHERE `id_user` = '$id_user'"  
);  
return $query;  
}  
public function getAktvPegawai($id_user) {  
    $query = $this->db->query("UPDATE `capres`  
        SET `status_user` = '1'  
        WHERE `id_user` = '$id_user'"  
    );  
    $query = $this->db->query("UPDATE `login`  
        SET `status` = '1'  
        WHERE `id_user` = '$id_user'"  
    );  
    return $query;  
}  
public function reset_pegawai($id_user) {  
    $query = $this->db->query("UPDATE `login`  
        SET `Password` = '1234'  
        WHERE `id_user` = '$id_user'"  
    );  
    return $query;  
}  
public function getDetailPegawai($id_user) {  
    return $query = $this->db->query("SELECT *  
        FROM capres join tb_nilai_pm on capres.id_user =  
        tb_nilai_pm.id_user join login on capres.id_user = login.id_user join tb_level on  
        tb_level.id_level = login.id_level  
        where capres.id_user=' $id_user' order by  
        tb_nilai_pm.id_kriteria_pm ASC")->result_array();  
}  
public function ceknilaipm($tahunajaran,$id_user) {  
    return $query = $this->db->query("SELECT *  
        FROM tb_nilai_pm where id_user=' $id_user' AND  
        tahun_ajaran = '$tahunajaran'")->result_array();
```

```

    }
    public function cekrekap($tahunajaran) {
        return $query = $this->db->query("SELECT * FROM tb_rekapnilai where
tahunajaran = '$tahunajaran'")->result_array();
    }
    public function getDetailSub($ambil,$i) {
        return $query = $this->db->query("SELECT nama_subkriteria
        FROM tb_subkriteria where bobot_prioritas_sub='$ambil'
and id_kriteria_pm='$i'")
        ->result_array();
    }

    public function getEditPegawai($input,$id_user) {
        return $query = $this->db->query("UPDATE capres
        SET nip = '$input[nip]', namaPeg = '$input[namaPeg]', jenisKel
        = '$input[jenisKel]', tanggalL= '$input[tanggalL]', tempatTL = '$input[tempatTL]',
alamatS = '$input[alamatS]', nope = '$input[nope]', tanggalMKerja =
        '$input[tanggalMKerja]'
        WHERE id_user=$id_user");
    }
}
}
?>

```

#### D15. Penulisan Kode Program class models/M\_hasil

Tabel D 15 Penulisan Kode Program class models/M\_hasil

```

<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

Class M_hasil extends CI_Model {

    function __construct() {
        parent::__construct();
    }
}

```

```
//function insert_next($id_user,$nilai){
//          $insert=$this->db->query("INSERT INTO tb_hasil_pm
(id,id_user,nilai) values ('null','.$id_user.','.$nilai.'");
//}
public function gettahun(){
return $query = $this->db->query(" SELECT tahun_ajaran from tb_nilai_pm order
by tahun_ajaran DESC LIMIT 1") ->result_array();
}
function update_nilai($id_user, $nilai){
    $data = array(
        'hasil' => $nilai);
    $this->db->where('id_user', $id_user);
    $this->db->update('capres', $data);
}
public function cek_kepala_sekolah_baru(){
return $query = $this->db->query("SELECT id_user from capres order by
hasil DESC LIMIT 1") ->result_array();
}
public function cekdatarekap($tahunajaran,$id_user){
return $query = $this->db->query("SELECT * from tb_rekapnilai where
tahunajaran='$tahunajaran' AND id_user='$id_user'") ->result_array();
}
public function ambil_data_rekap(){
return $query = $this->db->query("SELECT * from tb_nilai_pm WHERE
tahun_ajaran = (
SELECT max( tahun_ajaran ) FROM tb_nilai_pm )") ->result_array();
}
public function ambil_data_capres($id_user){
return $query = $this->db->query("SELECT * from capres where
id_user='$id_user'") ->result_array();
}

public function getNilaiPM(){
return $query = $this->db->query("SELECT * from tb_nilai_pm WHERE
tahun_ajaran = (
SELECT max( tahun_ajaran ) FROM tb_nilai_pm ) order by id_user,id_kriteria_pm
ASC ") ->result_array();
}
public function jumlahcapres(){
return $query = $this->db->query(" SELECT * from tb_nilai_pm WHERE
tahun_ajaran = (
```

```
SELECT max( tahun_ajaran ) FROM tb_nilai_pm ) group by id_user ") -
>result_array();
}
public function gethasil(){
    return $query = $this->db->query(" SELECT capres.namaPeg,capres.hasil
from capres join tb_nilai_pm on capres.id_user = tb_nilai_pm.id_user
    where capres.status_user ='1' and tb_nilai_pm.tahun_ajaran = (
SELECT max( tb_nilai_pm.tahun_ajaran ) FROM tb_nilai_pm ) group by
capres.hasil desc, capres.id_user desc ") ->result_array();
}
public function getCapres() {
    return $query = $this->db->query("SELECT *
        FROM capres join tb_nilai_pm on capres.id_user =
tb_nilai_pm.id_user WHERE tahun_ajaran = (
SELECT max( tahun_ajaran ) FROM tb_nilai_pm ) group by tb_nilai_pm.id_user
")->result_array(); } //biar urut id usernya
public function getKriteria(){
    return $query = $this->db->query(" SELECT * from tb_kriteria") -
>result_array();
}

public function getIdeal(){
    return $query = $this->db->query(" SELECT * from tb_kriteria_pm") -
>result_array();
}
public function getHapusPegawai($id_user) {
    $query = $this->db->query("UPDATE `capres`
        SET `status_user` ='1'
        WHERE `id_user` ='$id_user'"
    );
    return $query;
}

public function change_head($id_ganti) {
    $q = $this->db->query("UPDATE login
        SET id_level ='2'
        where id_level='3'"
    );
    $query = $this->db->query("UPDATE login
        SET id_level ='3'
        where id_user='$id_ganti'"
    );
};
```



```

        return $query;
    }
    public function getDetailPegawai($id_user) {
        return $query = $this->db->query("SELECT *
            FROM capres
            where `id_user`=' $id_user'")->result_array();
    }
    public function getreset() {
        return $query = $this->db->query("SELECT *
            FROM tb_nilai_pm LIMIT 1")->result_array();
    }
    public function getEditNilai($input,$id_user) {
        return $query = $this->db->query("UPDATE nilai_pm
            SET nilai =' $input[nilai]' WHERE no_id=$id_user");
    }
}
function
insertnilairekap($id_kriteria_pm,$id_user,$nilai,$tahunajaran,$hasil,$nilai_akhir){
    $insert=$this->db->query("INSERT INTO tb_rekapnilai
(id_kriteria_pm,id_user,nilai,tahunajaran,hasil,nilai_akhir)
values
('.$id_kriteria_pm.','.$id_user.','.$nilai.','.$tahunajaran.','.$hasil.','.$nilai_
akhir.'"));
    //$insert = $this->db->query("DELETE from tb_nilai_pm where
id_kriteria_pm=' $id_kriteria_pm' AND id_user=' $id_user' AND
tahun_ajaran=' $tahunajaran'");
}
}
?>

```

#### D16. Penulisan Kode Program class models/M\_manager

Tabel D 16 Penulisan Kode Program class models/M\_manager

```

<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

```

```
Class M_manager extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    public function getrekapdetailnilai($id_user){
        return $query = $this->db->query(" SELECT * from tb_nilai_pm join
        capres on tb_nilai_pm.id_user = capres.id_user
        where tb_nilai_pm.id_user='$id_user' AND tb_nilai_pm.status='1'") -
        >result_array();
    }

    public function EditStatus() {
        //return $query = $this->db->query("UPDATE verifikasi SET status = '1'
        WHERE status='0'");
        return $query = $this->db->query("UPDATE tb_nilai_pm SET status = '1'
        WHERE status='0'");
        //$insert=$this->db->query("INSERT INTO verifikasi (id,status) values
        ('null','1')");
    }
    public function getkriteria() {
        return $query = $this->db->query("SELECT nama_kriteria from tb_kriteria")-
        >result_array();
    }
    public function gettahun(){
        return $query = $this->db->query(" SELECT tahun_ajaran from tb_nilai_pm
        order by tahun_ajaran DESC LIMIT 1") ->result_array();
    }
    public function getrekap(){
        $id=$this->session->userdata('id');
        return $query = $this->db->query(" SELECT * from tb_rekapnilai where
        id_user='$id' group by tahunajaran ") ->result_array();
    }
    public function getsubkriteria($id_kriteria,$nilai_kriteria){
        return $query = $this->db->query("SELECT nama_subkriteria from
        tb_subkriteria where id_kriteria_pm = '$id_kriteria' and bobot_prioritas_sub=
        '$nilai_kriteria'")->result_array();
    }

    public function getrekapdetailnilai2($id_user,$tahunajaran){
```

```

        return $query = $this->db->query(" SELECT * from tb_nilai_pm join
        capres on tb_nilai_pm.id_user = capres.id_user
        where tb_nilai_pm.id_user='$id_user' AND
        tb_nilai_pm.tahun_ajaran='$tahunajaran'" )->result_array();
    }
    public function getNilaiPM(){
        $id=$this->session->userdata('id');
        return $query = $this->db->query("SELECT * from tb_nilai_pm join
        tb_kriteria_pm
        on tb_nilai_pm.id_kriteria_pm = tb_kriteria_pm.id_kriteria_pm where
        tb_nilai_pm.id_user='$id'" )->result_array();
    }
    public function gethasil(){
        return $query = $this->db->query(" SELECT
        capres.id_user,capres.nip,capres.namaPeg,capres.jenisKel,capres.tanggalL,
        capres.alamatS,capres.nope,capres.tanggalMKerja,capres.status_user,
        capres.hasil,tb_nilai_pm.id_kriteria_pm,tb_nilai_pm.id_user,tb_nilai_pm.nilai,
        tb_nilai_pm.tahun_ajaran,tb_nilai_pm.status
        from capres right join tb_nilai_pm on capres.id_user = tb_nilai_pm.id_user
        where capres.status_user ='1' and tb_nilai_pm.tahun_ajaran = (
        SELECT max( tb_nilai_pm.tahun_ajaran ) FROM tb_nilai_pm ) group by
        capres.hasil desc, capres.id_user desc ") ->result_array();
    }
    public function getver(){
        return $query = $this->db->query(" SELECT status from tb_nilai_pm
        order by id_user DESC LIMIT 1") ->result_array();
    }
    public function getlogin($id_user){
        return $query = $this->db->query(" SELECT * from login join tb_level
        on login.id_level = tb_level.id_level where login.id_user = '$id_user' ") -
        >result_array();
    }
}
?>

```

#### D17. Penulisan Kode Program class models/M\_pm

Tabel D 17 Penulisan Kode Program class models/M\_pm

```
<?php
```

```
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

Class M_pm extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    //function insert_next($id_user,$nilai){
        //      $insert=$this->db->query("INSERT INTO tb_hasil_pm
(id,id_user,nilai) values ('null','" . $id_user . "','" . $nilai . "')");
    //}
    function update_nilai_pm($id_kriteria, $user, $hasil2){
        $data = array(
            'hasil' => $hasil2);
        $this->db->where('id_kriteria_pm', $id_kriteria);
        $this->db->where('id_user', $user);
        $this->db->update('tb_nilai_pm', $data);
    }
    //function delete_next(){
        //      $delete=$this->db->query("DELETE from tb_hasil_pm where
nilai !='");
    //}

    public function getNilaiPM(){
        return $query = $this->db->query("SELECT * from tb_nilai_pm
WHERE tahun_ajaran = (
SELECT max( tahun_ajaran ) FROM tb_nilai_pm ) order by
id_user,id_kriteria_pm ASC ") ->result_array();
    }
    public function jumlahcapres(){
        return $query = $this->db->query(" SELECT * from tb_nilai_pm
WHERE tahun_ajaran = (
SELECT max( tahun_ajaran ) FROM tb_nilai_pm ) group by id_user ") -
>result_array();
    }
    public function getCapres() {
        return $query = $this->db->query("SELECT *
FROM capres join tb_nilai_pm on capres.id_user =
tb_nilai_pm.id_user WHERE tahun_ajaran = (
```

```
SELECT max( tahun_ajaran ) FROM tb_nilai_pm ) group by
tb_nilai_pm.id_user ")->result_array(); } //biar urut id usernya
public function getKriteria(){
    return $query = $this->db->query(" SELECT * from tb_kriteria_pm") -
->result_array();
}

public function getIdeal(){
    return $query = $this->db->query(" SELECT * from tb_kriteria_pm") -
->result_array();
}

public function gettahun(){
    return $query = $this->db->query(" SELECT tahun_ajaran from
tb_nilai_pm order by tahun_ajaran DESC LIMIT 1") ->result_array();
}

public function getHapusPegawai($id_user) {
    $query = $this->db->query("UPDATE `capres`
        SET `status_user` = '1'
        WHERE `id_user` = '$id_user'"
    );
    return $query;
}

public function getDetailPegawai($id_user) {
    return $query = $this->db->query("SELECT *
        FROM capres
        where `id_user` = '$id_user'")->result_array();
}

public function getnilaiideal(){
    return $query = $this->db->query(" SELECT id_kriteria_pm,
nama_kriteria_pm, ideal from tb_kriteria_pm ") ->result_array();
}

public function getEditNilai($input,$id_user) {
    return $query = $this->db->query("UPDATE nilai_pm
        SET nilai = '$input[nilai]' WHERE no_id=$id_user");
}

public function getDetailIdeal($id_kriteria_pm) {
```

```

return $query = $this->db->query("SELECT *
                                FROM tb_kriteria_pm
                                where `id_kriteria_pm`='$_id_kriteria_pm'");
>result_array();

}

public function getEditIdeal($input,$id_kriteria_pm) {
return $query = $this->db->query("UPDATE tb_kriteria_pm
SET ideal ='$input[ideal]'
WHERE id_kriteria_pm=$_id_kriteria_pm");
}
}
?>

```

#### D18. Penulisan Kode Program class models/M\_rekap

Tabel D 18 Penulisan Kode Program class models/M\_rekap

```

<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

Class M_rekap extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    public function getrekap(){
        return $query = $this->db->query(" SELECT * from tb_rekapnilai
        group by tahunajaran") ->result_array();
    }

    public function getrekapdetail($tahunajaran){
        return $query = $this->db->query(" SELECT * from tb_rekapnilai join
        capres on tb_rekapnilai.id_user = capres.id_user
        where tb_rekapnilai.tahunajaran='$_tahunajaran' group by tb_rekapnilai.id_user
        order by tb_rekapnilai.nilai_akhir DESC") ->result_array();
    }
}

```

```
    }  
    public function getsubkriteria($id_kriteria,$nilai_kriteria){  
        return $query = $this->db->query("SELECT nama_subkriteria from  
tb_subkriteria where id_kriteria_pm = '$id_kriteria' and bobot_prioritas_sub=  
'$nilai_kriteria'")->result_array();  
    }  
    public function getrekapdetailnilai($id_user,$tahunajaran){  
        return $query = $this->db->query(" SELECT * from tb_rekapnilai join  
capres on tb_rekapnilai.id_user = capres.id_user  
where tb_rekapnilai.id_user='$id_user' AND  
tb_rekapnilai.tahunajaran='$tahunajaran'") ->result_array();  
    }  
    public function getDetailPegawai($id_user,$tahunajaran,$tahunnnext) {  
        return $query = $this->db->query("SELECT *  
FROM capres join tb_rekapnilai on capres.id_user =  
tb_rekapnilai.id_user join login on capres.id_user = login.id_user join tb_level  
on tb_level.id_level = login.id_level  
where capres.id_user='$id_user' and  
tb_rekapnilai.tahunajaran = '$tahunajaran/$tahunnnext' order by  
tb_rekapnilai.id_kriteria_pm ASC")->result_array();  
    }  
  
    public function getDetailSub($ambil,$i) {  
        return $query = $this->db->query("SELECT nama_subkriteria  
FROM tb_subkriteria where  
bobot_prioritas_sub='$ambil' and id_kriteria_pm='$i'")->result_array();  
    }  
  
    public function getCari($key){  
        return $query = $this->db->query(" SELECT * from tb_rekapnilai  
where tahunajaran LIKE '%$key%' group by tahunajaran ") ->result_array();  
    }  
  
}
```

Tabel D 19 Penulisan Kode Program class models/Mdatauser

```
<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

Class Mdatauser extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    public function getDataDiri(){
        $id=$this->session->userdata('id');
        return $query = $this->db->query("SELECT * from capres join login
        on capres.id_user = login.id_user where capres.id_user='$id'" ) -
        >result_array();
    }

    public function getEditGuru($input,$id_user) {
        return $query = $this->db->query("UPDATE capres
        SET nip ='$input[nip]', namaPeg ='$input[namaPeg]', jenisKel
        ='$input[jenisKel]',
        tanggalL='$input[tanggalL]', tempatTL ='$input[tempatTL]', alamatS
        ='$input[alamatS]',
        nope = '$input[nope]'
        WHERE id_user=$id_user");
    }

    public function getEditGuruPass($id_user,$passbaru) {
        return $query = $this->db->query("UPDATE login
        SET Password ='$passbaru'
        WHERE id_user=$id_user");
    }

    public function getNip($banding){
        return $query = $this->db->query(" SELECT * from capres where
        nip='$banding' ") ->result_array();
    }

}

?>
```



## D20. Penulisan Kode Program class models/Mlogin

Tabel D 20 Penulisan Kode Program class models/Mlogin

```
<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class Mlogin extends CI_Model {
function __construct() {
    parent::__construct();
}

    function md_login($u, $p) {
$query = $this->db->query
("SELECT *
FROM `login` a join `tb_level` b on (a.id_level = b.id_level)
join `capres` c on (a.id_user=c.id_user)
WHERE a.`username`='\" . $u . \"'
and a.`password`='\" . $p . \"'
and a.`status`='1'
and a.`id_level`= b.`id_level`");
if ($query->num_rows() > 0) {//tanya
foreach ($query->result_array() as $row) {
    $this->session->set_userdata('user_on', TRUE);
    $this->session->set_userdata('id', $row['id_user']);
    $this->session->set_userdata ('username', $row['username']);
    $this->session->set_userdata ('namaPeg', $row['namaPeg']);
    $this->session->set_userdata('password', $row['password']);
    $this->session->set_userdata('id_level', $row['id_level']);
    $this->session->set_userdata('level', $row['level']);
    if ($row['id_level'] == 1) {
        $this->session->set_userdata('admin', TRUE);
    } elseif($row['id_level'] == 2){
        $this->session->set_userdata('guru', TRUE);
    } elseif($row['id_level'] == 3){
        $this->session->set_userdata('kepsek', TRUE);
    } else{
        $this->session->set_userdata('karyawan', TRUE);
    }
}
}
return TRUE;
}
```

```
} else {  
    return FALSE;  
}  
}  
}
```

## D21. Penulisan Kode Program class models/Mprofile

Tabel D 21 Penulisan Kode Program class models/Mprofile

```
<?php  
class Mprofile extends CI_Model{  
  
    function get_namaPeg($ambil){  
        $this->db->select('*');  
        $this->db->from('capres');  
        $this->db->where('status_user', 1);  
        $this->db->where('nip', $ambil);  
  
        return $this->db->get();  
    }  
    function get_namaPeg3(){  
        $this->db->select('*');  
        $this->db->from('capres');  
        $this->db->where('status_user', 1);  
  
        return $this->db->get();  
    }  
    public function get_namaPeg2($id_user) {  
        return $query = $this->db->query("SELECT *  
            FROM capres where id_user = '$id_user'")->result_array(); }  
    //biar urut id usernya  
  
    function get_kualifikasi(){  
        $this->db->select('*');  
        $this->db->from('tb_subkriteria');  
        $this->db->where('id_kriteria', 1);  
        return $this->db->get();  
    }  
}
```

```
function get_kualifikasi2(){
    $this->db->select('*');
    $this->db->from('tb_subkriteria');
    $this->db->where('id_kriteria', 1);
    return $this->db->get();
}
public function getCari($key){
    return $query = $this->db->query(" SELECT * from capres
where (nip LIKE '%$key%' OR namaPeg LIKE '%$key%') AND (id_user !=
'0') ") ->result_array();
}

function get_usia($ambil){
    $this->db->select('*');
    $this->db->from('capres');
    $this->db->where('nip', $ambil);
return $this->db->get();
}
function get_usia2(){
    $this->db->select('*');
    $this->db->from('capres');
    return $this->db->get();
}
function get_kesehatan(){
    $this->db->select('*');
    $this->db->from('tb_subkriteria');
    $this->db->where('id_kriteria', 3);
return $this->db->get();
}
function get_catathukum(){
    $this->db->select('*');
    $this->db->from('tb_subkriteria');
    $this->db->where('id_kriteria', 4);
return $this->db->get();
}
function get_sertifikat(){
    $this->db->select('*');
    $this->db->from('tb_subkriteria');
    $this->db->where('id_kriteria', 5);
return $this->db->get();
}
function get_pengalaman($ambil){
```

```
        $this->db->select('*');
    $this->db->from('capres');
        $this->db->where('nip', $ambil);
    return $this->db->get();
    }
    function get_golongan(){
        $this->db->select('*');
    $this->db->from('tb_subkriteria');
    $this->db->where('id_kriteria', 7);
    return $this->db->get();
    }
    function get_DP3(){
        $this->db->select('*');
    $this->db->from('tb_subkriteria');
    $this->db->where('id_kriteria', 8);
    return $this->db->get();
    }
    function get_tugastambahan(){
        $this->db->select('*');
    $this->db->from('tb_subkriteria');
    $this->db->where('id_kriteria', 9);
    return $this->db->get();
    }

function insert($id_kriteria_pm,$id_user,$nilai,$hasil){
    $insert=$this->db->query("INSERT INTO tb_nilai_pm
(id_kriteria_pm,id_user,nilai,hasil) values
('".$id_kriteria_pm."','".$id_user."','".$nilai."','".$hasil."");
}
function updatenilai($id_kriteria_pm,$id_user,$nilai,$hasil){
$query=$this->db->query("UPDATE tb_nilai_pm set
nilai='$nilai',hasil='$hasil' where id_user='$id_user' and
id_kriteria_pm='$id_kriteria_pm'");
}
function update($id_kriteria_pm,$id_user,$nilai,$hasil){
    $query=$this->db->query("UPDATE tb_nilai_pm set
nilai='$nilai'where id_user='$id_user'");
    return $query;
}
function md_update_nilai_pm($id_kriteria_pm, $id_user, $nilai){
```

```
$this->db->where('id_kriteria_pm', $id_kriteria_pm);
$this->db->where('id_user', $id_user);
$this->db->update('tb_nilai_pm', $nilai);
}
// function updatepm($id_kriteria_pm, $id_user, $nilai){
//     $this->db->where('id_user', $id_user);
//     $this->db->where('id_kriteria_pm', $id_kriteria_pm);
//     $this->db->update('nilai', $nilai);
// }

// function updatepm($id_kriteria_pm, $id_user,$nilai){
//     $query = $this->db->query("UPDATE `tb_nilai_pm` SET `nilai` ='$nilai'
where id_kriteria_pm = '$id_kriteria_pm' , id_user = '$id_user'"); }

function get_dataapres(){
    $this->db->from("apres");
    return $this->db->get();
}

function get_nilaipm(){
    $this->db->from("tb_nilai_pm");
    return $this->db->get();
}

public function get_cek($id_user){
    return $query = $this->db->query(" SELECT * from tb_nilai_pm where
id_user='$id_user' ") ->result_array();
}

public function get_cek2($id_user){
    return $query = $this->db->query(" SELECT * from
tb_nilai_pm join tb_subkriteria on tb_nilai_pm.id_kriteria_pm =
tb_subkriteria.id_kriteria_pm and
tb_nilai_pm.nilai = tb_subkriteria.bobot_prioritas_sub
where id_user='$id_user' ") ->result_array();
}

//public function get_cek2($id_user){
//     return $query = $this->db->query(" SELECT * from
tb_nilai_pm join tb_subkriteria on tb_nilai_pm.id_kriteria_pm =
tb_subkriteria.id_kriteria_pm and
//         tb_nilai_pm.nilai =
tb_subkriteria.bobot_prioritas_sub
//         where id_user='$id_user' ") ->result_array();
//}
```

```
public function getKrit(){
    return $query = $this->db->query(" SELECT * from tb_kriteria_pm") -
>result_array();
}
function delete($id_user){
    $delete=$this->db->query("DELETE FROM capres
WHERE id_user = ".$_user."");
}

function get_ranking(){
    $this->db->from("capres");
    $this->db->order_by("nilai",'desc');
    return $this->db->get();
}

public function delete2($id_user){
    $delete = $this->db->query("DELETE from tb_nilai_pm where
id_user='$id_user'");
}

public function getCapres() {
    return $query = $this->db->query("SELECT *
FROM capres join tb_nilai_pm on capres.id_user =
tb_nilai_pm.id_user group by tb_nilai_pm.id_user ")>result_array(); } //biar
urut id usernya
public function getNilaiPM(){
    return $query = $this->db->query("SELECT * from tb_nilai_pm
order by id_user,id_kriteria_pm ASC") ->result_array();
}
public function getCapres2($key) {
    return $query = $this->db->query("SELECT *
FROM capres join tb_nilai_pm on capres.id_user =
tb_nilai_pm.id_user where (capres.nip LIKE '%$key%' OR capres.namaPeg
LIKE '%$key%') AND (capres.id_user != '0') group by tb_nilai_pm.id_user
")>result_array(); } //biar uru
}
?>
```

## Lampiran E. Pengujian Black Box

| No. | Fitur                    | Aksi   | Hasil  | Kesimpulan                |
|-----|--------------------------|--|--|---------------------------|
| 1.  | <i>Login</i>             | Inputkan username dan password dan menekan tombol Login      | <i>Login</i> berhasil dan menampilkan halaman awal masing-masing aktor level   | [√] Berhasil<br>[ ] Gagal |
|     |                          | Kolom <i>username</i> atau <i>password</i> salah             | Menampilkan pesan “ <i>username</i> dan <i>password</i> salah“   | [√] Berhasil<br>[ ] Gagal |
|     |                          | Salah satu kolom <i>username</i> atau <i>password</i> kosong | Menampilkan pesan “ <i>Isi isian ini</i> ” dengan menunjuk pada kolom yang belum diisi   | [√] Berhasil<br>[ ] Gagal |
| 2.  | <i>View Data Pegawai</i> | Klik menu data guru  | Menampilkan halaman data daftar guru dan tabel berisi no, nip, nama pegawai, status, dan detail  | [√] Berhasil<br>[ ] Gagal |
|     |                          | Klik tombol detail pada salah satu baris dikolom             | Menampilkan halaman detail data guru berupa level, NIP, nama pegawai, jenis kelamin, tanggal lahir, alamat kelahiran, alamat tinggal sekarang, nomer telepon rumah / nomer ponsel, dan tanggal mulai kerja, tahun ajaran, nilai kualifikasi, nilai kesehatan, nilai sertifikat, nilai golongan, nilai usia, nilai catatan hukum, nilai pengalaman, nilai DP3, dan nilai tambahan | [√] Berhasil<br>[ ] Gagal |

|    |                        |                           |   |                           |
|----|------------------------|---------------------------|---|---------------------------|
| 3. | <i>Input Data Guru</i> | Klik menu Data Guru       | Menampilkan halaman data daftar guru dan tabel berisi no, nip, nama pegawai, status, detail   | [√] Berhasil<br>[ ] Gagal |
|    |                        | Klik tombol “Tambah Baru” | Menampilkan halaman form tambah data guru yang terdiri dari level (berupa combo box yang berisi pilihan Kepala Sekolah dan Guru), tahun ajaran (berupa combo box yang berisi pilihan tahun ajaran), nip, nama pegawai, jenis kelamin (berupa combo box yang berisi pilihan pria dan wanita), tanggal lahir, alamat kelahiran, alamat tinggal sekarang, nomer telepon rumah / nomer ponsel, tanggal mulai kerja, kualifikasi akademik (combo box berisi pilihan D4, S1, S2), kesehatan (combo box berisi pilihan sangat sehat, sehat, kurang sehat, tidak sehat), catatan hukum (combo box berisi pilihan tidak pernah, hukuman ringan 1-3 kali, dan hukuman ringan > 3 kali), sertifikat (combo box berisi pilihan memiliki sertifikat pendidik dan tidak memiliki sertifikat pendidik), golongan (combo box berisi | [√] Berhasil<br>[ ] Gagal |



|  |  |  |  |  |
|--|--|--|--|--|
|  |  |  | pilihan IIIC, IIID, IVA, IVB, IVC, IVD, IVE)<br>DP3 (combo box berisi pilihan (nilai amat baik untuk unsur kesetiaan dan nilai amat baik untuk unsur lainnya dan nilai amat baik untuk unsur kesetiaan dan nilai baik untuk unsur lainnya), tugas tambahan (combo box yang berisi pilihan memiliki tugas tambahan dan tidak memiliki tugas tambahan) |  |
|  |  | Mengisi seluruh isian form tambah data guru<br>Klik tombol “ <i>save</i> ” | Menampilkan pesan “SUKSES ENTRY DATA”  | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
|  |  | Klik tombol “ <i>cancel</i> ”  | Menampilkan halaman data daftar guru dan tabel berisi no, nip, nama pegawai, status, detail yang didalamnya terdapat tombol icon mata (detail), edit, silang (non aktifkan user), centang (mengaktifkan user), refresh (menganti password menjadi default = 1234), serta tombol “nilai baru”   | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |

|  |                            |  |  |                           |
|--|----------------------------|--|--|---------------------------|
|  |                            | Belum Mengisi Field Input Data Guru                      | Menampilkan pesan “Isi Isian Ini” (untuk textfield) atau “pilih salah satu dari item terdaftar” (untuk combo box)  | [√] Berhasil<br>[ ] Gagal |
|  |                            | Inputan Data Guru Tidak Valid                            | Menampilkan pesan “Masukkan angka!” (Untuk inputan NIP dan Nomer Telepon jika bukan angka)   | [√] Berhasil<br>[ ] Gagal |
|  |                            | Inputan NIP SAMA   | Menampilkan pesan “GAGAL! Data Sudah Ada”  | [√] Berhasil<br>[ ] Gagal |
|  | Manajemen Data Guru (Edit) | Klik menu “Data Guru”                                    | Menampilkan halaman data daftar guru dan tabel berisi no, nip, nama pegawai, status, detail yang didalamnya terdapat tombol icon mata (detail), edit, silang (non aktifkan user), centang (mengaktifkan user), refresh (menganti password menjadi default = 1234), serta tombol “nilai baru” | [√] Berhasil<br>[ ] Gagal |
|  |                            | Klik icon “Edit” pada salah satu baris yang ingin diedit | Menampilkan halaman edit data guru yang berisi form biodata profil guru yang telah terisi level, NIP, Nama Pegawai, Jenis Kelamin, Tanggal Lahir, Alamat Kelahiran, Alamat Tinggal   | [√] Berhasil<br>[ ] Gagal |

|  |   |  |   |  |
|--|---|--|---|--|
|  |   |  | Sekarang, Nomer Telepon Rumah / Nomer Ponsel, Tanggal Mulai Kerja, Tahun Ajaran, Nilai Kualifikasi, Nilai Kesehatan, Nilai Sertifikat, Nilai Golongan, Nilai Catatan Hukum, Nilai DP3, dan Nilai Tambahan |  |
|  | Menggubah isian form edit data guru dan Klik tombol "Save"          | Menampilkan pesan "Sukses Edit Data Guru"  | [√] Berhasil<br>[ ] Gagal   |  |
|  | Menggubah isian form edit data daftar guru dan Klik tombol "Cancel" | Menampilkan halaman data daftar guru dan tabel berisi no, nip, nama pegawai, status, detail yang didalamnya terdapat tombol icon mata (detail), edit, silang (non aktifkan user), centang (mengaktifkan user), refresh (menganti password menjadi default = 1234), serta tombol "nilai baru" | [√] Berhasil<br>[ ] Gagal   |  |
|  | Belum Mengisi Field   | Menampilkan pesan "Masukkan angka!" (Untuk inputan NIP dan Nomer Telepon jika bukan angka)   | [√] Berhasil<br>[ ] Gagal   |  |
|  | Inputan NIP SAMA  | Menampilkan pesan "Gagal, NIP sudah digunakan"   | [√] Berhasil<br>[ ] Gagal   |  |

|                                     |   |   |                           |
|-------------------------------------|---|---|---------------------------|
| Manajemen Tambah Nilai Guru (View)  | Klik menu “Data Guru”                               | Menampilkan halaman data daftar guru dan tabel berisi no, nip, nama pegawai, status, detail yang didalamnya terdapat tombol icon mata (detail), edit, silang (non aktifkan user), centang (mengaktifkan user), refresh (menganti password menjadi default = 1234), serta tombol “nilai baru”  | [√] Berhasil<br>[ ] Gagal |
|                                     | Klik tombol “Tambah Nilai” pada salah baris ditabel | Menampilkan halaman Tambah Nilai Guru berupa form berisi level, NIP, nama pegawai, jenis kelamin, tanggal lahir, alamat lahir, nomer telepon rumah/nomer ponsel, tanggal mulai kerja dari function gettambahnilai() yang berasal dari tabel capres, tb_nilai_pm, dan login berserta form berupa tahun ajaran, kualifikasi akademik, kesehatan, catatan hukum, sertifikat, golongan, DP3, dan tugas tambahan | [√] Berhasil<br>[ ] Gagal |
| Manajemen Tambah Nilai Guru (Input) | Klik menu “Data Guru”                               | Menampilkan halaman data daftar guru dan tabel berisi no, nip, nama pegawai, status, detail yang didalamnya terdapat tombol icon mata (detail),   | [√] Berhasil<br>[ ] Gagal |

|  |  |   |  |                           |
|--|--|---|--|---------------------------|
|  |  |   | edit, silang (non aktifkan user), centang (mengaktifkan user), refresh (menganti password menjadi default = 1234), serta tombol “nilai baru”   |                           |
|  |  | Klik tombol “Tambah Nilai” pada salah baris ditabel     | Menampilkan halaman Tambah Nilai Guru berupa form berisi level, NIP, nama pegawai, jenis kelamin, tanggal lahir, alamat lahir, nomer telepon rumah/nomer ponsel, tanggal mulai kerja dari function gettambahnilai() yang berasal dari tabel capres, tb_nilai_pm, dan login beserta form berupa tahun ajaran, kualifikasi akademik, kesehatan, catatan hukum, sertifikat, golongan, DP3, dan tugas tambahan | [√] Berhasil<br>[ ] Gagal |
|  |  | Mengisi form tambah nilai guru dan Klik tombol “Save”   | Menampilkan pesan “Berhasil Entry Data”  | [√] Berhasil<br>[ ] Gagal |
|  |  | Mengisi form tambah nilai guru dan Klik tombol “cancel” | Menampilkan halaman data daftar guru dan tabel berisi no, nip, nama pegawai, status, detail yang didalamnya terdapat tombol icon mata (detail), edit, silang (non aktifkan user), centang  | [√] Berhasil<br>[ ] Gagal |

|   |   |                   |  |                           |
|---|---|-------------------|--|---------------------------|
|   |   |                   | (mengaktifkan user), refresh (menganti password menjadi default = 1234), serta tombol “nilai baru” |                           |
|   |   | Tahun Ajaran Sama | Menampilkan pesan “GAGAL! Data Sudah Ada”  | [√] Berhasil<br>[ ] Gagal |
| Manajemen Data User (Update) Aktif Status User    | Klik menu “Data Guru”                       |                   | Menampilkan halaman data daftar guru berupa nama pegawai dan status                                | [√] Berhasil<br>[ ] Gagal |
|   | Klik icon “centang √” pada salah satu baris |                   | Menampilkan pesan “User berhasil diaktifkan!”  | [√] Berhasil<br>[ ] Gagal |
| Manajemen Data User (Update) Nonaktif Status User | Klik menu “Data Guru”                       |                   | Menampilkan halaman data daftar guru berupa nama pegawai dan status dari tabel capres              | [√] Berhasil<br>[ ] Gagal |
|   | Klik icon “silang X” pada salah satu baris  |                   | Menampilkan pesan “User berhasil dinonaktifkan!”   | [√] Berhasil<br>[ ] Gagal |
| Manajemen Data User (Update) Reset Password User  | Klik menu “Data Guru”                       |                   | Menampilkan halaman data daftar guru berupa nama pegawai dan status                                | [√] Berhasil<br>[ ] Gagal |
|   | Klik icon “refresh” pada salah satu baris   |                   | Menampilkan pesan “Password Berhasil Direset dengan Hasil = 1234”                                  | [√] Berhasil<br>[ ] Gagal |

|                                    |  |  |                           |
|------------------------------------|--|--|---------------------------|
| Manajemen Data Pribadi Guru (View) | Klik menu “Biodata Pribadi”                                | Menampilkan halaman Biodata Pribadi Guru dengan data NIP, nama, jenis kelamin, tempat tanggal lahir, alamat, dan no hp | [√] Berhasil<br>[ ] Gagal |
| Manajemen Pribadi Guru (Edit)      | Klik menu “Biodata Pribadi”                                | Menampilkan halaman Biodata Pribadi Guru dengan data NIP, nama, jenis kelamin, tempat tanggal lahir, alamat, dan no hp | [√] Berhasil<br>[ ] Gagal |
|                                    | Klik tombol “Edit Biodata”                                 | Menampilkan form pop up edit biodata berisi NIP, nama, jenis kelamin, tempat tanggal lahir, alamat dan no hp           | [√] Berhasil<br>[ ] Gagal |
|                                    | Menggubah isian form edit biodata dan Klik tombol “Save”   | Menampilkan pesan “Berhasil, Data Berhasil di Update”  | [√] Berhasil<br>[ ] Gagal |
|                                    | Menggubah isian form edit Biodata dan Klik tombol “cancel” | Menampilkan halaman Biodata Pribadi Guru dengan data NIP, nama, jenis kelamin, tempat tanggal lahir, alamat, dan no hp | [√] Berhasil<br>[ ] Gagal |
|                                    | Belum Mengisi Field  | Menampilkan pesan “Isi isian ini”  | [√] Berhasil<br>[ ] Gagal |
|                                    | Inputan Tidak Valid  | Menampilkan pesan “Gagal, NIP Harus Angka!” (jika NIP salah tipe data) atau pesan                                      | [√] Berhasil<br>[ ] Gagal |

|                                |  |  |   |                           |
|--------------------------------|--|--|---|---------------------------|
|                                |  |  | “Gagal, Nomer Harus Angka!” (jika nomer HP salah tipe data) |                           |
|                                |  | NIP yang Diinputkan Sama   | Menampilkan pesan “Gagal, NIP sudah digunakan”              | [√] Berhasil<br>[ ] Gagal |
| Update Password Pribadi (Edit) | Klik menu “Biodata Pribadi”                                | Menampilkan halaman Biodata Pribadi Guru dengan data NIP, nama, jenis kelamin, tempat tanggal lahir, alamat, dan no hp | [√] Berhasil<br>[ ] Gagal                                   |                           |
|                                | Klik tombol “Ganti Password                                | Menampilkan halaman Ganti Password   | [√] Berhasil<br>[ ] Gagal                                   |                           |
|                                | Mengisi halaman form ganti password dan Klik tombol “Save” | Menampilkan pesan “Password Berhasil Diganti”  | [√] Berhasil<br>[ ] Gagal                                   |                           |
|                                | Belum Mengisi Field  | Menampilkan pesan “Isi isian ini”  | [√] Berhasil<br>[ ] Gagal                                   |                           |
|                                | Password Lama Salah  | Menampilkan pesan “Password Lama Yang Anda Masukkan Salah”   | [√] Berhasil<br>[ ] Gagal                                   |                           |
|                                | Password Baru Tidak Sama                                   | Menampilkan pesan “Password Baru Yang Anda Masukkan Tidak Sama”  | [√] Berhasil<br>[ ] Gagal                                   |                           |



|   |   |  |                           |
|---|---|--|---------------------------|
| Melihat Kriteria Penilaian Kinerja Guru(View) | Klik menu “Kriteria Penilaian”                            | Menampilkan halaman Kriteria Penilaian Kinerja       | [√] Berhasil<br>[ ] Gagal |
| Melihat Nilai Kinerja (View)                  | Klik menu “Nilai Kinerja”                                 | Menampilkan halaman Nilai Kinerja Guru               | [√] Berhasil<br>[ ] Gagal |
| Manajemen Nilai Ideal Profile Matching (View) | Klik menu “Perhitungan Profile”                           | Menampilkan halaman Data Profile Matching            | [√] Berhasil<br>[ ] Gagal |
|   | Klik tombol “Update Nilai Ideal”                          | Menampilkan halaman nilai kinerja ideal              | [√] Berhasil<br>[ ] Gagal |
| Manajemen Nilai Ideal Profile Matching (Edit) | Klik menu “Perhitungan Profile”                           | Menampilkan halaman Data Profile Matching            | [√] Berhasil<br>[ ] Gagal |
|   | Klik tombol “Update Nilai Ideal”                          | Menampilkan halaman nilai kinerja ideal              | [√] Berhasil<br>[ ] Gagal |
|   | Klik tombol “Edit” pada salah satu baris                  | Menampilkan halaman edit nilai ideal                 | [√] Berhasil<br>[ ] Gagal |
|   | Menganti isian form edit nilai ideal dan Klik tombol save | Menampilkan pesan “Berhasil, Data Berhasil Disimpan” | [√] Berhasil<br>[ ] Gagal |
|   | Menganti isian form nilai ideal dan Klik tombol “Batal”   | Menampilkan halaman nilai kinerja Ideal              | [√] Berhasil              |

|  |   |   |  |              |
|--|---|---|--|--------------|
|  |   |   |  | [ ] Gagal    |
|  |   | Belum Mengisi Field   | Menampilkan pesan “Isi Isian ini”                    | [√] Berhasil |
|  |   | Salah Tipe Data   | Menampilkan pesan “GAGAL! Nilai Ideal Harus Angka    | [ ] Gagal    |
|  |   |   |  | [√] Berhasil |
|  |   |   |  | [ ] Gagal    |
|  | Manajemen Bobot Kriteria AHP (View)                   | Klik menu “Perhitungan AHP”   | Menampilkan halaman Perhitungan AHP                  | [√] Berhasil |
|  |   |   |  | [ ] Gagal    |
|  | Manajemen Bobot Kriteria AHP (Edit)                   | Klik menu “Perhitungan AHP”   | Menampilkan halaman Perhitungan AHP                  | [√] Berhasil |
|  |   |   |  | [ ] Gagal    |
|  |   | Klik tombol “Update”  | Menampilkan halaman Update Bobot Kriteria            | [√] Berhasil |
|  |   |   |  | [ ] Gagal    |
|  |   | Mengisi form Update Bobot Nilai AHP dan Klik tombol “Save”          | Menampilkan pesan “Berhasil, Data Berhasil Diupdate” | [√] Berhasil |
|  |   |   |  | [ ] Gagal    |
|  |   | . Menganti isian form update bobot dan Klik tombol “Batal” kriteria | Menampilkan halaman Perhitungan AHP                  | [√] Berhasil |
|  |   |   |  | [ ] Gagal    |
|  | Usecase Melihat Nilai Kinerja Profile Matching (View) | Klik menu “Perhitungan Profile”                                     | Menampilkan halaman Data Profile Matching            | [√] Berhasil |
|  |   |   |  | [ ] Gagal    |

|  |  |  |  |
|--|--|--|--|
| Melihat Bobot Kriteria AHP (View)                    | Klik menu “Perhitungan AHP”                | Menampilkan halaman Perhitungan AHP                      | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| Melihat Peringkat Nilai Kinerja(View)                | Klik menu “Hasil Peringkat”                | Menampilkan Halaman Hasil Penentuan Peringkat            | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| Manajemen Verifikasi Peringkat Nilai Kinerja (View)  | Klik menu “Nilai Kinerja”                  | Menampilkan halaman Nilai Kinerja Guru                   | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
|  | Klik tombol “Peringkat Nilai Kinerja Guru” | Menampilkan halaman peringkat nilai kinerja guru         | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| Manajemen Verifikasi Peringkat Nilai Kinerja (View)  | Klik menu “Nilai Kinerja”                  | Menampilkan halaman Nilai Kinerja Guru                   | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
|  | Klik tombol “Peringkat Nilai Kinerja Guru” | Menampilkan halaman peringkat nilai kinerja guru         | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
|  | Klik tombol “Verifikasi”                   | Menampilkan pesan “Berhasil, Data Berhasil Diverifikasi” | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| Melihat Peringkat Nilai Kinerja Terverifikasi (View) | Klik menu “Nilai Kinerja”                  | Menampilkan halaman Nilai Kinerja Guru                   | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
|  | Klik tombol “Peringkat Nilai Kinerja Guru” | Menampilkan halaman peringkat nilai kinerja guru         | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |

|   |  |  |  |
|---|--|--|--|
| Melihat Update Rekap Hasil Peringkat (View)     | Klik menu “Hasil Peringkat”                      | Menampilkan halaman Hasil Penentuan Peringkat            | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| Manajemen Update Rekap Hasil Peringkat (Update) | Klik menu “Hasil Peringkat”                      | Menampilkan halaman Hasil Penentuan Peringkat            | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
|   | Klik tombol “Selesai”                            | Menampilkan pesan “Berhasil, Rekap Data Telah Dilakukan” | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| Melihat Rekap Penilaian (View)                  | Klik menu “Rekap”                                | Menampilkan halaman Data Rekap                           | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
|   | Klik salah satu baris tahun ajaran dikolom tabel | Menampilkan tabel peringkat nilai guru                   | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
|   | Klik salah satu baris nama di kolom tabel        | Menampilkan halaman data rekap user                      | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| Logout  | Klik menu “Logout”                               | Menampilkan halaman Login sistem                         | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |