

**PENGEMBANGAN SISTEM INFORMASI AKADEMIK SEKOLAH
MENENGAH KEJURUAN (SMK)
(STUDI PADA SMK MUHAMMADIYAH 1 KEMLAGI)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Pendidikan

Disusun oleh:

Muhammad Taftiyan Ilham Akbar

NIM: 175150601111001



PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI

JURUSAN SISTEM INFORMASI

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2021

PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI AKADEMIK SEKOLAH MENENGAH KEJURUAN (SMK)
(Studi pada SMK Muhammadiyah 1 Kemlagi)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Pendidikan

Disusun Oleh :
Muhammad Taftiyan Ilham Akbar
NIM: 175150601111001

Skripsi ini telah diuji dan dinyatakan lulus pada
23 Juli 2021
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing 1

Satrio Agung Wicaksono, S.Kom., M.Kom.
NIP:198605212012121001

Dosen Pembimbing 2

Faizatul Amalia, S.Pd., M.Pd
NIK:2013098608212001

Mengetahui
Ketua Jurusan Sistem Informasi



Issa Arwani, S.Kom., M.Sc.
NIP: 19830922201212 1 003

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 23 Juli 2021



Muhammad Taftiyan Ilham Akbar

NIM: 175150601111001

ABSTRAK

Muhammad Taftian Ilham Akbar, Pengembangan Sistem Informasi Akademik Sekolah Menengah Kejuruan (Studi pada SMK Muhammadiyah 1 Kemlagi)

Pembimbing: Satrio Agung Wicaksono, S.Kom., M.Kom. dan Faizatul Amalia, S.Pd., M.Pd.

SMK Muhammadiyah 1 Kemlagi merupakan sekolah swasta di Kabupaten Mojokerto yang berdiri pada tahun 2016. Saat penelitian dilakukan SMK Muhammadiyah 1 Kemlagi masih menggunakan program pengolah data Microsoft Excell dan Microsoft Word yang tidak terhubung pada program basis data, sehingga sering dijumpai masalah dalam pengelolaan data dengan jumlah banyak seperti data yang tidak sinkron dan pencarian data yang lama serta sulit yang diakibatkan dokumen-dokumen banyak yang terpisah. Penelitian ini bertujuan memecahkan masalah tersebut dengan mengembangkan sebuah sistem informasi untuk mengolah data-data akademik sekolah. Sistem informasi yang dikembangkan bernama sistem informasi akademik SMK Muhammadiyah 1 Kemlagi atau disingkat SIAKAD SMK MUTU. Sistem informasi yang dikembangkan berbasis web dan dibangun menggunakan bahasa pemrograman PHP dengan *framework* Codeigniter yang berarsitektur MVC dan program basis data MySQL. Metode pengembangan sistem menerapkan model *waterfall* dan pada pemodelan sistem menggunakan bahasa UML (*Unified Modelling Language*), pengujian sistem menggunakan *blackbox testing* dan uji penerimaan pengguna (UAT). Hasil dari pengujian *blackbox* menyatakan terhadap 20 kasus uji fungsi didapati hasil validitas sebesar 100% dan sistem berjalan sesuai dengan spesifikasi yang telah dinyatakan, hasil pengujian penerimaan pengguna (UAT) didapati hasil sebesar 82,9% dari 10 (sepuluh) peserta yang diuji. Merujuk pada hasil pengujian tersebut dapat disimpulkan bahwa SIAKAD SMK MUTU dinyatakan valid dan sangat dapat diterima oleh pengguna.

Kata kunci: sistem informasi akademik, sekolah menengah kejuruan, *waterfall model*, UML, *blackbox testing*, UAT.

ABSTRACT

Muhammad Taftiyan Ilham Akbar, *Development of a Vocational High School Academic Information System (Study at Muhammadiyah 1 Kemlagi Vocational High School)*

Supervisors: Satrio Agung Wicaksono, S.KOM., M.Kom. and Faizatul Amalia, S.Pd., M.Pd.

SMK Muhammadiyah 1 Kemlagi is a private school in Mojokerto Regency which was established in 2016. When the research was conducted, SMK Muhammadiyah 1 Kemlagi was still using Microsoft Excel and Microsoft Word program for data processing. That's software did not connected to the database system, so there were often problems in managing data in a large amount. Many such as out of sync data and long and difficult data searches caused by many separate documents. This study aims to solve this problem by developing an information system to process school academic data. The information system developed is called the academic information system of SMK Muhammadiyah 1 Kemlagi or abbreviated as SIAKAD SMK MUTU. The information system developed is web-based and built using the PHP programming language with the Codeigniter framework with MVC architecture and MySQL database program. The system development method applies the waterfall model and the system modeling uses the UML (Unified Modeling Language) language, system testing uses blackbox testing and user acceptance test (UAT). The results of the blackbox test stated that the 20 function test cases found the validity of 100% and the system was running according to the stated specifications, the results of the user acceptance test (UAT) were found to be 82.9% of the 10 (ten) participants tested. Referring to the test results, it can be concluded that SIAKAD SMK MUTU is declared valid and very acceptable to users.

Keywords: academic information system, vocational high school, waterfall model, UML, blackbox testing, UAT.

| | DAFTAR ISI |
|--|-------------------|
| PENGESAHAN | ii |
| PERNYATAAN ORISINALITAS | iii |
| PRAKATA | iv |
| ABSTRAK | vi |
| ABSTRACT | vii |
| DAFTAR ISI | viii |
| DAFTAR TABEL | xi |
| DAFTAR GAMBAR | xv |
| DAFTAR LAMPIRAN | xxiv |
| BAB 1 PENDAHULUAN | 1 |
| 1.1 Latar Belakang..... | 1 |
| 1.2 Rumusan Masalah..... | 3 |
| 1.3 Tujuan | 3 |
| 1.4 Manfaat..... | 3 |
| 1.5 Batasan Masalah..... | 4 |
| 1.6 Sistematika Pembahasan..... | 4 |
| BAB 2 LANDASAN KEPUSTAKAAN | 6 |
| 2.1 Kajian Pustaka..... | 6 |
| 2.2 Profil SMK Muhammadiyah 1 Kemlagi | 7 |
| 2.2.1 Pelaksanaan Kegiatan Akademik SMK Muhammadiyah 1 Kemlagi..... | 9 |
| 2.3 Sistem Informasi | 9 |
| 2.3.1 Sistem Informasi Akademik..... | 9 |
| 2.4 Proses Bisnis..... | 10 |
| 2.4.1 <i>Business Process Model and Notation (BPMN)</i> | 10 |
| 2.5 <i>System Development Life Cycle (SDLC)</i> | 12 |
| 2.6 <i>Waterfall Model</i> | 12 |
| 2.7 <i>Unified Modeling Language (UML)</i> | 14 |
| 2.7.1 <i>Use Case Diagram</i> | 14 |

| | |
|---|-----------|
| 2.7.2 Activity Diagram..... | 16 |
| 2.7.3 Sequence Diagram | 17 |
| 2.7.4 Class Diagram..... | 19 |
| 2.8 Models Views Controllers (MVC) | 20 |
| 2.9 WEB..... | 21 |
| 2.9.1 Komponen Sistem Web..... | 21 |
| 2.9.2 Web Server..... | 22 |
| 2.9.3 PHP (<i>PHP Hypertext Preprocessor</i>) | 22 |
| 2.9.4 Peramban Web (<i>Web Browser</i>) | 22 |
| 2.10 Database Management System (DBMS) | 23 |
| 2.10.1 SQL | 24 |
| 2.10.2 MySQL | 24 |
| 2.10.3 Normalisasi Database | 24 |
| 2.11 Framework | 25 |
| 2.11.1 Codeigniter | 25 |
| 2.11.2 Bootstrap..... | 25 |
| 2.12 Skala Likert..... | 26 |
| 2.13 Pengujian | 26 |
| 2.13.1 Blackbox Testing..... | 26 |
| 2.13.2 User Acceptance Test | 27 |
| BAB 3 METODOLOGI | 30 |
| 3.1 Pengumpulan Data | 31 |
| 3.2 Studi Literatur | 31 |
| 3.3 Analisis Kebutuhan | 31 |
| 3.4 Perancangan | 32 |
| 3.5 Implementasi | 32 |
| 3.6 Pengujian | 32 |
| 3.7 Kesimpulan dan Saran | 33 |
| BAB 4 HASIL DAN PEMBAHASAN | 34 |
| 4.1 Analisis Kebutuhan | 34 |
| 4.1.1 Identifikasi Proses Bisnis Saat Ini | 34 |
| 4.1.2 Analisis Proses Bisnis Saat Ini | 47 |

| | |
|---|------------|
| 4.1.3 Deskripsi Sistem Informasi..... | 49 |
| 4.1.4 Identifikasi Pengguna..... | 50 |
| 4.1.5 Identifikasi Kebutuhan | 50 |
| 4.2 Perancangan | 56 |
| 4.2.1 Use Case Diagram | 57 |
| 4.2.2 Use Case Scenario | 57 |
| 4.2.3 Activity Diagram..... | 99 |
| 4.2.4 Arsitektur Sistem..... | 141 |
| 4.2.5 Class Diagram..... | 142 |
| 4.2.6 Sequence Diagram | 144 |
| 4.2.7 Perancangan Database | 187 |
| 4.2.8 Perancangan Antarmuka..... | 200 |
| 4.3 Implementasi | 240 |
| 4.3.1 Batasan Implementasi..... | 240 |
| 4.3.2 Spesifikasi Perangkat Lunak | 241 |
| 4.3.3 Implementasi Basis Data | 242 |
| 4.3.4 Implementasi Fungsi Utama | 250 |
| 4.3.5 Implementasi Antarmuka | 261 |
| 4.4 Pengujian | 291 |
| 4.4.1 Pengujian Validasi | 292 |
| 4.4.2 Pengujian Penerimaan Pengguna | 318 |
| BAB 5 PENUTUP | 321 |
| 5.1 Kesimpulan..... | 321 |
| 5.2 Saran | 322 |
| DAFTAR PUSTAKA..... | 323 |
| LAMPIRAN..... | 327 |

DAFTAR TABEL

| | |
|--|----|
| Tabel 2. 1 Notasi Diagram <i>Use Case</i> | 14 |
| Tabel 2. 2 Simbol-simbol Pada <i>Activity Diagram</i> | 16 |
| Tabel 2. 3 Simbol-Simbol Pada <i>Sequence Diagram</i> | 18 |
| Tabel 2. 4 Skala Likert..... | 26 |
| Tabel 4. 1 Tabel Analisis Proses Bisnis Saat ini..... | 47 |
| Tabel 4. 2 Identifikasi Pengguna Sistem..... | 50 |
| Tabel 4. 3 Identifikasi Fitur Semua Pengguna..... | 52 |
| Tabel 4. 4 Identifikasi Fitur Pengguna Admin Sistem..... | 52 |
| Tabel 4. 5 Identifikasi Kebutuhan Fungsional Admin BKK | 54 |
| Tabel 4. 6 Identifikasi Kebutuhan Fungsional Waka Kurikulum..... | 54 |
| Tabel 4. 7 Identifikasi Kebutuhan Fungsionalitas Pengguna Guru | 55 |
| Tabel 4. 8 Identifikasi Kebutuhan Fungsionalitas Pengguna Siswa | 56 |
| Tabel 4. 9 <i>Use Case Scenario</i> Melakukan <i>Login</i> Sistem | 58 |
| Tabel 4. 10 <i>Use Case Scenario</i> melakukan <i>Logout</i> Sistem | 59 |
| Tabel 4. 11 <i>Use Case Scenario</i> Melihat Kalender Akademik | 59 |
| Tabel 4. 12 <i>Use Case Scenario</i> Melihat Data Jurusan | 60 |
| Tabel 4. 13 <i>Use Case Scenario</i> Menambah Data Jurusan | 61 |
| Tabel 4. 14 <i>Use Case Scenario</i> Mengubah Data Jurusan | 62 |
| Tabel 4. 15 <i>Use Case Scenario</i> Menghapus Data Jurusan..... | 63 |
| Tabel 4. 16 <i>Use Case Scenario</i> Melihat Data Kelas | 64 |
| Tabel 4. 17 <i>Use Case Scenario</i> Menambah Data Kelas | 64 |
| Tabel 4. 18 <i>Use Case Scenario</i> Mengubah Data Kelas | 65 |
| Tabel 4. 19 <i>Use Case Scenario</i> Menghapus Data Kelas..... | 66 |
| Tabel 4. 20 <i>Use Case Scenario</i> Melihat Data Mata Pelajaran Berdasarkan Jurusan | 67 |
| Tabel 4. 21 <i>Use Case Scenario</i> Menambah Data Mata Pelajaran | 68 |
| Tabel 4. 22 <i>Use Case Scenario</i> Mengubah Data Mata Pelajaran | 69 |
| Tabel 4. 23 <i>Use Case Scenario</i> Menghapus Data Mata Pelajaran..... | 70 |
| Tabel 4. 24 <i>Use Case Scenario</i> Melihat Data Guru..... | 71 |
| Tabel 4. 25 <i>Use Scenario</i> Melihat Detail Guru | 72 |

| | |
|---|----|
| Tabel 4. 26 Use Case Scenario Menambah Data Guru..... | 72 |
| Tabel 4. 27 Use Case Scenario Mengubah Data Guru..... | 73 |
| Tabel 4. 28 Use Case Scenario Menghapus Data Guru | 74 |
| Tabel 4. 29 Use Case Scenario Melihat Data Siswa..... | 75 |
| Tabel 4. 30 Use Case Scenario Melihat Detail Siswa..... | 76 |
| Tabel 4. 31 Use Case Scenario Menambah Data Siswa..... | 77 |
| Tabel 4. 32 Use Case Scenario Mengubah Data Siswa..... | 78 |
| Tabel 4. 33 Use Case Scenario Menghapus Data Siswa..... | 79 |
| Tabel 4. 34 Use Case Scenario Mencari Data Siswa | 79 |
| Tabel 4. 35 Use Case Scenario Melihat Data Jadwal Pelajaran Berdasarkan Kelas | 80 |
| Tabel 4. 36 Use Case Scenario Menambah Data Jadwal Pelajaran..... | 81 |
| Tabel 4. 37 Use Case Scenario Mengubah Data Jadwal Pelajaran..... | 82 |
| Tabel 4. 38 Use Case Scenario Menghapus Data Jadwal Pelajaran | 83 |
| Tabel 4. 39 Use Case Scenario Membuat Kegiatan Akademik..... | 84 |
| Tabel 4. 40 Use Case Scenario Mengubah Kalender Akademik..... | 85 |
| Tabel 4. 41 Use Case Scenario Menghapus Kalender Akademik | 86 |
| Tabel 4. 42 Use Case Scenario Memvalidasi Pelaksanaan Prakerin Siswa..... | 87 |
| Tabel 4. 43 Use Case Scenario Melihat Data Mitra Industri | 88 |
| Tabel 4. 44 Use Case Scenario Menambah Data Mitra Industri | 89 |
| Tabel 4. 45 Use Case Scenario Mengubah Data Mitra Industri | 90 |
| Tabel 4. 46 Use Case Scenario Menghapus Data Mitra Industri..... | 91 |
| Tabel 4. 47 Use Case Scenario Melihat Data Pelaksanaan Prakerin | 92 |
| Tabel 4. 48 Use Case Scenario Memvalidasi Kalender Akademik..... | 92 |
| Tabel 4. 49 Use Case Scenario Memvalidasi Penetapan Wali Kelas | 93 |
| Tabel 4. 50 Use Case Scenario Memperbarui Data Diri | 94 |
| Tabel 4. 51 Use Case Scenario Melihat Kelas Yang Diwakili | 95 |
| Tabel 4. 52 Use Case Scenario Memasukkan Nilai Hasil Evaluasi Siswa | 95 |
| Tabel 4. 53 Use Case Scenario Mengubah Nilai Hasil Evaluasi Siswa | 96 |
| Tabel 4. 54 Use Case Scenario Melihat Informasi Kelas Saat Ini..... | 97 |
| Tabel 4. 55 Use Case Scenario Melihat Nilai Hasil Evaluasi | 97 |
| Tabel 4. 56 Use Case Scenario Mengajukan Penuntasan Pelaksanaan Prakerin.. | 98 |

| | |
|---|-----|
| Tabel 4. 57 Use Case Scenario Mengubah Data Penuntasan Pelaksanaan Prakerin | 99 |
| Tabel 4. 58 Tabel Data Pengguna (Users) | 188 |
| Tabel 4. 59 Tabel Data Siswa..... | 188 |
| Tabel 4. 60 Tabel Data Guru..... | 190 |
| Tabel 4. 61 Tabel Data Jurusan | 191 |
| Tabel 4. 62 Tabel Data Agama | 191 |
| Tabel 4. 63 Tabel Data Kelas | 192 |
| Tabel 4. 64 Tabel Data Riwayat Kelas | 192 |
| Tabel 4. 65 Tabel Data Status Akademik..... | 193 |
| Tabel 4. 66 Tabel Data Orang Tua | 194 |
| Tabel 4. 67 Tabel Data Sekolah Sebelumnya | 194 |
| Tabel 4. 68 Tabel Data mitra Industri..... | 195 |
| Tabel 4. 69 Tabel Data Prakerin | 196 |
| Tabel 4. 70 Tabel Data Jadwal Pelajaran..... | 196 |
| Tabel 4. 71 Tabel Data Nilai | 197 |
| Tabel 4. 72 Tabel Data Kegiatan Akademik..... | 198 |
| Tabel 4. 73 Daftar Spesifikasi Perangkat Lunak | 241 |
| Tabel 4. 74 Kode Fungsi Utama <i>Render View</i> | 250 |
| Tabel 4. 75 Kode Fungsi Utama Render View Dengan Data | 251 |
| Tabel 4. 76 Kode Program Fungsi Utama Login Sistem Informasi | 252 |
| Tabel 4. 77 Kode Program Fungsi Utama <i>Logout</i> Sistem Informasi..... | 254 |
| Tabel 4. 78 Kode Program Fungsi Ambil Daftar Data..... | 254 |
| Tabel 4. 79 Kode Program Fungsi Utama Ambil Data Pada Lebih Dari Satu Tabel | 255 |
| Tabel 4. 80 Kode Program Fungsi Utama Lihat Detail Data..... | 255 |
| Tabel 4. 81 Kode Program Fungsi Utama <i>Input Data (Controller)</i> | 256 |
| Tabel 4. 82 Kode Program Fungsi Utama <i>Input Data (Model)</i> | 257 |
| Tabel 4. 83 Kode Program Fungsi Utama Ubah Data (<i>Controller</i>) | 257 |
| Tabel 4. 84 Kode Program Fungsi Utama Ubah Data (<i>Model</i>)..... | 258 |
| Tabel 4. 85 Kode Program Fungsi Utama Hapus Data | 259 |
| Tabel 4. 86 Kode Program Fungsi Utama ACC Data..... | 259 |

| | |
|---|-----|
| Tabel 4. 87 Kode Program Fungsi Utama Tolak Data..... | 259 |
| Tabel 4. 88 Kode Program Fungsi Utama Unggah Gambar | 260 |
| Tabel 4. 89 Kode Program Fungsi Utama Hapus Data Gambar | 261 |
| Tabel 4. 90 Rincian Pengujian Login Sistem..... | 292 |
| Tabel 4. 91 Rincian Pengujian Logout Sistem | 293 |
| Tabel 4. 92 Rincian Pengujian Lihat Kalender Akademik | 293 |
| Tabel 4. 93 Rincian Pengujian Mengelola Data Jurusan | 294 |
| Tabel 4. 94 Rincian Pengujian Mengelola Data Kelas | 296 |
| Tabel 4. 95 Rincian Pengujian Mengelola Data Mata Pelajaran | 298 |
| Tabel 4. 96 Rincian Pengujian Mengelola Data Guru..... | 300 |
| Tabel 4. 97 Rincian Pengujian Mengelola Data Siswa..... | 302 |
| Tabel 4. 98 Rincian Pengujian Mengelola Data Jadwal Pelajaran..... | 304 |
| Tabel 4. 99 Rincian Pengujian Mengelola Data Kegiatan Akademik..... | 306 |
| Tabel 4. 100 Rincian Pengujian Validasi Pelaksanaan Prakerin | 308 |
| Tabel 4. 101 Rincian Pengujian Mengelola Data Mitra Industri | 309 |
| Tabel 4. 102 Rincian Pengujian Melihat Pelaksanaan Prakerin | 311 |
| Tabel 4. 103 Rincian Pengujian Validasi Kalender Akademik | 311 |
| Tabel 4. 104 Rincian Pengujian Validasi Penetapan Kelas | 312 |
| Tabel 4. 105 Rincian Pengujian Memperbarui Data Diri..... | 313 |
| Tabel 4. 106 Rincian Pengujian Melihat Kelas Yang Diwakili | 313 |
| Tabel 4. 107 Rincian Pengujian Mengelola Nilai Rapor Siswa | 314 |
| Tabel 4. 108 Rincian Pengujian Melihat Rapor | 315 |
| Tabel 4. 109 Rincian Pengujian Mengelola Pengajuan Prakerina..... | 316 |
| Tabel 4. 110 Hasil Uji Penerimaan Pengguna | 318 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 2. 1 Struktur Organisasi SMK Muhammadiyah 1 Kemlagi | 8 |
| Gambar 2. 2 Elemen-Elemen Pada BPMN | 10 |
| Gambar 2. 3 Notasi-Notasi Pada <i>Swimlane</i> | 10 |
| Gambar 2. 4 Notasi-Notasi Pada <i>Connecting Objects</i> | 11 |
| Gambar 2. 5 Notasi-Notasi Pada <i>Artifacts</i> | 11 |
| Gambar 2. 6 Notasi-Notasi Pada <i>Flow Objects</i> | 12 |
| Gambar 2. 7 <i>Waterfall Model</i> | 13 |
| Gambar 2. 8 Contoh Diagram <i>Use case</i> | 16 |
| Gambar 2. 9 Contoh <i>Activity Diagram</i> | 17 |
| Gambar 2. 10 Contoh <i>Sequence Diagram</i> | 19 |
| Gambar 2. 11 <i>Class Diagram</i> Tunggal | 19 |
| Gambar 2. 12 Tipe Relasi Pada <i>Class Diagram</i> | 20 |
| Gambar 2. 13 Hubungan <i>Models</i> , <i>Views</i> , dan <i>Controllers</i> | 20 |
| Gambar 2. 14 Arsitektur Aplikasi Web Dinamis | 22 |
| Gambar 2. 15 <i>Blackbox Testing</i> | 27 |
| Gambar 2. 16 <i>User Acceptance Test</i> | 28 |
| Gambar 3. 1 Metodologi Penelitian..... | 30 |
| Gambar 4. 1 Pemodelan Proses Bisnis Pengelolaan Data Jurusan..... | 34 |
| Gambar 4. 2 Pemodelan Proses Bisnis Pengelolaan Data Kelas | 35 |
| Gambar 4. 3 Pemodelan Proses Bisnis Mengelola Data Mata Pelajaran..... | 36 |
| Gambar 4. 4 Pemodelan Proses Bisnis Pengelolaan Jadwal Pelajaran | 38 |
| Gambar 4. 5 Pemodelan Proses Bisnis Pengelolaan Kalender Akademik..... | 39 |
| Gambar 4. 6 Pemodelan Proses Bisnis Penentuan Wali Kelas..... | 40 |
| Gambar 4. 7 Pemodelan Proses Bisnis Pengelolaan Data Guru | 40 |
| Gambar 4. 8 Pemodelan Proses Bisnis Pengelolaan Data Siswa | 41 |
| Gambar 4. 9 Pemodelan Proses Bisnis Pengelolaan Mitra Industri..... | 42 |
| Gambar 4. 10 Pemodelan Proses Bisnis Validasi Pelaksanaan Prakerin..... | 44 |
| Gambar 4. 11 Pemodelan Proses Bisnis Penginformasian Jadwal Pelajaran | 45 |
| Gambar 4. 12 Pemodelan Proses Bisnis Evaluasi Siswa..... | 46 |
| Gambar 4. 13 Format Penomoran Kebutuhan | 51 |

| | |
|---|-----|
| Gambar 4. 14 Use Case Diagram | 57 |
| Gambar 4. 15 Activity Diagram Proses Login | 100 |
| Gambar 4. 16 Activity Diagram Proses Logout | 101 |
| Gambar 4. 17 Activity Diagram Proses Melihat Kalender Akademik | 101 |
| Gambar 4. 18 Activity Diagram Melihat Data Jurusan | 102 |
| Gambar 4. 19 Activity Diagram Menambah Data Jurusan | 103 |
| Gambar 4. 20 Activity Diagram Mengubah Data Jurusan | 104 |
| Gambar 4. 21 Activity Diagram Menghapus Data Jurusan | 105 |
| Gambar 4. 22 Activity Diagram Melihat Data Kelas | 106 |
| Gambar 4. 23 Activity Diagram Menambah Data Kelas | 106 |
| Gambar 4. 24 Activity Diagram mengubah Data Kelas | 107 |
| Gambar 4. 25 Activity Diagram Menghapus Data Kelas | 108 |
| Gambar 4. 26 Activity Diagram Menetapkan Wali Kelas | 109 |
| Gambar 4. 27 Activity Diagram Melihat Data Mata Pelajaran Berdasarkan Jurusan | 110 |
| Gambar 4. 28 Activity Diagram Menambah Data Mata Pelajaran | 111 |
| Gambar 4. 29 Activity Diagram Mengubah Data Mata Pelajaran | 112 |
| Gambar 4. 30 Activity Diagram Menghapus Mata Pelajaran | 113 |
| Gambar 4. 31 Activity Diagram Melihat Data Guru | 114 |
| Gambar 4. 32 Activity Diagram Melihat Detail Data Guru | 114 |
| Gambar 4. 33 Activity Diagram menambah Data Guru | 115 |
| Gambar 4. 34 Activity Diagram Mengubah Data Guru | 116 |
| Gambar 4. 35 Activity Diagram Menghapus Data Guru | 116 |
| Gambar 4. 36 Activity Diagram Melihat Data Siswa | 117 |
| Gambar 4. 37 Activity Diagram Melihat Detail Siswa | 118 |
| Gambar 4. 38 Activity Diagram Menambah Data Siswa | 119 |
| Gambar 4. 39 Activity Diagram Mengubah Data Siswa | 120 |
| Gambar 4. 40 Activity Diagram Menghapus Data Siswa | 121 |
| Gambar 4. 41 Activity Diagram Mencari Data Siswa | 121 |
| Gambar 4. 42 Activity Diagram Melihat Data Jadwal Pelajaran Berdasarkan Kelas | 122 |
| Gambar 4. 43 Activity Diagram Menambah Data Jadwal Pelajaran | 123 |

| | |
|---|-----|
| Gambar 4. 44 Activity Diagram Mengubah Data Jadwal Pelajaran | 124 |
| Gambar 4. 45 Activity Diagram menghapus Data Jadwal Pelajaran | 125 |
| Gambar 4. 46 Activity Diagram Membuat Kegiatan Akademik | 126 |
| Gambar 4. 47 Activity Diagram Mengubah Kegiatan Akademik | 127 |
| Gambar 4. 48 Activity Diagram Menghapus Kegiatan Akademik | 128 |
| Gambar 4. 49 Activity Diagram Memvalidasi Prakerin Siswa | 129 |
| Gambar 4. 50 Activity Diagram Melihat Mitra Industri | 130 |
| Gambar 4. 51 Activity Diagram Menambah Mitra Industri | 131 |
| Gambar 4. 52 Activity Diagram Mengubah Mitra Industri | 132 |
| Gambar 4. 53 Activity Diagram Menghapus Mitra Industri | 133 |
| Gambar 4. 54 Activity Diagram Melihat Data Pelaksanaan Prakerin | 133 |
| Gambar 4. 55 Activity Diagram Memvalidasi Kalender Akademik | 134 |
| Gambar 4. 56 Activity Diagram Menetapkan Wali Kelas | 135 |
| Gambar 4. 57 Activity Diagram Memperbarui Data Diri | 136 |
| Gambar 4. 58 Activity Diagram Melihat Kelas Yang Diajar | 137 |
| Gambar 4. 59 Activity Diagram Memasukkan Nilai Hasil Evaluasi Siswa | 137 |
| Gambar 4. 60 Activity Diagram Mengubah Nilai Hasil Evaluasi | 138 |
| Gambar 4. 61 Activity Diagram Melihat Detail Kelas | 139 |
| Gambar 4. 62 Activity Diagram Melihat Hasil Evaluasi | 139 |
| Gambar 4. 63 Activity Diagram Pengajuan Penuntasan Prakerin | 140 |
| Gambar 4. 64 Activity Diagram Mengubah Data Pengajuan Prakerin | 141 |
| Gambar 4. 65 Arsitektur Sistem SIAKAD SMK MUTU | 142 |
| Gambar 4. 66 Class Diagram Bagian Controller | 142 |
| Gambar 4. 67 Class Diagram Bagian Model | 143 |
| Gambar 4. 68 Sequence Diagram Login Sistem | 144 |
| Gambar 4. 69 Sequence Diagram Logout Sistem | 145 |
| Gambar 4. 70 Sequence Diagram Lihat Kalender Akademik | 146 |
| Gambar 4. 71 Sequence Diagram Lihat Daftar Jurusan | 146 |
| Gambar 4. 72 Sequence Diagram Tambah Data Jurusan | 147 |
| Gambar 4. 73 Sequence Diagram Ubah Data Jurusan | 148 |
| Gambar 4. 74 Sequence Diagram Hapus Data Jurusan | 149 |
| Gambar 4. 75 Sequence Diagram Lihat Daftar Kelas | 150 |

| | |
|--|-----|
| Gambar 4. 76 Sequence Diagram Lihat Detail Kelas | 150 |
| Gambar 4. 77 Sequence Diagram Tambah Data Kelas..... | 151 |
| Gambar 4. 78 Sequence Diagram Ubah Data Kelas | 152 |
| Gambar 4. 79 Sequence Diagram Hapus Data Kelas..... | 153 |
| Gambar 4. 80 Sequence Diagram Lihat Daftar Mata Pelajaran | 154 |
| Gambar 4. 81 Sequence Diagram Filter Daftar Mata Pelajaran | 154 |
| Gambar 4. 82 Sequence Diagram Tambah Data Mata Pelajaran | 155 |
| Gambar 4. 83 Sequence Diagram Ubah Data Mata Pelajaran..... | 156 |
| Gambar 4. 84 Sequence Diagram Hapus Data Mata Pelajaran | 157 |
| Gambar 4. 85 Sequence Diagram Lihat Daftar Jadwal Pelajaran | 157 |
| Gambar 4. 86 Sequence Diagram Lihat Detail Jadwal Pelajaran | 158 |
| Gambar 4. 87 Sequence Diagram Tambah Jadwal Pelajaran | 159 |
| Gambar 4. 88 Sequence Diagram Ubah Jadwal Pelajaran | 160 |
| Gambar 4. 89 Sequence Diagram Hapus Jadwal Pelajaran | 161 |
| Gambar 4. 90 Sequence Diagram Lihat Daftar Guru | 162 |
| Gambar 4. 91 Sequence Diagram Detail Data Guru | 162 |
| Gambar 4. 92 Sequence Diagram Tambah Data Guru | 163 |
| Gambar 4. 93 Sequence Diagram Ubah Data Guru | 164 |
| Gambar 4. 94 Sequence Diagram Hapus Data Guru | 164 |
| Gambar 4. 95 Sequence Diagram Lihat Daftar Siswa..... | 165 |
| Gambar 4. 96 Sequence Diagram Cari Data Siswa..... | 166 |
| Gambar 4. 97 Sequence Diagram Detail Data Siswa | 166 |
| Gambar 4. 98 Sequence Diagram Tambah Data Siswa | 167 |
| Gambar 4. 99 Sequence Diagram Ubah Data Siswa | 168 |
| Gambar 4. 100 Sequence Diagram Hapus Data Siswa | 169 |
| Gambar 4. 101 Sequence Diagram Lihat Daftar Kegiatan Akademik | 169 |
| Gambar 4. 102 Sequence Diagram Tambah Data Kegiatan Akademik..... | 170 |
| Gambar 4. 103 Sequence Diagram Ubah Data Kegiatan Akademik | 171 |
| Gambar 4. 104 Sequence Diagram Hapus Data Kegiatan Akademik..... | 172 |
| Gambar 4. 105 Sequence Diagram Verifikasi Pelaksanaan Prakerin | 173 |
| Gambar 4. 106 Sequence Diagram Lihat Daftar Mitra Industri | 174 |
| Gambar 4. 107 Sequence Diagram Tambah Data Mitra Industri..... | 174 |

| | |
|--|-----|
| Gambar 4. 108 Sequence Diagram Ubah Data Mitra Industri..... | 175 |
| Gambar 4. 109 Sequence Diagram Hapus Data Mitra Industri..... | 176 |
| Gambar 4. 110 Sequence Diagram Lihat Pelaksanaan Prakerin | 177 |
| Gambar 4. 111 Sequence Diagram Validasi Kalender Akademik..... | 178 |
| Gambar 4. 112 Sequence Diagram Verifikasi Wali Kelas | 178 |
| Gambar 4. 113 Sequence Diagram Perbarui Data Diri Guru..... | 179 |
| Gambar 4. 114 Sequence Diagram Perbarui Data Diri Siswa..... | 180 |
| Gambar 4. 115 Sequence Diagram Lihat Kelas Yang Diwakili | 181 |
| Gambar 4. 116 Sequence Diagram Input Nilai Hasil Evaluasi | 182 |
| Gambar 4. 117 Sequence Diagram Lihat Daftar Nilai Hasil Evaluasi..... | 182 |
| Gambar 4. 118 Sequence Diagram Ubah Nilai Hasil Evaluasi | 183 |
| Gambar 4. 119 Sequence Diagram Lihat Daftar Rapor | 184 |
| Gambar 4. 120 Sequence Diagram Detail Rapor | 184 |
| Gambar 4. 121 Sequence Diagram Pengajuan Penuntasan Prakerin | 185 |
| Gambar 4. 122 Sequence Diagram Lihat Daftar Pengajuan Prakerin | 186 |
| Gambar 4. 123 Sequence Diagram Ubah Pengajuan Penuntasan Prakerin | 187 |
| Gambar 4. 124 Physical Data Model Database Siakad | 200 |
| Gambar 4. 125 Antarmuka Sistem informasi Default | 201 |
| Gambar 4. 126 Rancangan Antarmuka Halaman Login | 201 |
| Gambar 4. 127 Rancangan Antarmuka Halaman Dashboard Admin Sistem | 202 |
| Gambar 4. 128 Rancangan Antarmuka Halaman Dashboard Admin BKK | 203 |
| Gambar 4. 129 Rancangan Antarmuka Halaman Dashboard Waka Kurikulum | 203 |
| Gambar 4. 130 Rancangan Antarmuka Halaman Dashboard Guru | 204 |
| Gambar 4. 131 Rancangan Antarmuka Halaman Dashboard Siswa | 204 |
| Gambar 4. 132 Rancangan Antarmuka Halaman Lihat Kalender Akademik..... | 205 |
| Gambar 4. 133 Rancangan Antarmuka Halaman Daftar Jurusan | 205 |
| Gambar 4. 134 Rancangan Antarmuka Halaman Tambah Data Jurusan..... | 206 |
| Gambar 4. 135 Rancangan Antarmuka Halaman Update Data Jurusan..... | 207 |
| Gambar 4. 136 Rancangan Antarmuka Halaman Hapus Data Jurusan..... | 207 |
| Gambar 4. 137 Rancangan Antarmuka Halaman Daftar Kelas | 208 |
| Gambar 4. 138 Rancangan Antarmuka Halaman Detail Kelas | 209 |
| Gambar 4. 139 Rancangan Antarmuka Halaman Tambah Data Kelas..... | 209 |

| | |
|--|-----|
| Gambar 4. 140 Rancangan Antarmuka Halaman Ubah Data Kelas | 210 |
| Gambar 4. 141 Rancangan Antarmuka Halaman Hapus Data Kelas..... | 211 |
| Gambar 4. 142 Rancangan Antarmuka Halaman Daftar Mata Pelajaran | 211 |
| Gambar 4. 143 Rancangan Antarmuka Halaman Tambah Mata Pelajaran | 212 |
| Gambar 4. 144 Rancangan Antarmuka Halaman Ubah Mata Pelajaran..... | 212 |
| Gambar 4. 145 Rancangan Antarmuka Halaman Hapus Data Mata Pelajaran .. | 213 |
| Gambar 4. 146 Rancangan Antarmuka Halaman Daftar Guru..... | 214 |
| Gambar 4. 147 Rancangan Antarmuka Halaman Detail Guru..... | 215 |
| Gambar 4. 148 Rancangan Antarmuka Halaman Tambah Data Guru | 215 |
| Gambar 4. 149 Rancangan Antarmuka Halaman <i>Update</i> Data Guru | 216 |
| Gambar 4. 150 Rancangan Antarmuka Halaman Hapus Data Guru | 216 |
| Gambar 4. 151 Rancangan Antarmuka Halaman Daftar Siswa..... | 217 |
| Gambar 4. 152 Rancangan Antarmuka Halaman Detail Siswa Bagian Satu..... | 218 |
| Gambar 4. 153 Rancangan Antarmuka Halaman Detail Siswa Bagian Dua | 218 |
| Gambar 4. 154 Rancangan Antarmuka Halaman Tambah Data Siswa Bagian Satu | 219 |
| Gambar 4. 155 Rancangan Antarmuka Halaman Tambah Data Siswa Bagian Dua | 219 |
| Gambar 4. 156 Rancangan Antarmuka Halaman <i>Update</i> Data Siswa Bagian Satu | 220 |
| Gambar 4. 157 Rancangan Antarmuka Halaman <i>Update</i> Data Siswa Bagian Dua | 220 |
| Gambar 4. 158 Rancangan Antarmuka Halaman Hapus Data Siswa | 221 |
| Gambar 4. 159 Rancangan Antarmuka Halaman Daftar Jadwal Pelajaran..... | 222 |
| Gambar 4. 160 Rancangan Antarmuka Halaman Detail Jadwal Pelajaran | 222 |
| Gambar 4. 161 Rancangan Antarmuka Halaman Tambah Jadwal Pelajaran..... | 223 |
| Gambar 4. 162 Rancangan Antarmuka Halaman Ubah Jadwal Pelajaran | 224 |
| Gambar 4. 163 Rancangan Antarmuka Halaman Hapus Jadwal Pelajaran..... | 224 |
| Gambar 4. 164 Rancangan Antarmuka Halaman Daftar Kegiatan Akademik.... | 225 |
| Gambar 4. 165 Rancangan Antarmuka Halaman Tambah Kegiatan Akademik.. | 226 |
| Gambar 4. 166 Rancangan Antarmuka Halaman Ubah Data Kegiatan Akademik | 227 |

| | |
|---|-----|
| Gambar 4. 167 Rancangan Antarmuka Halaman Hapus Data Kegiatan Akademik | 227 |
| Gambar 4. 168 Rancangan Antarmuka Halaman Validasi Pelaksanaan Prakerin | 228 |
| Gambar 4. 169 Rancangan Antarmuka Halaman Daftar Mitra Industri | 229 |
| Gambar 4. 170 Rancangan Antarmuka Halaman Tambah Data Mitra Industri .. | 229 |
| Gambar 4. 171 Rancangan Antarmuka Halaman Ubah Data Mitra Industri | 230 |
| Gambar 4. 172 Rancangan Antarmuka Halaman Hapus Data Mitra Industri | 230 |
| Gambar 4. 173 Rancangan Antarmuka Halaman Pelaksanaan Prakerin | 231 |
| Gambar 4. 174 Rancangan Antarmuka Halaman Validasi Kalender Akademik .. | 232 |
| Gambar 4. 175 Rancangan Antarmuka Halaman Validasi Wali Kelas | 233 |
| Gambar 4. 176 Rancangan Antarmuka Halaman Perbarui Data Diri Siswa | 233 |
| Gambar 4. 177 Rancangan Antarmuka Halaman Perbarui Data Diri Guru | 234 |
| Gambar 4. 178 Rancangan Antarmuka Halaman Kelas Yang Diwakili | 235 |
| Gambar 4. 179 Rancangan Antarmuka Halaman Nilai Hasil Evaluasi | 235 |
| Gambar 4. 180 Rancangan Antarmuka Halaman Daftar Nilai Hasil Evaluasi | 236 |
| Gambar 4. 181 Rancangan Antarmuka Halaman Ubah Nilai Hasil Evaluasi | 237 |
| Gambar 4. 182 Rancangan Antarmuka Halaman Daftar Rapor | 237 |
| Gambar 4. 183 Rancangan Antarmuka Halaman Detail Rapor | 238 |
| Gambar 4. 184 Rancangan Antarmuka Halaman Pengajuan Pelaksanaan Prakerin | 239 |
| Gambar 4. 185 Rancangan Antarmuka Halaman Daftar Pengajuan Prakerin | 239 |
| Gambar 4. 186 Rancangan Antarmuka Halaman Ubah Pengajuan Pelaksanaan Prakerin | 240 |
| Gambar 4. 188 Antarmuka Login Sistem | 261 |
| Gambar 4. 189 Antarmuka Logout Sistem | 262 |
| Gambar 4. 190 Antarmuka <i>Dashboard</i> | 262 |
| Gambar 4. 191 Antarmuka Kalender Akademik | 263 |
| Gambar 4. 192 Antarmuka Daftar Kegiatan Akademik | 263 |
| Gambar 4. 193 Antarmuka Tambah Kegiatan Akademik | 264 |
| Gambar 4. 194 Antarmuka Ubah Data Kegiatan Akademik | 264 |
| Gambar 4. 195 Antarmuka Hapus Data Kegiatan Akademik | 265 |
| Gambar 4. 196 Antarmuka Daftar Jurusan | 265 |
| Gambar 4. 197 Antarmuka Tambah Data Jurusan | 266 |

| | |
|---|-----|
| Gambar 4. 198 Antarmuka Ubah Data Jurusan | 266 |
| Gambar 4. 199 Antarmuka Hapus Data Jurusan | 267 |
| Gambar 4. 200 Antarmuka Daftar Kelas | 267 |
| Gambar 4. 201 Antarmuka Detail Data Kelas | 267 |
| Gambar 4. 202 Antarmuka Tambah Data Kelas | 268 |
| Gambar 4. 203 Antarmuka Ubah Data Kelas | 268 |
| Gambar 4. 204 Antarmuka Hapus Data Kelas | 269 |
| Gambar 4. 205 Antarmuka Data Mata Pelajaran | 269 |
| Gambar 4. 206 Antarmuka Filter Mata Pelajaran | 270 |
| Gambar 4. 207 Antarmuka Tambah Data Mata Pelajaran | 270 |
| Gambar 4. 208 Antarmuka Ubah Data Mata Pelajaran | 271 |
| Gambar 4. 209 Antarmuka Hapus Data Mata Pelajaran | 271 |
| Gambar 4. 210 Antarmuka Daftar Jadwal Pelajaran | 272 |
| Gambar 4. 211 Antarmuka Detail Jadwal Pelajaran | 272 |
| Gambar 4. 212 Antarmuka Detail Jadwal Pelajaran (Admin) | 273 |
| Gambar 4. 213 Antarmuka Tambah Data Jadwal Pelajaran | 273 |
| Gambar 4. 214 Antarmuka Ubah Data Jadwal Pelajaran | 274 |
| Gambar 4. 215 Antarmuka Hapus Data Jadwal Pelajaran | 274 |
| Gambar 4. 216 Antarmuka Daftar Guru | 275 |
| Gambar 4. 217 Antarmuka Detail Data Guru | 275 |
| Gambar 4. 218 Antarmuka Tambah Data Guru | 276 |
| Gambar 4. 219 Antarmuka Ubah Data Guru | 277 |
| Gambar 4. 220 Antarmuka Hapus Data Guru | 278 |
| Gambar 4. 221 Antarmuka Daftar Siswa | 278 |
| Gambar 4. 222 Antarmuka Detail Data Siswa | 278 |
| Gambar 4. 223 Antarmuka Cari Data Siswa | 279 |
| Gambar 4. 224 Antarmuka Tambah Data Siswa | 280 |
| Gambar 4. 225 Antarmuka Ubah Data Siswa | 281 |
| Gambar 4. 226 Antarmuka Hapus Data Siswa | 282 |
| Gambar 4. 227 Antarmuka Daftar Mitra Industri | 282 |
| Gambar 4. 228 Antarmuka Tambah Mitra Industri | 283 |
| Gambar 4. 229 Antarmuka Ubah Data Mitra Industri | 283 |

| | |
|--|-----|
| Gambar 4. 230 Antarmuka Hapus Data Mitra Industri..... | 284 |
| Gambar 4. 231 Antarmuka Verifikasi Prakerin Siswa | 284 |
| Gambar 4. 232 Antarmuka Verifikasi Wali Kelas | 285 |
| Gambar 4. 233 Antarmuka Verifikasi Kalender Akademik..... | 285 |
| Gambar 4. 234 Antarmuka Perbarui Data Diri Siswa | 286 |
| Gambar 4. 235 Antarmuka Perbarui Data Diri Guru..... | 286 |
| Gambar 4. 236 Antarmuka Kelas Yang Diwakili | 287 |
| Gambar 4. 237 Antarmuka Data Nilaiya..... | 287 |
| Gambar 4. 238 Antarmuka Detail Nilai Siswa | 288 |
| Gambar 4. 239 Antarmuka Tambah Data Nilai Siswa..... | 288 |
| Gambar 4. 240 Antarmuka Ubah Data Nilai Siswa | 289 |
| Gambar 4. 241 Antarmuka Data Rapor..... | 289 |
| Gambar 4. 242 Antarmuka Detail Rapor..... | 290 |
| Gambar 4. 243 Antarmuka Data Pengajuan Prakerin..... | 290 |
| Gambar 4. 244 Antarmuka Pengajuan Penuntasan Prakerin | 291 |
| Gambar 4. 245 Antarmuka Ubah Data Pengajuan Prakerin | 291 |
| Gambar 5. 1 Penjelasan Sistem Informasi Pada Waka Kurikulum..... | 337 |
| Gambar 5. 2 Pengujian Sistem Informasi Oleh Admin BKK | 337 |
| Gambar 5. 3 Pengisian Kuesioner Oleh Siswa..... | 337 |



DAFTAR LAMPIRAN

| | |
|---|-----|
| Lampiran A Hasil Wawancara | 327 |
| Lampiran B Kuesioner UAT..... | 333 |
| Lampiran C Rekapitulasi Hasil Uji Penerimaan Pengguna | 336 |
| Lampiran D Dokumentasi..... | 337 |



1.1 Latar Belakang

Sekolah merupakan suatu sistem yang terdiri dari komponen-komponen yang saling terkait dan saling mempengaruhi dalam mencapai tujuan yang telah ditetapkan. Dalam pencapaian tujuan tersebut diperlukan pengelolaan dan administrasi persekolahan yang baik (Patsun, 2015). Sekolah perlu melakukan evaluasi dan perbaikan secara terus menerus demi mencapai visi dan misi yang telah ditentukan. Saat ini penggunaan teknologi informasi dalam lingkungan persekolahan sudah menjadi hal yang umum, namun penggunaan perangkat teknologi informasi pada banyak sekolah masih terbatas pada aplikasi umum, atau sudah menggunakan perangkat komputer namun belum terhubung dengan internet sehingga banyak terdapat kekurangan seperti lemahnya sistem administrasi, pengulangan pekerjaan (repetitif), kualitas data yang rendah. Sehingga untuk mengatasi masalah tersebut beberapa organisasi menerapkan sistem informasi dalam proses bisnisnya (Satia, 2015).

Sistem informasi adalah sebuah sistem yang dibuat oleh manusia dengan tujuan penyampaian informasi yang berisi sekumpulan prosedur-prosedur organisasi yang ada dan akan memberikan informasi bagi para pengambil keputusan dan untuk memanipulasi jalannya organisasi (Akbar, et al., 2021). Sistem informasi dibuat untuk mempermudah pekerjaan yang bersifat administratif sehingga lebih mudah dan meminimalisir terjadinya kesalahan informasi. Salah satu sistem informasi yang banyak diterapkan pada lingkungan sekolah adalah sistem informasi akademik sekolah yaitu suatu aplikasi yang membantu kegiatan pengelolaan data-data akademik di sekolah, data-data akademik meliputi data siswa, data guru, kenaikan kelas, perpindahan sekolah dan kelulusan siswa. Sekolah menengah kejuruan (SMK) memiliki perbedaan dengan tingkat pendidikan yang lain. SMK memiliki proses bisnis untuk melakukan pencatatan data akademik seperti praktik kerja industri, pelayanan bursa kerja khusus sekolah dan pencatatan pekerjaan alumni untuk evaluasi lebih lanjut. Perkembangan teknologi informasi pada bidang pendidikan saat ini memungkinkan untuk pelaksanaan aktivitas akademik secara daring, sistem informasi akademik yang banyak digunakan adalah sistem berbasis web, yang artinya perlu jaringan internet dalam pengaksesannya (Sutabri, 2012).

SMK Muhammadiyah 1 Kemlagi (SMK MUTU) merupakan sekolah swasta yang terletak di Jalan Kepuh Sawo, Desa Mojokumpul, Kec. Kemlagi, Mojokerto, Jawa Timur. SMK MUTU mengusung visi "menjadi sekolah islami yang kreatif, inovatif dan unggul dalam bidang IPTEK dan IMTAQ", SMK MUTU mencoba menjawab tantangan masa depan dengan menghadirkan inovasi dan kelengkapan fasilitas untuk menghadirkan pendidikan berkualitas di Mojokerto. Berdasarkan wawancara dengan Bapak Nur selaku kepala Bursa Kerja Khusus (BKK) SMK Muhammadiyah 1 Kemlagi saat ini masih menggunakan program pengolah data yang tidak terintegrasi dengan basis data (Microsoft Excel), sehingga masih pada

hasil wawancara pada Lampiran A terdapat permasalahan pengelolaan data akademik, data akademik yang terus bertambah membuat pengelolaan data akademik menjadi semakin susah dan rawan terjadi kesalahan.

Data yang berkualitas adalah data yang memenuhi tiga aspek, keakuratan, ketepatan waktu dan relevansi. SMK MUTU memiliki sarana prasarana teknologi informasi yang memadai dalam penerapan sistem informasi akademik sekolah. Sehingga diharapkan segera menerapkan sistem informasi akademik yang bersifat *Online*. Dengan bantuan sistem informasi, pencatatan menjadi lebih konsisten, pencarian data menjadi lebih mudah, dan pelaporan lebih mudah. Dengan adanya sistem informasi akademik sekolah diharapkan data akademik dapat tersinkronisasi dengan baik dan mengurangi inkonsistensi data sehingga kegiatan akademik di SMK Muhammadiyah 1 Kemlagi dapat berjalan dengan baik.

Model *waterfall* merupakan salah satu model dalam SDLC yang banyak digunakan dalam pengembangan sistem informasi, model *waterfall* digunakan apabila kebutuhan sistem telah teridentifikasi dengan jelas dan dipastikan tidak mengalami perubahan dalam kurun waktu pengembangan. Tahap – tahap model *waterfall* terdiri dari analisis kebutuhan, pemodelan, implementasi, pengujian dan evaluasi (Sommerville, 2011). Dalam pengembangan sistem informasi dilakukan pemodelan mengenai arsitektur dan desain sistem untuk memperoleh gambaran mengenai sistem informasi yang akan dibuat, bahasa pemodelan yang saat ini menjadi standar adalah *unified modelling language (UML)*. Setelah sistem berhasil dikembangkan, perlu dipastikan bahwa sistem informasi yang dikembangkan telah memenuhi ekspektasi yang diharapkan, karena itu dilakukan tahap pengujian untuk menguji fungsionalitas (*black-box testing*) dan pengujian terhadap penerimaan pengguna (*user acceptance test*). Hasil wawancara dengan pihak sekolah, pengguna sistem informasi akademik sekolah yang teridentifikasi adalah pengguna dalam lingkup internal sekolah saja yang meliputi siswa, guru, admin sistem, admin Bursa Kerja Khusus (BKK) Sekolah dan Waka Kurikulum.

Berdasarkan permasalahan yang terjadi, serta dengan pertimbangan, penjabaran dan pemikiran yang telah diuraikan. Penulis memutuskan untuk melakukan penelitian dan pengembangan sistem informasi berbasis web dengan menerapkan SDLC *waterfall* dalam bentuk skripsi dengan judul "**Pengembangan Sistem Informasi Akademik Sekolah Menengah Kejuruan (Studi Pada SMK Muhammadiyah 1 Kemlagi)**". Penelitian ini diharapkan dapat membantu kegiatan akademik di SMK Muhammadiyah 1 Kemlagi menjadi lebih baik dan lebih optimal dan menyesuaikan dengan perkembangan teknologi pada saat ini, sehingga dapat mewujudkan visi yang dijunjung oleh SMK Muhammadiyah 1 Kemlagi.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijabarkan sebelumnya, maka dapat ditarik rumusan masalah sebagai berikut:

1. Bagaimana mengatasi permasalahan pengelolaan data dan kegiatan yang terjadi pada SMK Muhammadiyah 1 Kemlagi menggunakan sistem informasi akademik?
2. Bagaimana tahapan pengembangan sistem informasi akademik pada SMK Muhammadiyah 1 Kemlagi?
3. Bagaimana hasil pengujian validasi dan penerimaan pengguna terhadap penerapan sistem informasi akademik SMK Muhammadiyah 1 Kemlagi?

1.3 Tujuan

Berdasarkan rumusan masalah yang telah disebutkan diatas, maka tujuan yang ingin dicapai pada penelitian ini adalah sebagai berikut:

1. Mengatasi permasalahan pengelolaan data dan kegiatan akademik yang terjadi pada SMK Muhammadiyah 1 Kemlagi dengan menggunakan sistem informasi akademik.
2. Mengetahui tahapan-tahapan yang dilakukan dalam proses pengembangan sistem informasi akademik SMK Muhammadiyah 1 Kemlagi.
3. Mengukur validasi sistem informasi akademik SMK Muhammadiyah 1 Kemlagi beserta dengan penerimaan pengguna sistem informasi akademik SMK Muhammadiyah 1 Kemlagi.

1.4 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat bagi berbagai pemangku kepentingan yang meliputi:

1. Bagi Sekolah

Mempermudah dan mempercepat proses bisnis dalam pengelolaan data akademik sekolah secara efektif dan efisien menggunakan sistem informasi.

2. Bagi Peneliti

Dapat mengimplementasikan ilmu yang diperoleh di bangku perkuliahan dengan praktik langsung pengembangan sistem informasi akademik sekolah menengah kejuruan (SMK).

3. Bagi Universitas Brawijaya

Sebagai bahan masukan/referensi pengembangan sistem informasi terhadap penelitian yang serupa.



1.5 Batasan Masalah

- Untuk menghindari kajian yang terlalu luas, maka ruang lingkup pada penelitian ini adalah mengkaji hal-hal sebagai berikut:
1. Pengembangan sistem informasi mengadopsi model *waterfall* yang meliputi tahapan analisis kebutuhan, perancangan sistem, implementasi sistem, dan pengujian sistem.
 2. Data yang digunakan dalam proses analisis kebutuhan sistem diperoleh dari wawancara dan observasi dengan SMK Muhammadiyah 1 Kemlagi.
 3. Sistem Informasi Akademik tidak termasuk fitur penerimaan peserta didik baru (PPDB).
 4. Pengembangan dibatasi sampai pada tahap pengujian.
 5. Pengujian menggunakan metode *Black-box testing*, dan *User Acceptance Test*.

1.6 Sistematika Pembahasan

Penelitian ini menggunakan sistematika pembahasan sebagai berikut:

BAB 1 PENDAHULUAN

Bab Pendahuluan menguraikan latar belakang penelitian, rumusan masalah, manfaat penelitian, batasan masalah, dan sistematika pembahasan.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini dijelaskan mengenai teori-teori yang mendukung dan relevan dengan penelitian yang digunakan. Serta terdapat profil dari tempat penelitian, serta disertakan beberapa penelitian sebelumnya yang memiliki beberapa kesamaan dengan penelitian ini.

BAB 3 METODOLOGI PENELITIAN

Pada bab ini membahas metode dan langkah-langkah yang digunakan dalam penelitian, yang meliputi pengumpulan data, studi literatur, analisis kebutuhan sistem, perancangan sistem, implementasi, pengujian dan terakhir adalah kesimpulan dan saran.

BAB 4 HASIL DAN PEMBAHASAN

Pada bab ini menjelaskan tentang proses pengumpulan data untuk penelitian, studi literatur yang relevan, hasil analisis kebutuhan sistem, hasil perancangan sistem, hasil implementasi dan hasil pengujian dengan melihat keluaran dengan hasil yang telah direncanakan pada tahap analisis kebutuhan sistem.



BAB 5 PENUTUP

Pada bab ini memuat kesimpulan dari pengembangan sistem informasi akademik sekolah menengah kejuruan (SMK) pada SMK Muhammadiyah 1 Kemlagi, serta saran untuk penelitian yang dilakukan guna pengembangan lebih lanjut.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Pada penelitian ini digunakan tiga referensi penelitian sebelumnya yang relevan dengan penelitian ini. Penggunaan referensi atas penelitian sebelumnya berguna untuk menggali informasi atas bidang yang diteliti sehingga penelitian akan lebih terarah sehingga penelitian dapat dilaksanakan dengan lebih mudah.

Penelitian berjudul Pengembangan Sistem Informasi Akademik Pada Sekolah Dasar Islam Terpadu Insantama (SDIT) Malang Berbasis Website oleh Andra Pargiyani, Bayu Priyambadha dan Ahmad Arwan. Penelitian ini menjabarkan proses pengembangan sistem informasi akademik dengan menggunakan model *waterfall* untuk mempermudah pengelolaan manajemen sekolah yang meliputi data penilaian hasil belajar dan pencatatan SPP. Pengembangan sistem informasi tersebut membuat proses administasi menjadi lebih mudah karena petugas tidak perlu lagi menggunakan program *excel* yang kurang praktis dalam pendistribusian serta rawan terinfeksi virus komputer (Pargiyani, et al., 2019).

Penelitian berjudul Perancangan Sistem Informasi Akademik Pada Sekolah Menengah Kejuruan Negeri (SMKN) 1 Linau Buo Menggunakan PHP MySQL oleh Eko Amri Jaya, Veni Wedyawati, dan M.Ridwan. Pada penelitian ini dilakukan pembuatan sistem informasi akademik sekolah menengah kejuruan dengan menggunakan bahasa pemrograman PHP dan MySQL sebagai DBMSnya. Penelitian dilatarbelakangi oleh penyebaran informasi yang kurang efektif akibat masih menggunakan cara manual dalam penerapannya. Penelitian ini dibatasi pada sistem informasi data promosi, pendaftaran, data sekolah, data siswa, data guru, data karyawan dan informasi agenda kegiatan sekolah. Hasil penelitian didapati bahwa dengan menerapkan sistem informasi penyampaian informasi akademik menjadi lebih mudah dan fleksibel, sehingga masyarakat umum dapat memberi saran dan kritik terhadap manajemen informasi sekolah (Jaya, et al., 2018).

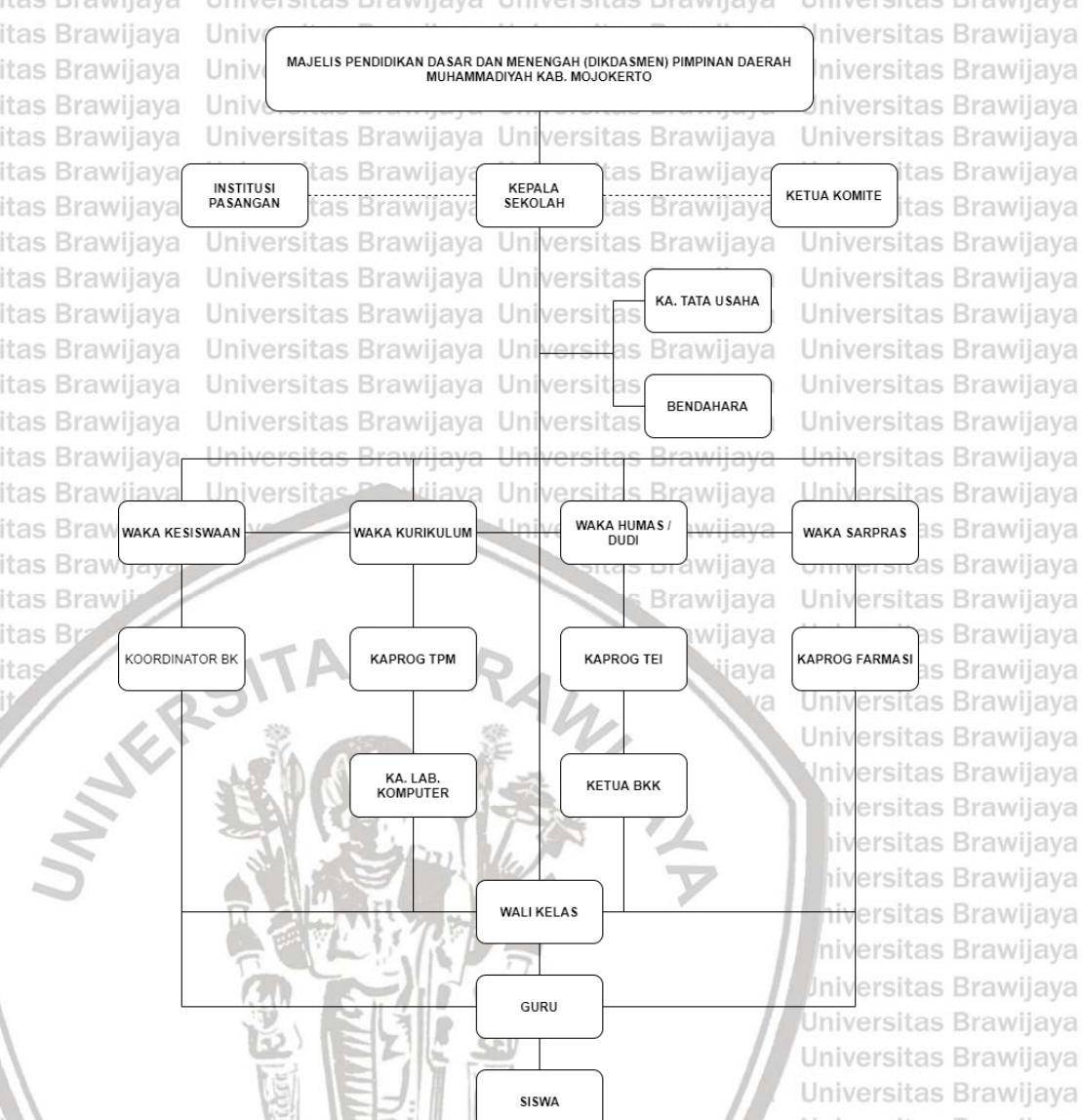
Penelitian berjudul Sistem Informasi Akademik SMA Nurul Falaah Berbasis WEB oleh Maulana Rahman dan Juliyanto. Pada Penelitian ini dilakukan pengembangan sistem informasi berbasis web untuk mengatasi permasalahan distribusi informasi dan pengelolaan data akademik yang masih dikelola secara manual di beberapa komputer sekolah saja (belum disimpan secara *online*). Penelitian ini menerapkan *SDLC model waterfall* dan menggunakan *UML* yang terdiri dari *use case diagram*, *entity relationship diagram* dan *activity diagram* dalam perancangan sistemnya, teknologi yang digunakan meliputi PHP, MySQL, HTML, CSS dan Javascript. Data akademik yang diolah oleh sistem informasi meliputi data siswa, data karyawan dan data mengenai pelajaran sekolah. Hasil dari penelitian ini didapati bahwa pengelolaan data siswa, data guru dan data karyawan serta mempermudah pengaksesan informasi tentang pengumuman sekolah oleh publik (Rahman & Juliyanto, 2019).

Dari tiga penelitian yang telah dijabarkan sebelumnya dapat diambil kesimpulan bahwa sistem informasi dapat membantu kegiatan akademik sekolah menjadi lebih efektif dan efisien karena akses yang lebih mudah, data terintegrasi dengan basis data dan dapat mengurangi pekerjaan yang berulang. Penelitian yang dilakukan oleh penulis adalah pengembangan sistem informasi akademik sekolah berbasis *web* yang dapat melakukan integrasi data dengan basis data sekolah sehingga dapat mengurangi redundansi data dan inkonsistensi data, selain itu dapat mengurangi pekerjaan yang bersifat repetitif seperti pemindahan *file* dari komputer satu ke komputer lain. Perbedaan dari penelitian yang dilakukan penulis dengan penelitian sebelumnya meliputi subjek penelitian, tempat pelaksanaan penelitian, kebutuhan fungsional dan non fungsional sistem, dan proses bisnis yang diadopsi.

2.2 Profil SMK Muhammadiyah 1 Kemlagi

Sekolah Menengah Kejuruan Muhammadiyah 1 Kemlagi atau biasa disebut SMK MUTU Kemlagi diselenggarakan oleh Majelis Pendidikan Dasar dan Menengah Pimpinan Daerah Muhammadiyah Kabupaten Mojokerto, SMK MUTU Kemlagi telah mendapatkan izin pendirian yang ditandatangani oleh Bupati Mojokerto dengan SK nomor 188.45/632/HK/614-012/2016 pada tanggal 1 September 2016. SMK MUTU Kemlagi dipimpin oleh M. Maksum, mencoba menjawab tantangan ke depan dan geliat perkembangan Mojokerto di sisi utara dengan menghadirkan sejumlah inovasi dan kelengkapan fasilitas sekolah. SMK MUTU Kemlagi juga mencoba menghadirkan pendidikan yang berkualitas yang didukung oleh saran prasarana yang ada. SMK Muhammadiyah 1 Kemlagi merupakan sekolah swasta yang terletak di Jalan Kepuh Sawo, Desa Mojokumpul, Kec. Kemlagi, Mojokerto, Jawa Timur. SMK MUTU Kemlagi saat ini memiliki 4 (empat) jurusan yaitu Farmasi, Teknik Mekanik Industri, Teknik Permesinan dan Teknik Elektronika Industri.

SMK MUTU Kemlagi memiliki visi menjadi sekolah yang islami, kreatif, inovatif dan unggul dalam bidang IPTEK dan IMTAQ. Dalam mencapai visinya SMK MUTU Kemlagi memiliki misi, diantaranya (1) Mendidik generasi muda sesuai tuntutan IMTAQ dan IPTEK; (2) Menyiapkan insan pembangunan yang religius, peka lingkungan; (3) Menyiapkan tenaga industri/usahawan yang mandiri; (4) Mencetak tenaga kerja / usahawan yang mandiri; (5) Menyiapkan lulusan untuk memasuki jenjang pendidikan lebih lanjut/perguruan tinggi. Struktur organisasi SMK MUTU Kemlagi dapat dilihat pada Gambar 2. 1.



Gambar 2. 1 Struktur Organisasi SMK Muhammadiyah 1 Kemlagi

Sumber: (Dokumen SMK Muhammadiyah 1 Kemlagi)

Seluruh kegiatan di SMK MUTU Kemlagi diawasi langsung oleh Majelis Pendidikan Dasar dan Menengah (Dikdasmen) Pimpinan Daerah Muhammadiyah (PDM) Kabupaten Mojokerto, yang membawahi Kepala Sekolah, Institusi Pasangan dan Komite Sekolah, sekolah memiliki dua badan khusus yang meliputi badan tata usaha dan bendahara, dalam pekerjaannya kepala sekolah dibantu oleh Waka Kesiswaan yang membawahi secara langsung Bimbingan Konseling (BK), Waka Kurikulum membawahi Program Teknik Permesinan yang membawahi Kepala Lab Komputer, Waka Humas/DU/DI yang membawahi Program Teknik Elektronika Industri dan Bursa Kerja Khusus SMK, dan Waka Sarana Prasarana yang membawahi Program Farmasi. Masing-masing program/paket keahlian memiliki seorang wali kelas untuk setiap kelas di semua jenjang, wali kelas berkoordinasi dengan guru mata pelajaran yang berinteraksi langsung dengan siswa.

2.2.1 Pelaksanaan Kegiatan Akademik SMK Muhammadiyah 1 Kemlagi

SMK Muhammadiyah 1 Kemlagi melakukan evaluasi tahunan untuk mengkaji pelaksanaan kegiatan akademik saat ini dan memungkinkan perbaikan untuk pelaksanaan tahun ajaran baru. Keseluruhan pelaksanaan kegiatan akademik di SMK Muhammadiyah 1 Kemlagi diatur secara langsung oleh Waka bagian kurikulum dengan berkoordinasi dengan kepala sekolah dan waka bagian lain. Evaluasi siswa dilaksanakan sebanyak 4 (empat) kali tiap tahun ajaran, yaitu pada tengah semester dan akhir semester. Kegiatan praktik kerja industri (prakerin) merupakan kegiatan wajib dilakukan oleh siswa saat menginjak kelas XI (sebelas), siswa diberikan kebebasan dalam memilih tempat prakerin namun dengan batasan-batasan yang ditentukan, apabila siswa tidak memiliki tujuan prakerin maka sekolah akan menyalurkan pada mitra industri sekolah. Setelah melaksanakan kegiatan prakerin siswa diwajibkan untuk mengumpulkan laporan pelaksanaan pada bagian BKK. BKK selanjutnya menyimpan data pelaksanaan prakerin siswa dan melakukan pencarian apakah ada tempat industri yang dapat dijadikan mitra (SMK Muhammadiyah 1 Kemlagi, 2016).

2.3 Sistem Informasi

Menurut Tata Sutabri, sistem informasi merupakan suatu sistem yang ada di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasional dari suatu organisasi yang sifatnya manajerial dengan kegiatan strategi organisasi untuk dapat menyediakan laporan-laporan yang diperlukan pada pihak luar tertentu (Sutabri, 2012).

Secara teknis, sistem informasi didefinisikan sebagai seperangkat komponen yang saling terakit dalam pengambilan, pemrosesan, penyimpanan dan pendistribusian suatu informasi agar dapat mendukung dalam pengambilan keputusan dan pengendalian satuan organisasi. Dapat dibilang sistem informasi merupakan sistem yang dapat mengolah data dengan baik mulai dari data sederhana sampai data yang bersifat kompleks (Pratama, et al., 2018).

2.3.1 Sistem Informasi Akademik

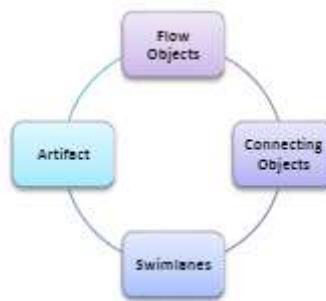
Akademik merupakan ruang atau kondisi yang memungkinkan orang-orang dapat menyampaikan dan menerima gagasan, ilmu pengetahuan dan pemikiran dan dapat menguji hal-hal tersebut secara jujur dan terbuka. Sistem informasi akademik biasa disebut SIAKAD merupakan salah satu sistem informasi yang memiliki fungsi untuk pengelolaan data akademik suatu lembaga pendidikan. Selain itu sistem informasi akademik menyediakan layanan informasi akademik seperti penjadwalan, penilaian siswa (rapor), pembagian kelas siswa, dan hal lain yang berkaitan dengan akademik. Penerapan sistem informasi akademik bertujuan untuk mendukung kinerja sekolah sehingga pelayanan menjadi lebih optimal (Rahmatya & Faris, 2018).

2.4 Proses Bisnis

Proses bisnis adalah sebuah gambaran aktivitas yang berlangsung dalam suatu organisasi. Aktivitas-aktivitas tersebut mencakup inisiasi masukkan (*input*), transformasi informasi, dan hasil keluaran. Hasil keluaran dalam proses bisnis menjadi suatu yang bernilai dalam keberlangsungan organisasi. Dalam pengembangan sistem informasi sebuah proses bisnis organisasi dimodelkan dalam *Business Process Modeling Notation (BPMN)* sebagai acuan desain sistem informasi yang akan dikembangkan (Nurhayati & Setiadi, 2017).

2.4.1 Business Process Model and Notation (BPMN)

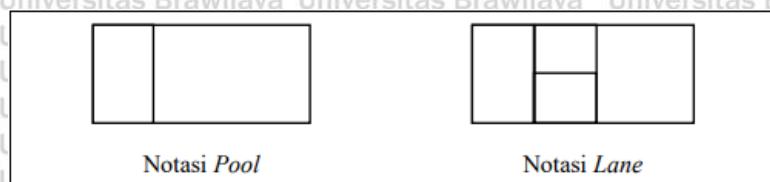
BPMN adalah sebuah teknik pemodelan yang menggunakan berbagai diagram untuk memodelkan sebuah proses bisnis. Penggunaan BPMN dimaksudkan untuk memudahkan pemahaman terhadap sebuah proses bisnis yang kompleks seperti *web services*. BPMN digunakan sebagai standar dalam pemodelan sebuah proses bisnis sebuah sistem informasi (Alam, et al., 2019). BPMN menyediakan notasi dalam bentuk grafis untuk menjelaskan sebuah proses bisnis. Notasi-notasi yang ada dalam BPMN terdapat pada Gambar 2. 2.



Gambar 2. 2 Elemen-Elemen Pada BPMN

Sumber: (Rahmawati, et al., 2017)

BPMN memiliki 4 (empat) elemen yaitu *flow objects*, *artifact*, *connecting objects*, dan *swimlanes* masing-masing elemen tersebut terdiri dari beberapa notasi. Penjabaran dari masing-masing notasi tersebut adalah sebagai berikut (Ismanto, et al., 2020):

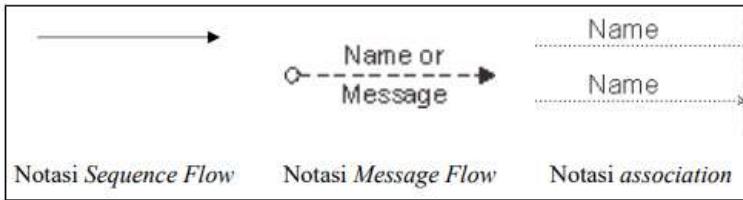


Gambar 2. 3 Notasi-Notasi Pada Swimlane

Sumber: (Ismanto, et al., 2020)

Pool merupakan representasi dari *container* sebuah proses, sedangkan *lane* adalah partisi/bagian dari proses bisnis seperti akses pengguna dalam sistem.

2. *Connecting objects* adalah notasi yang berperan sebagai penghubung antar objek yang ada pada proses bisnis. Masing-masing notasi yang termasuk *connecting objects* adalah *sequence flow*, *message flow*, dan *association*. Notasi-notasi tersebut terdapat pada Gambar 2. 4.

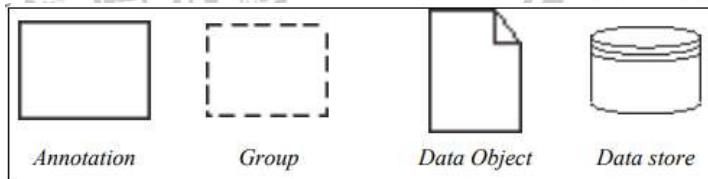


Gambar 2. 4 Notasi-Notasi Pada Connecting Objects

Sumber: (Ismanto, et al., 2020)

Sequence flow berfungsi untuk menghubungkan objek-objek yang ada dalam satu *pool* yang sama. *Message flow* berfungsi sebagai penghubung antar objek yang mengalir antar proses di *pool* yang berbeda, *association* berfungsi sebagai penghubung objek di *artifact*.

3. *Artifacts* adalah informasi tambahan untuk menerangkan sebuah proses. Notasi-notasi yang ada pada *artifacts* adalah *annotation*, *group*, *data object* dan *data store*. Diagram dari masing-masing notasi tersebut terdapat pada Gambar 2. 5.

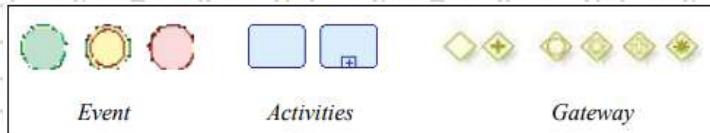


Gambar 2. 5 Notasi-Notasi Pada Artifacts

Sumber: (Ismanto, et al., 2020)

Annotation adalah penjelasan dari sebuah objek dalam *flow*, *group* merupakan pengelompokan dari beberapa objek dalam *flow*, *data object* digunakan apabila sebuah aktivitas menghasilkan sebuah dokumen, dan *data store* adalah aplikasi yang digunakan atau dihasilkan dari suatu aktivitas.

4. *Flow objects* adalah objek-objek yang ada dalam suatu proses. Notasi-notasi yang termasuk dalam *flow objects* adalah sebagai berikut *event*, *activity*, dan *gateway*. Diagram masing-masing notasi tersebut dapat dilihat pada Gambar 2. 6.



Gambar 2.6 Notasi-Notasi Pada *Flow Objects*

Sumber: (Ismanto, et al., 2020)

Events adalah sebuah kejadian yang sifatnya pasif, sedangkan *activity* adalah sebuah kejadian yang sifatnya aktif. *Gateway* digunakan untuk memisahkan aktivitas.

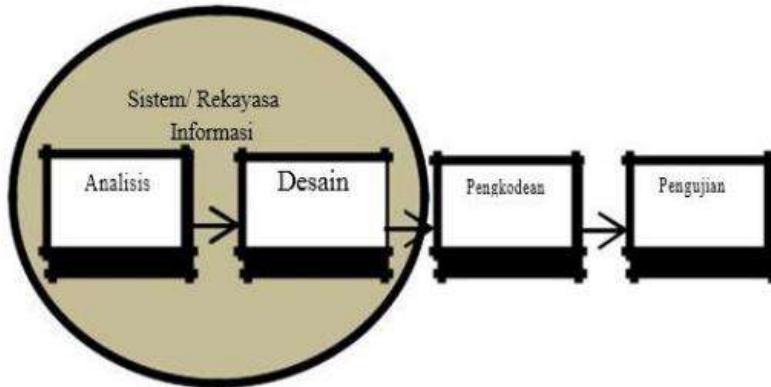
2.5 System Development Life Cycle (SDLC)

Dalam bidang rekayasa perangkat lunak, *System Development Life Cycle* (SDLC) adalah proses pembuatan atau pengubahan suatu sistem dan model serta metodologi yang digunakan untuk mengembangkan suatu sistem, dalam prosesnya SDLC membawahi beberapa metodologi pengembangan yang terdiri dari kerangka kerja untuk perencanaan, kontrol dan pembuatan sistem dalam proses pengembangan perangkat lunak (Roebuck, 2011). Gambaran dari suatu proses dalam perancangan sistem akan selalu bergerak seperti roda yang melewati beberapa tahapan dan akan kembali pada tahap awal apabila sistem yang sudah ada dianggap tidak efisien lagi untuk diterapkan (Abdullah, 2017).

2.6 Waterfall Model

Waterfall model (model air terjun) merupakan paradigma tertua dalam rekayasa perangkat lunak, karena itu *waterfall model* disebut sebagai siklus hidup klasik, yang menyarankan secara sistematis dan pendekatan secara sekuensial dalam pengembangan perangkat lunak. Model *waterfall* dimulai dari spesifikasi kebutuhan dan perkembangan melalui perencanaan, pemodelan, konstruksi dan penyebaran, kulminasi dalam proses dukungan dari perangkat lunak yang selesai (Pressman & Maxim, 2015).

Waterfall model mengambil proses aktivitas fundamental dari spesifikasi, pengembangan, validasi, dan evolusi dan merepresentasikannya dalam proses yang terpisah seperti spesifikasi kebutuhan, desain perangkat lunak, implementasi, pengujian, dan lain-lain (Sommerville, 2011). Karena tahapan-tahapan menurun dari tahap satu ke tahap lain maka disebut model air terjun ilustrasi model ini terdapat pada Gambar 2.7.



Gambar 2.7 **Waterfall Model**

Sumber: (Suryanto, 2016)

Siklus hidup dalam *waterfall model* dijabarkan sebagai berikut:

1. Analisis: tahap analisis dilakukan untuk mengidentifikasi kebutuhan sistem, oleh karena itu tahap ini disebut tahap analisis kebutuhan sistem. Analisis mencakup permasalahan yang ada, proses bisnis saat ini dan usulan perbaikan-perbaikan dari permasalahan.
2. Desain: tahap desain atau perancangan dilakukan untuk mengidentifikasi rancangan komponen-komponen dari sistem informasi yang meliputi: arsitektur sistem, dan diagram yang memodelkan modul-modul fungsionalitas sistem. Hasil dari tahap ini akan menjadi acuan pada tahap implementasi atau pengkodean.
3. Pengkodean: tahap pengkodean merupakan tahap implementasi rancangan sistem informasi menjadi produk sistem informasi yang dapat digunakan.
4. Pengujian: tahap pengujian berfokus pada pengujian dari segi fungsional dan penerimaan pengguna, tahap ini memastikan bahwa semua fungsi sudah diuji dan diperoleh hasil yang sesuai dengan yang ditetapkan.

Tahapan-tahapan proses pada model *waterfall* bersifat sekuensial, sehingga sebuah proses tidak dapat dimulai sebelum proses sebelumnya telah selesai dilakukan. Model air terjun membuat perangkat lunak yang kompleks mudah diatur dan dapat selesai tepat waktu tanpa biaya yang berlebihan. Seiring berjalannya waktu berbagai nama telah diberikan di setiap proses/tahap namun metodologi dasarnya tetap tidak berubah, oleh karena itu tahapan-tahapan pengembangan sistem kadang disebut dengan nama yang berbeda (Simarmata, 2010).

2.7 Unified Modeling Language (UML)

UML adalah bahasa pemodelan standar yang digunakan untuk menspesifikasi, membangun, memvisualisasikan dan mendokumentasikan rancangan dari suatu sistem perangkat lunak (Akil, 2016). Bahasa UML memiliki sintaks dan semantik, sehingga dalam pembuatan model menggunakan konsep UML terdapat aturan-aturan yang harus diikuti sesuai dengan standar yang ada. Blok pembangun UML adalah diagram yang terdiri dari diagram rinci dan diagram yang bersifat umum (Muslihudin & Oktafianto, 2016). UML digunakan dengan tujuan untuk memberikan model yang siap pakai, dengan bahasa pemodelan visual yang ekspresif untuk mengembangkan sebuah sistem dan dapat melakukan pertukaran model dengan mudah, dapat dimengerti secara umum, dan terlepas dari bahasa pemrograman yang dipakai (Waluyo & Munawar, 2017).

2.7.1 Use Case Diagram

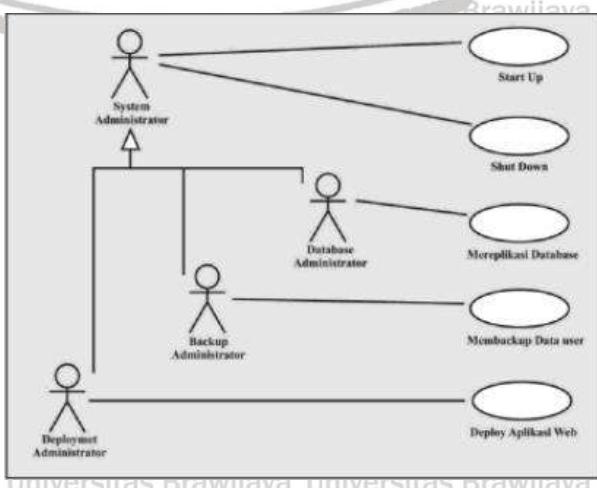
Diagram *use case* merupakan suatu teknik pemodelan terhadap *behavior* sistem yang akan dikembangkan. *Use case* menjabarkan interaksi antara satu atau lebih aktor dengan sistem yang akan dikembangkan (Waluyo & Munawar, 2017). *Use case* berfungsi untuk pemodelan proses bisnis berdasarkan perspektif pengguna, dalam artian *use case* merepresentasikan serangkaian operasi yang dilakukan oleh pengguna (Azis, 2005). Notasi *use case* dijelaskan pada Tabel 2.1, contoh diagram *use case* pada Gambar 2.8.

Tabel 2.1 Notasi Diagram Use Case

| No | Nama | Simbol | Keterangan |
|----|----------|---|---|
| 1 | Aktor |  | Aktor merepresentasikan peran yang diisi oleh pengguna dalam interaksi dengan sistem. Aktor bukan berarti seorang individu atau pekerjaan tertentu. Aktor dapat juga sebagai sebuah organisasi dan sistem lain. |
| 2 | Use case |  | Sebuah abstraksi dari interaksi yang terjadi dalam sistem. |

| No | Nama | Simbol | Keterangan |
|----|-------------|--------|---|
| 3 | Association | — | Hubungan antara aktor dengan <i>use case</i> saat terjadi interaksi antara kedunya. |
| 4 | Include | -----> | Hubungan antara sebuah <i>use case</i> dengan sebuah <i>use case</i> yang bersifat abstrak. |
| 5 | Extends | <----- | Digunakan untuk menyederhanakan sebuah fungsionalitas apabila sebuah <i>use case</i> mengandung fungsionalitas yang kompleks. |
| 6 | Inheritance | —> | Apabila dua atau lebih aktor berbagi perilaku umum yang dimiliki. |
| 7 | Depends On | -----> | Hubungan antar <i>use case</i> yang menegaskan bahwa sebuah <i>use case</i> tidak bisa dilakukan apabila <i>use case</i> lain belum selesai dikerjakan. |

Sumber: (Whitten & Bentley, 2007)



Gambar 2.8 Contoh Diagram *Use case*

Sumber: (Muslihudin & Oktafianto, 2016)

Gambar 2. 8 merupakan ilustrasi interaksi masing-masing aktor dalam aktivitas siklus hidup sistem. Dimungkinkan masing-masing aktor memiliki level pengaksesan masing-masing sesuai dengan proses bisnis yang diidentifikasi. Umumnya aktor yang memiliki hak akses tertinggi adalah administrator, untuk beberapa tujuan aktor administrator hampir diberikan seluruh hak akses terhadap fungsional sistem.

2.7.2 Activity Diagram

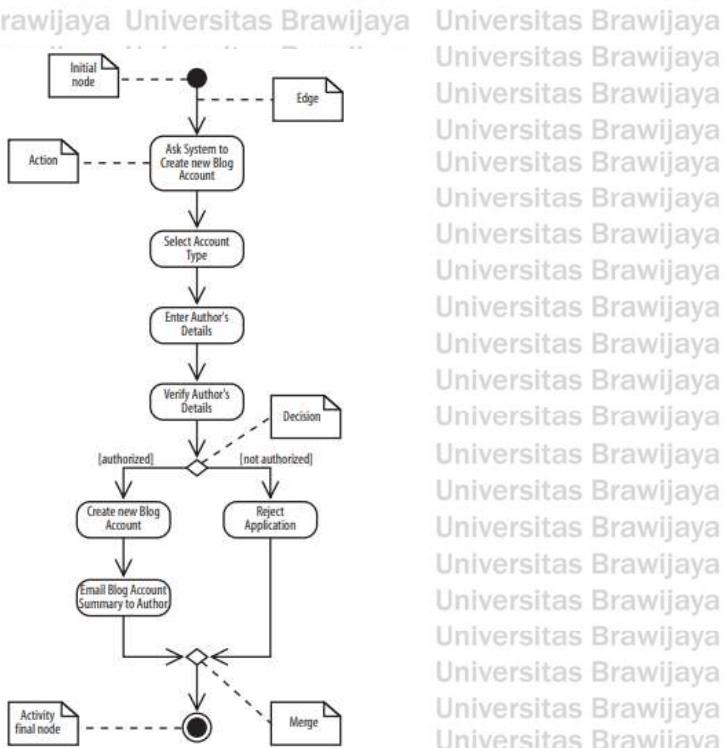
Activity diagram menggambarkan alur kerja atau aktivitas dari sebuah proses bisnis sistem informasi. *Activity diagram* hanya menggambarkan aktivitas antar komponen dalam siklus hidup sistem, sedangkan aktivitas antara aktor dan sistem dalam jangka waktu tertentu didefinisikan dalam *sequence diagram* (Waluyo & Munawar, 2017). Simbol-simbol yang umum digunakan pada *activity diagram* terdapat pada Tabel 2. 2.

Penggunaan *activity diagram* sangat baik dalam memodelkan proses bisnis karena *activity diagram* menunjukkan bagaimana komponen-komponen dalam sistem dapat dikonstruksi sesuai dengan siklus hidup sistem. *Activity diagram* juga merupakan salah satu bagian diagram UML yang mudah dipahami karena memiliki kesamaan simbol dengan diagram alir (Miles & Hamilton, 2006). Contoh *activity diagram* pada Gambar 2. 9.

Tabel 2. 2 Simbol-simbol Pada *Activity Diagram*

| No | Simbol | Nama | Keterangan |
|----|---------------|----------------------------|--|
| | ● | <i>Initial Node</i> | Memulai status awal dalam siklus hidup sistem. |
| | ● | <i>Final activity node</i> | Status akhir siklus hidup sistem. |
| | ○ | <i>Activity</i> | Menggambarkan interaksi masing-masing antarmuka. |
| | ◇ | <i>Decision</i> | Percabangan aktivitas apabila terdapat lebih dari satu pilihan alur. |
| | — | <i>Join</i> | Penggabungan dua atau lebih aktivitas menjadi satu aktivitas. |
| | Swimlane name | <i>Swimlane</i> | Memisahkan proses bisnis sesuai aktivitas yang dilakukan. |

Sumber: (Waluyo & Munawar, 2017)



Gambar 2. 9 Contoh Activity Diagram

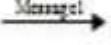
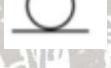
Sumber: (Miles & Hamilton, 2006)

Activity diagram mendefinisikan setiap proses dalam siklus hidup sistem, termasuk alur percabangan yang mungkin diperlukan, atau perulangan yang dilakukan, setiap *activity diagram* dimulai dengan *initial node* sebagai inisiasi permulaan siklus hidup sistem dan selalu diakhiri oleh *activity final node* yang mendefinisikan akhir dari siklus hidup sistem.

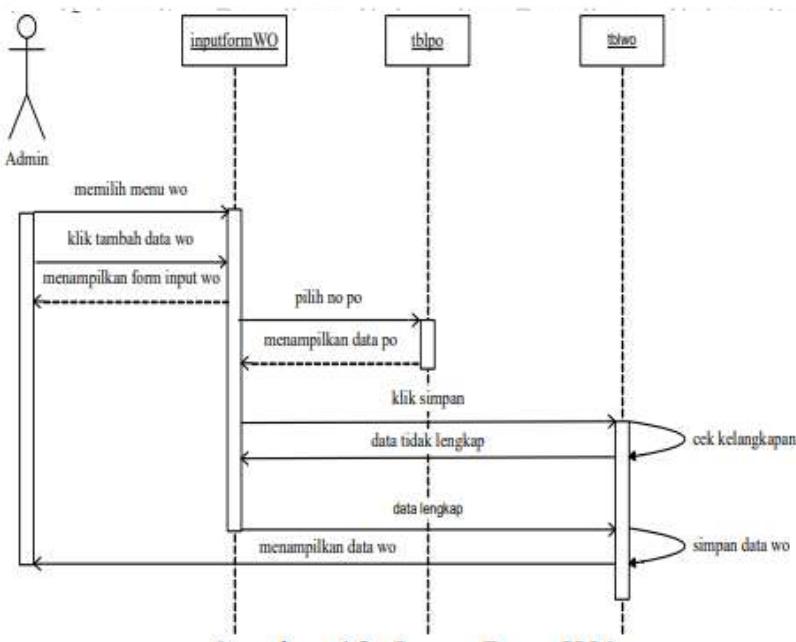
2.7.3 Sequence Diagram

Sequence diagram digunakan untuk memodelkan interaksi antar objek dalam waktu tertentu dalam siklus hidup sistem. Keberadaan *sequence diagram* menjadi penting untuk memodelkan bagaimana interaksi sebuah aktor dengan sistem dalam waktu tertentu dengan kronologi tertentu. Sehingga kita dapat mendeskripsikan sebuah interaksi yang harus dijalankan apabila sebuah *use case* dieksekusi. Notasi *sequence diagram* dapat dilihat pada **Error! Reference source not found..** Contoh *sequence diagram* ada pada Gambar 2. 10.

Tabel 2. 3 Simbol-Simbol Pada Sequence Diagram

| No | Nama | Simbol | Keterangan |
|----|-------------------|---|---|
| 1 | <i>Lifeline</i> | | Merupakan indikasi sebuah <i>object</i> dalam waktu tertentu dalam siklus hidup sistem. |
| 2 | <i>Activation</i> | | Merupakan sebuah indikasi sebuah aksi yang akan dipicu oleh <i>object</i> . |
| 3 | <i>Message</i> | Uni  | Merupakan komponen untuk mengindikasikan komunikasi antar <i>object</i> . |
| 4 | <i>Boundary</i> |  | Merupakan komponen untuk mendefinisikan interaksi antara aktor dengan sistem, dan merepresentasikan hasilnya pada antarmuka. |
| 5 | <i>Controller</i> |  | Merupakan komponen untuk merepresentasikan koordinasi antar <i>object</i> . |
| 6 | <i>Entity</i> |  | Merupakan komponen untuk memodelkan <i>object</i> terhadap pengeloaan data. |
| 7 | <i>Object</i> |  | Merupakan <i>instance</i> dari sebuah <i>class</i> , digambarkan sebagai <i>class</i> dengan nama objek yang ada dalam <i>class</i> tersebut. |
| 8 | Aktor |  | Merupakan komponen yang melakukan interaksi dengan sistem, tidak berbeda dengan aktor pada <i>use case diagram</i> . |

Sumber: (Waluyo & Munawar, 2017)



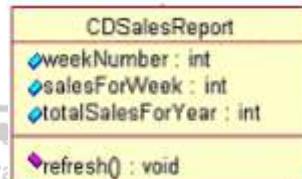
Gambar 2. 10 Contoh Sequence Diagram

Sumber: (Waluyo & Munawar, 2017)

Sequence diagram memodelkan bagaimana urutan proses sebuah fungsi saat terjadi interaksi antara aktor dengan sistem, proses terjadi tidak selalu linear namun dimungkinkan terjadi siklus bolak-balik antar fungsi satu dengan fungsi lain bergantung proses bisnis sistem.

2.7.4 Class Diagram

Class diagram merupakan notasi mendasar dalam diagram UML, class diagram digunakan untuk merepresentasikan class dengan atribut dan fungsi didalamnya. Contoh diagram *class* dapat dilihat pada Gambar 2. 11. *Class diagram* memodelkan struktur sistem dari segi pendefinisian kelas-kelas pembangun sistem (Waluyo & Munawar, 2017).



Gambar 2. 11 Class Diagram Tunggal

Dalam sistem informasi kompleks yang dibangun menggunakan bahasa pemrograman berorientasi objek, sebuah kelas memiliki relasi dengan kelas lain untuk membangun satu kesatuan sistem. Relasi antar *class* memiliki kekuatan yang berbeda bergantung pada ketergantungan suatu *class* terhadap *class* lain.

Dalam *class diagram* terdapat 5 (lima) jenis relasi yang digambarkan pada Gambar 2. 12 (Miles & Hamilton, 2006).



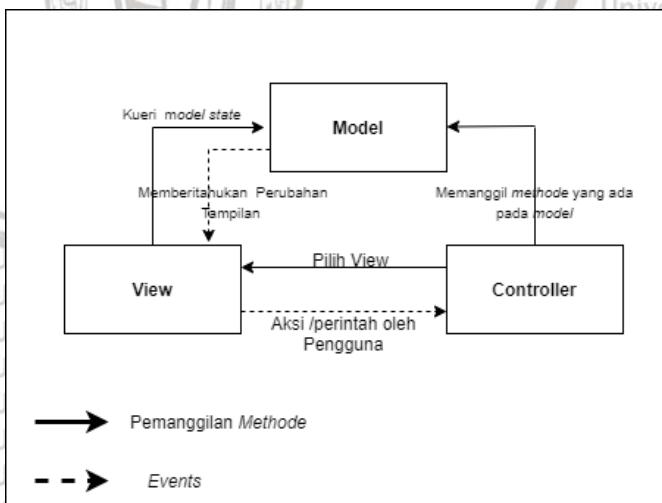
Gambar 2. 12 Tipe Relasi Pada *Class Diagram*

Sumber: (Miles & Hamilton, 2006)

- Dependecy:** terjadi saat suatu *class* bekerja dengan intruksi dari objek di *class* lain.
- Association:** Terjadi saat suatu *class* bekerja dengan *class* lain untuk menjalankan fungsi tertentu.
- Aggregation:** Terjadi saat suatu *class* menggunakan suatu fungsi atau atribut namun *class* tersebut membagikan atribut atau fungsi tersebut dengan *class* lain.
- Composition:** Apabila suatu *class* mengandung objek yang diperoleh dari *class* lain.
- Inheritance:** Apabila suatu *class* merupakan tipe data dari *class* lain.

2.8 Models Views Controllers (MVC)

MVC adalah sebuah arsitektur dalam pembuatan aplikasi yang memisahkan bagian-bagian sistem seperti data, presentasi dan flow aplikasi dalam file terpisah. Tujuan penerapan MVC adalah untuk mempermudah pengorganisasian kode sehingga lebih mudah dalam proses *maintenance*. Hubungan dalam MVC diilustrasikan pada Gambar 2. 13.



Gambar 2. 13 Hubungan *Models*, *Views*, dan *Controllers*.

Sumber: (Subari, et al., 2018)

Sesuai dengan namanya MVC terdiri dari 3 (tiga) bagian yang meliputi, (1) *Models*: berfungsi untuk mengatur data dari aplikasi dan membantu untuk pelaksanaan proses bisnis yang diperlukan oleh aplikasi; (2) *Views*: sebuah berkas untuk membuat tampilan informasi kepada *user*, berkas *views* tidak berisi logika namun hanya berfokus pada tampilan; (3) *Controllers*: sebuah berkas untuk mengatur pertukaran data antara *models* dengan *views* (Codeigniter Foundation, 2021).

2.9 WEB

Web adalah sebuah berkas teks yang didalamnya berisi *tag-tag* dengan format *hyper text markup language (HTML)*. Tag-tag tersebut dapat merujuk melalui tautan ke dokumen-dokumen lain seperti gambar, suara, dan teks biasa. Berkas-berkas tersebut disimpan dalam sebuah tempat yang disebut dengan *server* yang dapat diakses oleh *user* menggunakan peramban web (Wahana Komputer, 2000). Dalam sistem web terdapat kumpulan informasi/kumpulan halaman yang dapat diakses melalui jaringan internet. Sehingga informasi yang ada dapat diakses oleh setiap orang selama terhubung dengan jaringan internet (Junirianto, 2018).

Web pertama kali diusulkan oleh Tim Berners-lee melalui proposal yang ia tujuhan pada Organisasi Eropa untuk Riset Nuklir (CERN) pada tahun 1989, proposal tersebut membahas mengenai sebuah sistem informasi yang akan membantu mengorganisir informasi yang tersebar pada berbagai laboratorium. Kini web digunakan secara luas pada berbagai sektor dalam pembagian informasi (Brugger & Milligan, 2018).

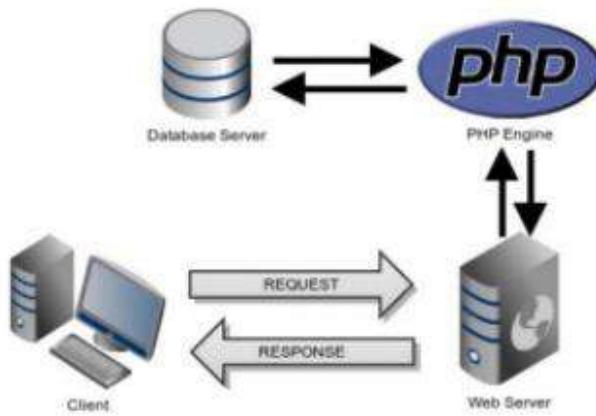
2.9.1 Komponen Sistem Web

Dalam membangun sebuah web, terdapat 3 (tiga) komponen utama yang memiliki peranan masing-masing terhadap kerja suatu sistem berbasis web, Komponen-komponen tersebut adalah sebagai berikut (Nixon, 2018):

- 1) HTML5: *Hyper text markup language (HTML)* generasi ke-5 yang merupakan versi paling baru dalam HTML, pada HTML5 terdapat dukungan multimedia seperti audio dan video, selain itu juga terdapat dukungan grafis yang memungkinkan pengguna menggambar objek melalui peramban web.
- 2) CSS3: *Cascading style sheet (CSS)* generasi ketiga merupakan pasangan yang krusial bagi HTML, CSS dapat digunakan untuk mengatur sebuah gaya pada elemen HTML, seperti ukuran dimensi, warna, spasi, dan lain-lain.
- 3) Javascript: Javascript memungkinkan untuk memberikan akses *scripting* pada elemen HTML, sehingga elemen HTML dapat dilakukan manipulasi seperti perubahan perilaku saat elemen diklik atau saat sebuah elemen disorot.

2.9.2 Web Server

Web server adalah perangkat lunak yang bertugas untuk menerima permintaan dari pengguna melalui peramban web dan mengirimkan hasil dari permintaan tersebut dalam bentuk halaman-halaman web. Selain itu web server juga bertugas untuk menyediakan sebuah sumber daya yang dibutuhkan oleh pengguna, seperti gambar, audio, dan video. Alur kerja dari web server diilustrasikan pada Gambar 2. 14.



Gambar 2. 14 Arsitektur Aplikasi Web Dinamis

Sumber: (Solichin, 2016)

Sebuah sistem web yang dinamis memerlukan *web server* dan *database*, data akan dikirim ke pengguna dalam format HTML, selain menerima data pengguna juga dapat mengirim data ke *web server* yang akan disimpan dalam *database*, hal tersebut diatur oleh *PHP Engine* (Solichin, 2016).

2.9.3 PHP (*PHP Hypertext Preprocessor*)

PHP Hypertext Preprocessor (dituliskan secara rekursif) adalah bahasa pemrograman yang berjalan pada sisi server (*sisi server*) yang populer digunakan pada pengembangan sebuah sistem berbasis web. PHP memungkinkan untuk melakukan berbagai akses dan control terhadap *web server* dan *database* seperti manipulasi berkas HTML, dan merubah data pada *database* (Nixon, 2018).

PHP memungkinkan pengembangan sistem berbasis web menjadi lebih mudah, hal ini dikarenakan PHP menyediakan fungsi bawaan yang diperlukan oleh pengembang dalam mengembangkan sistem. PHP juga menyediakan dukungan terhadap banyak DBMS, dan PHP bersifat sumber terbuka (Davis & Phillips, 2006).

2.9.4 Peramban Web (*Web Browser*)

Peramban web adalah perangkat lunak yang digunakan untuk mengambil dan menyajikan sumber daya dari *web server*, sumber daya web dapat berupa teks, gambar, suara, dan video (Solichin, 2016). Dalam penelitian kali ini peramban web yang digunakan dalam pengujian adalah sebagai berikut:

A. Google Chrome

Google Chrome merupakan peramban web yang dikembangkan oleh Google melalui proyek sumber terbuka Chromium. Google Chrome memiliki berbagai fitur seperti dapat membuka hingga 100 tab, tersedia di berbagai platform dan perangkat, sinkronisasi data, dan pengaya peramban. Google Chrome memiliki keamanan bawaan sehingga dapat melindungi pengguna dari situs berbahaya (Google Inc, 2021).

B. Mozilla Firefox

Firefox merupakan peramban web sumber terbuka yang dikembangkan oleh Mozilla pada tahun 1998, sebagai alternatif peramban yang ada, Firefox berfokus pada hak pengguna atas privasi data. Firefox memiliki berbagai fitur seperti perlindungan privasi, sinkronisasi antar perangkat, ubahsuai peramban, perlindungan perlakuan yang ditingkatkan, gambar dalam gambar serta tersedia antar platform dan perangkat (Mozilla Corporation, 2021).

2.10 Database Management System (DBMS)

Secara umum *database* atau basis data dapat diartikan sebagai tempat menyimpan sebuah data, basis data digunakan sebagai pengganti dari sistem penyimpanan berkas dokumen yang bersifat konvensional. Database terdiri dari kumpulan data yang terhubung untuk memenuhi kebutuhan suatu organisasi (Sucipto, et al., 2017). Struktur database terbagi menjadi 2 (dua), *database flat* dan database relasional, database relasional menggunakan tabel-tabel dalam penyimpanan data, pada setiap tabel terdapat kolom dan baris yang mendefinisikan data dan informasi yang disimpan (Wahana Komputer, 2010). Dalam database relasional terdapat komponen-komponen sebagai berikut (Setyawati, et al., 2020):

- A. *Table*: tempat untuk menyimpan data atau *record* dari data, data yang disimpan memiliki informasi yang sejenis.
- B. *Record*: Sebuah entri tunggal yang ada dalam tabel, entri terdiri dari sejumlah *field* data.
- C. *Index*: Tipe tabel yang berisi *field* khusus dan disimpan dalam urutan tertentu.
- D. *Field*: Item tertentu dalam record data yang menyimpan informasi mengenai atribut data seperti panjang maksimum.
- E. *Query*: Perintah SQL untuk pemanggilan dan operasi pada tabel.

Database Management System (DBMS) menyimpan data yang saling berelasi dan satu set program untuk mengakses data tersebut. Database dalam DBMS berisi kumpulan berkas yang berelasi, relasi tersebut diidentifikasi

dengan kunci dari setiap berkas. Sedangkan *Set program* pada DBMS bertugas untuk mengelola data pada basis data, seperti penambahan, penghapusan, pengubahan dan pembacaan data (Kristanto, 1994).

2.10.1 SQL

Structured Query Language (SQL) merupakan sebuah bahasa untuk melakukan kueri pada suatu database relasional. SQL merupakan bahasa formal yang memiliki sintaks yang didefinisikan, dengan SQL pengembang dapat melakukan manipulasi data pada database. Database melakukan eksekusi terhadap skrip SQL dan melakukan tugas yang dispesifikasi dalam skrip tersebut dan menampilkan hasilnya (Fehily, 2008).

2.10.2 MySQL

MySQL adalah sebuah perangkat lunak RDBMS (*Relational Database Management System*) yang dirintis oleh Michael Widenius, MySQL dapat menangani data bervolume besar tanpa menggunakan sumber daya komputasi yang besar, dapat melakukan pemrosesan data dengan sangat cepat, dan mendukung multi pengguna (Wahana Komputer, 2010). Dalam aplikasi web dinamis MySQL digunakan untuk mengolah data dalam database relasional, MySQL memiliki dukungan yang sangat baik dengan bahasa pemrograman PHP (Nixon, 2018).

2.10.3 Normalisasi Database

Database yang menyimpan banyak data perlu dilakukan normalisasi sehingga tidak terjadi anomali data. Normalisasi adalah proses penyesuaian struktur *database*, sehingga ambiguitas data dapat dihindari dan dihilangkan. Tahap normalisasi terdiri dari tahap paling ringan (1NF) sampai paling rapat (5NF), *database* dapat dikatakan sudah mencukupi biasanya hanya sampai pada tahap 3NF. Penjelasan bentuk-bentuk normalisasi dari 1NF sampai 3NF adalah sebagai berikut (Suryadi, 2019):

1. Bentuk Normal Tahap Pertama(1NF) adalah bentuk yang paling ringan dalam normalisasi database, ciri-ciri bentuk normal 1NF adalah sebagai berikut:

- Sebuah tabel tidak memiliki atribut banyak nilai atau nilai dalam suatu tabel harus tunggal.
- Tabel tidak boleh memiliki atribut turunan.
- Tabel tidak memiliki *record* bernilai ganda.
- Atribut dalam tabel harus bersifat *atomic* (tidak dapat dibagi-bagi).

2. Bentuk Normal Tahap Kedua(2NF) adalah bentuk menengah ke bawah dalam normalisasi database, ciri-ciri bentuk normal 2NF adalah sebagai berikut:

- Telah memenuhi bentuk normalisasi 1NF.
 - Tidak ada atribut yang memiliki ketergantungan secara parsial.
 - Tidak ada atribut yang tidak memiliki ketergantungan pada *primary key*.
3. Bentuk Normal Tahap Ketiga(3NF) adalah bentuk menengah keatas dalam normalisasi database, ciri-ciri bentuk normal 3NF adalah apabila bentuk normal 2NF telah terpenuhi dan tidak ada tidak ada atribut *non-primary key* yang memiliki ketergantungan dengan atribut *non-primary key* yang lain.

2.11 Framework

Framework atau kerangka kerja adalah kumpulan kode untuk memudahkan para *programmer* dalam pembuatan aplikasi, didalam sebuah *framework* terdapat berbagai fungsionalitas, seperti pengaya (*plugin*) dan abstraksi konsep untuk membentuk sebuah sistem yang terstruktur dan rapi. Penggunaan *framework* bukan berarti membebaskan seorang *programmer* dengan pengkodean, dengan menggunakan *framework*, *programmer* dapat menggunakan berbagai fungsionalitas yang telah disediakan oleh *framework* (Junirianto, 2018). Dalam penelitian ini *framework* yang digunakan adalah sebagai berikut:

2.11.1 Codeigniter

Codeigniter (CI) merupakan salah satu *framework* bahasa pemrograman PHP yang populer, karena sifatnya yang sumber terbuka (*open source*), memiliki fungsionalitas yang kaya akan pustaka dan bantuan, sehingga pengembangan aplikasi berbasis web akan lebih cepat. Codeigniter memiliki dukungan penuh pada bahasa pemrograman PHP versi 5.2.6 keatas, selain itu CI juga mendukung berbagai DBMS seperti MongoDB, MySql, dan PostgresSQL. CI juga menyediakan fungsionalitas untuk keamanan dari berbagai macam serangan terhadap sistem (Wadi, 2020).

2.11.2 Bootstrap

Bootstrap merupakan sebuah *framework* CSS yang didesain dan dikembangkan oleh pengembang media sosial Twitter, Bootstrap adalah salah satu *framework* CSS yang populer, karena bersifat sumber terbuka dan menyediakan berbagai macam utilitas yang berguna untuk membangun tampilan antarmuka. Bootstrap diluncurkan pada 19 Agustus 2011 dan telah dilakukan rilis sebanyak 20 (dua puluh) kali dengan perubahan besar pada versi 2 (dua) dan versi (3). Bootstrap menyediakan fungsionalitas desain web responsive sehingga tampilan web akan menyesuaikan dengan ukuran layar pengguna (Bootstrap Core Team, 2021).

2.12 Skala Likert

Skala Likert diciptakan Rensis Likert pada tahun 1932. Skala Likert merupakan sebuah teknik untuk melakukan pengukuran skala ordinal yang mengukur sebuah persepsi seseorang terhadap suatu hal atau peristiwa. Skala Likert terdiri atas 5 (lima) skala yang memberikan nilai terbesar untuk jawaban paling positif dan nilai terkecil untuk jawaban paling negatif, rincian penilaian skala Likert dapat dilihat pada Tabel 2. 4(Saputra, et al., 2018):

Tabel 2. 4 Skala Likert

| No | Persepsi | Nilai/Poin |
|----|---------------------|------------|
| 1 | Sangat Setuju | 5 (Lima) |
| 2 | Setuju | 4 (Empat) |
| 3 | Netral | 3 (Tiga) |
| 4 | Tidak setuju | 2 (Dua) |
| 5 | Sangat tidak setuju | 1 (Satu) |

2.13 Pengujian

Pengujian dilakukan untuk memastikan sebuah program telah memenuhi spesifikasi yang telah ditetapkan dan menghindari sebuah cacat pada program sebelum disebarluaskan pada pengguna, pada tahap pengujian pengembang menggunakan data buatan dan melakukan pengecekan terhadap keluaran dari program yang dilakukan. Tujuan utama pengujian perangkat lunak adalah sebagai berikut (Sommerville, 2011):

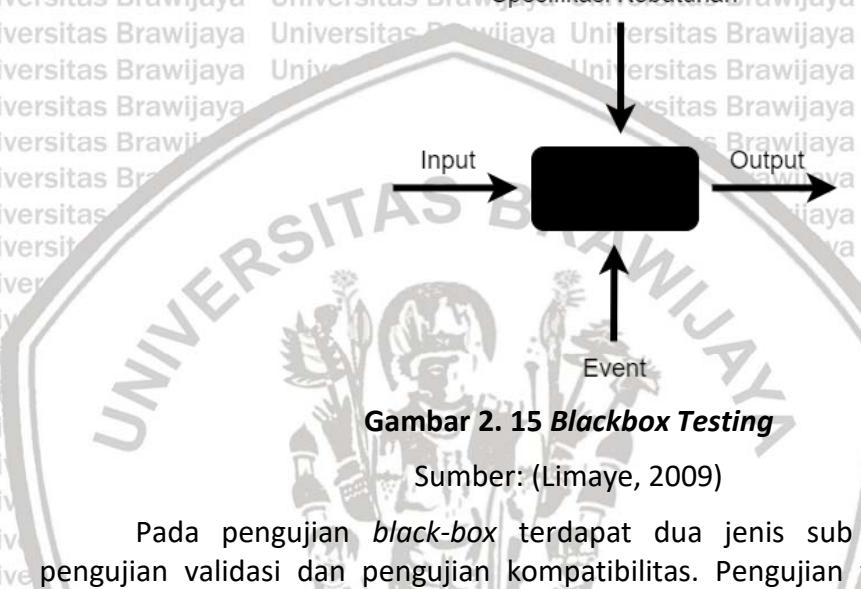
1. *Validation Testing*: berguna untuk memberikan gambaran untuk pengembang dan pengguna bahwa program telah memenuhi kebutuhan yang telah dispesifikasikan sebelumnya.
2. *Defect Testing*: berguna untuk menemukan situasi dimana program tidak berjalan sebagaimana yang telah dispesifikasikan sebelumnya.

Pengujian adalah aktivitas yang perlu dilakukan secara terencana, sehingga tahap pengujian memiliki *road map* yang mendeskripsikan bagian-bagian yang perlu diuji, kapan pengujian perlu dilakukan, dan berapa banyak sumber daya yang diperlukan, desain kasus, eksekusi, dan koleksi hasil tes dan evaluasinya (Pressman & Maxim, 2015).

2.13.1 Blackbox Testing

Blackbox testing merupakan pengujian terhadap kebutuhan fungsionalitas program yang menguji hasil *output* dari suatu *input* apabila dilakukan sebuah proses atau *event* ilustrasi *blackbox testing* dapat dilihat pada Gambar 2. 15.

- Blackbox testing* dilakukan untuk menemukan *error* dengan kategori berikut (Sommerville, 2011):
1. Fungsi yang salah atau tidak ditemukan.
 2. Error pada antarmuka.
 3. Kesalahan pada akses database atau struktur data.
 4. Kesalahan pada perilaku atau performa sistem.
 5. Kesalahan pada inisialisasi dan terminasi sistem.



Gambar 2. 15 *Blackbox Testing*

Sumber: (Limaye, 2009)

Pada pengujian *black-box* terdapat dua jenis sub pengujian, yaitu pengujian validasi dan pengujian kompatibilitas. Pengujian validasi dilakukan dengan melakukan demonstrasi fungsi lalu memeriksa hasil keluaran suatu fungsi dalam program dengan spesifikasi yang telah ditentukan. Pengujian kompatibilitas lebih berfokus pada konsistensi sebuah *output* program apabila dijalankan pada kondisi dan *platform* yang berbeda, misalnya sebuah sistem berbasis web harus menampilkan tampilan antarmuka yang konsisten atau berbeda apabila dijalankan pada perangkat yang berbeda, atau sebuah fungsi pada program apakah tetap konsisten menampilkan *output* yang sama meski dijalankan pada *runtime* yang berbeda (Pressman & Maxim, 2015).

2.13.2 User Acceptance Test

User Acceptance Test (UAT) dilakukan oleh pengguna akhir (*end user*) dengan keadaan yang nyata dan data yang nyata, pengujian dengan melibatkan pengguna akhir secara langsung untuk menghindari miskomunikasi antara kebutuhan pengguna dengan sistem yang telah dibuat (Otaduy & Diaz, 2017). UAT menggunakan skala Likert dalam menentukan persepsi pengguna. Item yang diukur dalam UAT sistem meliputi konten, modul, multimedia, navigasi dan kegunaan (Nuraini, et al., 2016). Contoh instrumen UAT pada Gambar 2. 16.

| Item | Frequency | | | | | |
|--|---------------------------|---------------------------|---------------|-------------|------------------------|---------------------|
| | 1 Strongly Disagree | 2 Somewhat Disagree | 3 Disagree | 4 Agree | 5 Somewhat Agree | 6 Strongly Agree |
| Content | | | | | | |
| The content is clear | | | | 3 (10 %) | 19 (63.3 %) | 8 (26.7 %) |
| The content is easy to understand | | | | 6 (20 %) | 19 (63.3 %) | 5 (16.7 %) |
| The content is related to Nutrition topic | | | | | 15 (50 %) | 15 (50 %) |
| The content in PLENut is interesting | 1 (3.3 %) | | 4 (13.3 %) | 15 (50 %) | 10 (33.3 %) | |
| Module | | | | | | |
| Introduction | | | | 8 (26.7 %) | 13 (43.3 %) | 9 (30 %) |
| Notes | 1 (3.3 %) | | 2 (6.7 %) | 16 (53.3 %) | 11 (36.7 %) | |
| Learning Styles | | | 3 (10 %) | 14 (46.7 %) | 13 (43.3 %) | |
| Exploration | | | 7 (23.3 %) | 10 (33.3 %) | 13 (43.3 %) | |
| PLE Elements | | | 4 (13.3 %) | 14 (46.7 %) | 12 (40 %) | |
| Forum | | | 5 (16.7 %) | 14 (46.7 %) | 11 (36.7 %) | |
| Glossary | | | 7 (23.3 %) | 8 (26.7 %) | 15 (50 %) | |
| Multimedia Element | | | | | | |
| Appropriate font type | | | | 4 (13.3 %) | 11 (36.7 %) | 15 (50 %) |
| Appropriate font size | | | | 2 (6.7 %) | 12 (40 %) | 16 (53.3 %) |
| Appropriate graphics | | | | 3 (10 %) | 7 (23.3 %) | 20 (66.7 %) |
| Appropriate button | | | | 4 (13.3 %) | 10 (33.3 %) | 16 (53.3 %) |
| Appropriate colour | | | | 3 (10 %) | 8 (26.7 %) | 19 (63.3 %) |
| Appropriate audio | 3 (10 %) | | 7 (23.3 %) | 12 (40 %) | 8 (26.7 %) | |
| Navigation | | | | | | |
| Navigation is easy | | | | 11 (36.7 %) | 10 (33.3 %) | 9 (30 %) |
| Navigation is clear and concise | | | | 11 (36.7 %) | 10 (33.3 %) | 9 (30 %) |
| Number of buttons / links reasonable | | | | 8 (26.7 %) | 18 (60 %) | 4 (13.3 %) |
| Links are consistent | | | | 9 (30 %) | 11 (36.7 %) | 10 (33.3 %) |
| Links are easy to access | 10 (33. %) | | 9 (30 %) | 10 (33.3 %) | 1 (3.3 %) | |
| Usefulness | | | | | | |
| PLENut is useful for Visual students (Picture) | | | | 3 (10 %) | 6 (20 %) | 21 (70 %) |
| PLENut is useful for Auditory students (Sound) | 1 (3.3 %) | | 10 (33.3 %) | 4 (13.3 %) | 15 (50 %) | |
| PLENut is useful for Kinaesthetic students (Touch) | | | | 6 (20 %) | 6 (20 %) | 18 (60 %) |

Gambar 2. 16 User Acceptance Test

Sumber: (Nuraini, et al., 2016)

Hasil dari pengukuran menggunakan skala dilakukan konversi dalam bentuk persentase yang dapat dipetakan sebagai berikut:

1. Sangat tidak dapat diterima: $0\% < \text{persentase yang diperoleh} \leq 20\%$.
2. Tidak dapat diterima: $20\% < \text{persentase yang diperoleh} \leq 40\%$.
3. Netral: $40\% < \text{persentase yang diperoleh} \leq 60\%$.
4. Dapat diterima: $60\% < \text{persentase yang diperoleh} \leq 80\%$.
5. Sangat dapat diterima: $80\% < \text{persentase yang diperoleh} \leq 100\%$.

Pada penelitian ini digunakan penyusunan instrument penerimaan pengguna menggunakan *technology acceptance model* (TAM). *Technology Acceptance Model* menerapkan teori sifat manusia dalam penggunaan teknologi. Asumsi pada TAM adalah saat pengguna menggunakan produk teknologi akan dipengaruhi oleh beberapa faktor atau domain yang mempengaruhinya. Faktor-faktor tersebut meliputi persepsi kebermanfaatan (*Perceived Usefulness*), persepsi kemudahan penggunaan (*Perceived Ease of Use*), sikap terhadap penggunaan (*Attitude Toward Using*) dan intensitas perilaku penggunaan (*Behavioural Intention*) yang mana perilaku pada empat domain tersebut dapat

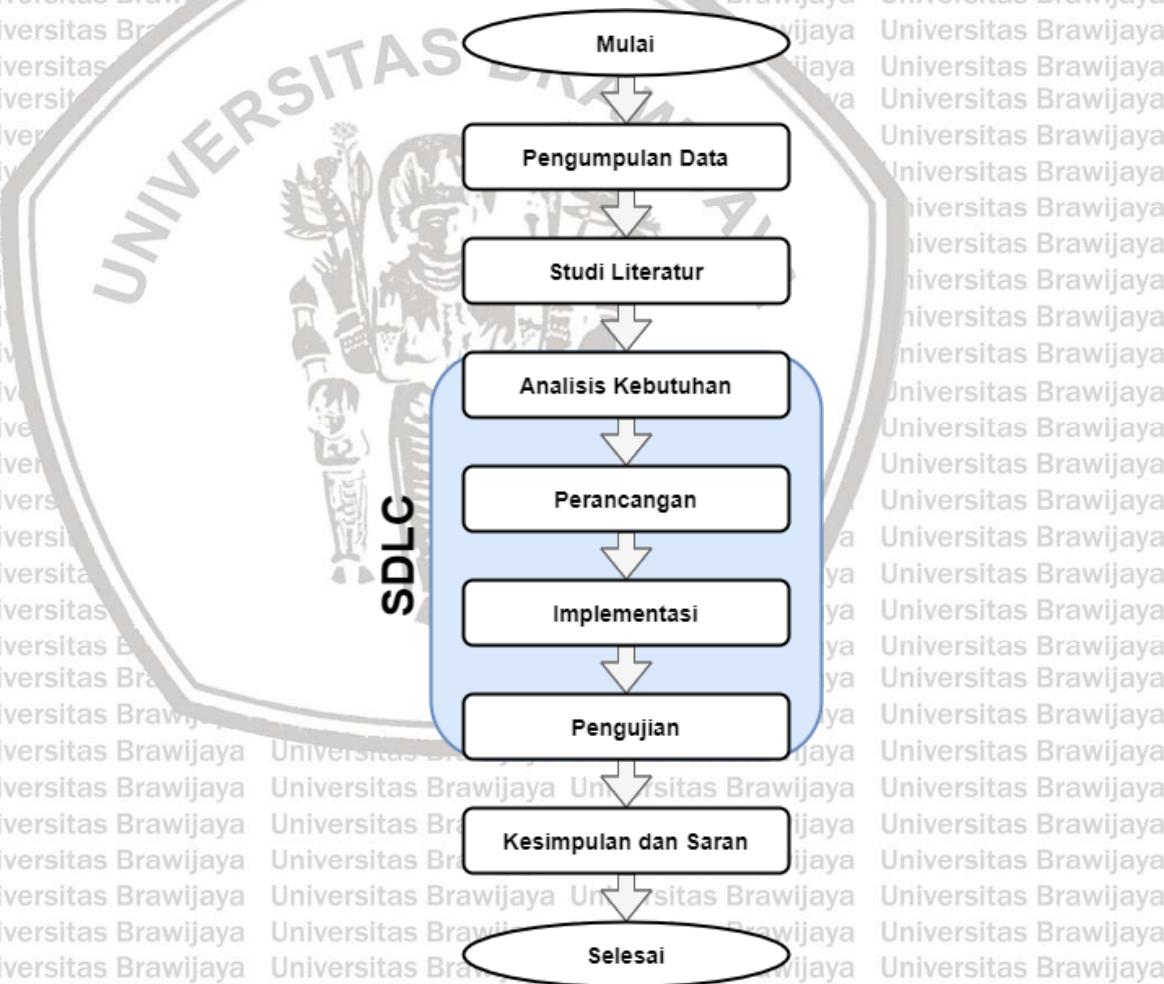
merepresentasikan penggunaan sistem secara aktual. Penjelasan masing-masing domain adalah sebagai berikut: (Fatmawati, 2015)

1. Persepsi Kebermanfaatan (*Perceived Usefulness*) adalah kumpulan pernyataan pengguna terhadap kegunaan sistem informasi. Indikator pada domain ini antara lain: mempercepat pekerjaan pengguna, membuat pekerjaan pengguna menjadi lebih efektif dan efisien, membuat pengguna menjadi lebih produktif dalam pekerjaan dan memberikan manfaat untuk pengguna.
2. Persepsi Kemudahan Penggunaan Sistem Informasi (*Perceived Ease of Use*) adalah kumpulan pernyataan pengguna terhadap kemudahan atau kesulitan penggunaan sistem informasi. Indikator pada domain ini antara lain: kemudahan untuk dipelajari pengguna, adaptasi yang mudah bagi pengguna, kemudahan menjadi ahli dan terampil dalam waktu singkat, dan kemudahan pengguna menjadi terampil dalam waktu singkat.
3. Sikap/Pandangan Terhadap Penggunaan Sistem Informasi (*Attitude Toward Using*) adalah kumpulan pernyataan pengguna terhadap sistem informasi baik berupa penolakan atau penerimaan dalam sikap pengguna. Indikator dalam domain ini adalah: kesukaan pengguna terhadap sistem informasi, pendapat pengguna terhadap penerapan sistem informasi dan kesenangan dalam penyelesaian pekerjaan menggunakan sistem informasi.
4. Minat Terhadap Penggunaan Sistem Informasi (*Behavioural Intention*) adalah kumpulan pernyataan pengguna terhadap minat penggunaan sistem informasi. Indikator dalam domain ini antara lain: Kesukaan pengguna terhadap sistem informasi apabila terdapat pilihan lain, minat pengguna terhadap sistem informasi apabila terdapat pilihan lain, penggunaan sistem informasi apabila terdapat pilihan lain, dan rekomendasi penggunaan sistem informasi pada orang lain.

BAB 3 METODOLOGI

Penelitian dilaksanakan di SMK Muhammadiyah 1 Kemlagi yang beralamatkan di Jalan Kepuh Sawo, Desa Mojokumpul, Kecamatan Kemlagi, Kabupaten Mojokerto, Jawa Timur. SMK Muhammadiyah 1 Kemlagi saat penelitian dilakukan masih mengalami kendala dalam pengelolaan data akademik sekolah, karena data yang dimasukkan belum terintegrasi dengan basis data sekolah. Penelitian direncanakan berlangsung pada bulan Februari sampai bulan Juni 2021.

Penelitian ini merupakan penelitian yang bertopik pengembangan sistem informasi, penulis mengembangkan sistem informasi akademik sekolah berbasis web untuk SMK Muhammadiyah 1 Kemlagi, dalam *SDLC* model yang digunakan adalah model *waterfall*, yang terdiri dari analisis kebutuhan, perancangan, implementasi dan pengujian.



Gambar 3. 1 Metodologi Penelitian

Berikut merupakan penjabaran lebih lanjut mengenai diagram alir pada Gambar 3. 1:

3.1 Pengumpulan Data

Tahap pengumpulan data berfungsi untuk mendapatkan data yang *real* yang berguna untuk mendukung penelitian yang dilakukan oleh penulis. Pada tahap ini penulis melakukan diskusi dengan pemangku kepentingan di SMK Muhammadiyah 1 Kemlagi, dalam hal ini adalah Waka Kesiswaan dan Kepala BKK/TU, *stakeholder* tersebut dipilih karena memiliki pengetahuan yang lebih mendalam terhadap permasalahan kegiatan akademik sekolah. Instrumen yang digunakan dalam pengumpulan data adalah wawancara, dengan kegiatan wawancara penulis bisa mendapatkan gambaran permasalahan yang terjadi pada SMK Muhammadiyah 1 Kemlagi. Penulis juga melakukan pengumpulan dokumen-dokumen mengenai SMK Muhammadiyah 1 Kemlagi untuk ditelaah lebih lanjut. Data dan informasi yang telah diperoleh pada tahap ini selanjutnya akan menjadi dasar dalam proses studi literatur.

3.2 Studi Literatur

Studi literatur berguna untuk mendapatkan landasan kerangka berpikir yang berkaitan dengan penelitian, sehingga bisa didapatkan konsep dasar, dasar teori dan metode yang tepat dalam pelaksanaan penelitian. Sehingga penelitian memiliki dasar ilmiah dan bisa dipertanggungjawabkan. Pada tahap ini penulis melakukan pencarian mengenai dasar-dasar teori yang berguna untuk mendukung jalannya penelitian. Referensi yang digunakan diambil dari buku, jurnal, penelitian terdahulu, dan dokumentasi resmi. Dasar teori yang digunakan adalah teori yang memiliki kaitan dengan pengembangan sistem informasi yang menerapkan model *waterfall* dengan menggunakan UML sebagai acuan dalam perancangan sistem, Implementasi sistem menggunakan bahasa pemrograman PHP dengan arsitektur MVC, Berkaitan dengan kegiatan manajemen akademik persekolahan, dan menggunakan *framework* Codeigniter dalam implementasinya. Hasil dari kegiatan studi literatur diharapkan dapat membantu dalam proses pengembangan perangkat lunak (*SDLC*) yang meliputi analisis kebutuhan, perancangan, implementasi dan pengujian. Sehingga dapat dihasilkan sistem informasi yang memenuhi kebutuhan tanpa mengeluarkan biaya yang besar.

3.3 Analisis Kebutuhan

Tahap analisis kebutuhan merupakan tahap awal dan paling penting dalam pengembangan sistem informasi, tahap ini diperlukan untuk mendapatkan batasan sistem dan kebutuhan sistem dengan melakukan diskusi lebih lanjut dengan para pemangku kepentingan (*stakeholder*) dan calon pengguna sistem. Pada tahap ini dilakukan penggalian kebutuhan sistem yang akan dikembangkan, kegiatan pertama yang dilakukan adalah *elisitasi* kebutuhan dengan para *stakeholder* dalam tahap ini dilakukan negosiasi terhadap pengembangan sistem yang meliputi batasan fitur, teknologi yang digunakan dan proses bisnis sistem.

Setelah dilakukan elisitasi maka dapat dijabarkan mengenai kebutuhan fungsional dan kebutuhan non fungsional sistem, alat dan bahan yang diperlukan dalam proses pengembangan, dan aktor yang terlibat. Data yang diperoleh pada tahap ini selanjutnya akan digunakan sebagai acuan dalam tahap perancangan sistem. Data tersebut meliputi kebutuhan fungsional dan non fungsional sistem, yang akan dimodelkan dengan UML, yang meliputi *use case diagram, activity diagram, sequence diagram* dan *class diagram*.

3.4 Perancangan

Tahap perancangan befungsi untuk membuat rancangan sistem informasi yang meliputi *use case diagram, activity diagram, sequence diagram* dan *class diagram* dengan mengacu pada data yang telah diperoleh pada tahap analisis kebutuhan. Pada tahap ini dilakukan perancangan sistem informasi akademik sekolah yang meliputi alur sistem dan arsitektur sistem dengan menggunakan data yang diperoleh pada tahap analisis kebutuhan. Kebutuhan-kebutuhan yang telah didefinisikan dimodelkan menggunakan bahasa UML, yang meliputi *class diagram* untuk memetakan *class-class* dan fungsi-fungsi yang akan diterapkan pada sistem, *sequence diagram* untuk mengatur perilaku dari sebuah objek dalam waktu hidup tertentu, *use case diagram* untuk memetakan interaksi pengguna dengan sistem. Hasil dari proses perancangan akan sangat membantu dalam tahap implementasi sistem informasi.

3.5 Implementasi

Tahap implementasi dilakukan untuk membangun sistem informasi melalui pengkodean yang mengacu pada hasil perancangan sistem. Tahap pertama yang dilakukan adalah membuat tabel-tabel yang telah dinormalisasi dalam MySQL dengan *field* dan tipe data yang telah didefinisikan pada tahap perancangan. Implementasi menggunakan bahasa pemrograman PHP dengan *framework* Codeigniter yang menerapkan arsitektur MVC untuk membangun alur logika pemrosesan data antara pengguna dengan *web server* dan *database*. Pembangunan antarmuka menggunakan *framework* Bootstrap. Hasil dalam tahap implementasi adalah sebuah sistem manajemen sekolah yang siap untuk dilakukan pengujian *black-box* dan *user-acceptance test*.

3.6 Pengujian

Tahap pengujian dilakukan untuk menguji apakah sistem informasi yang telah dibuat pada tahap implementasi telah memenuhi kebutuhan yang telah didefinisikan. Pengujian yang dilakukan pada tahap ini meliputi pengujian *black-box* dan UAT. Pengujian *black-box* meliputi pengujian validitas, fungsionalitas, dan kompatibilitas dengan membandingkan hasil keluaran menggunakan parameter yang telah ditetapkan di tahap analisis kebutuhan dalam pengujian *black-box* perlu dipastikan bahwa sistem telah memenuhi kebutuhan fungsional dan nonfungsional tanpa mengalami *error*. Pengujian *black-box* bersifat *iterative* yang mana apabila sistem tidak lolos pengujian maka akan dilakukan perbaikan dan

pengujian ulang sampai sistem lolos tahap pengujian. Pengujian UAT untuk mendapatkan persentase penerimaan pengguna terhadap sistem informasi akademik sekolah. Hasil yang diperoleh dalam tahap pengujian berguna dalam kegiatan penarikan kesimpulan dan saran dalam pelaksanaan penelitian.

3.7 Kesimpulan dan Saran

Kesimpulan dan saran berguna untuk menjawab rumusan masalah penelitian. Penarikan kesimpulan dilakukan setelah semua tahapan penelitian yang meliputi studi literatur, analisis kebutuhan, perancangan, implementasi dan pengujian selesai dilakukan. Hasil kesimpulan diperoleh berdasarkan tahap-tahap tersebut, yang mana hasil dari kesimpulan berguna untuk menjawab rumusan masalah penelitian serta untuk memberikan saran terhadap penelitian yang dilakukan demi pengembangan sistem atau penelitian lebih lanjut.



4.1 Analisis Kebutuhan

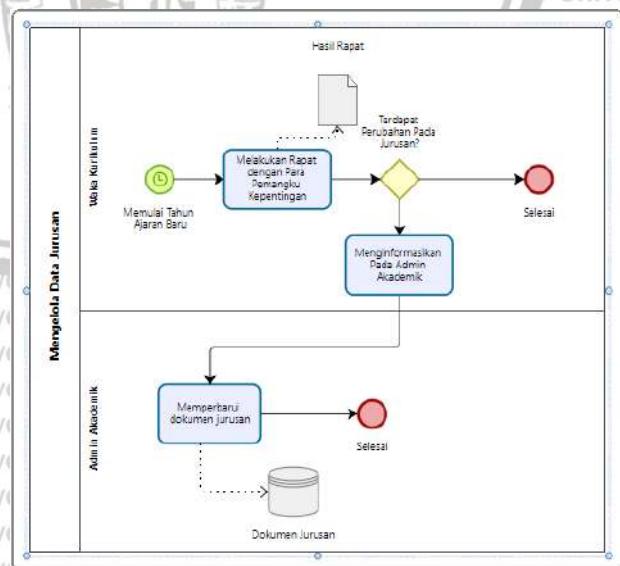
Analisis kebutuhan merupakan salah satu tahapan dalam pengembangan menggunakan model waterfall, dalam tahapan ini dilakukan penspesifikasi sifat kebutuhan fungsional dan nonfungsional perangkat lunak, deskripsi umum sistem, elisitasi kebutuhan, identifikasi aktor, dan pemberian kode fungsi sistem.

4.1.1 Identifikasi Proses Bisnis Saat Ini

Kegiatan akademik di SMK Muhammadiyah 1 Kemlagi merupakan suatu sistem yang terdiri dari berbagai proses bisnis yang menyertainya. Kegiatan akademik melibatkan berbagai *stakeholder* sehingga menjadi suatu proses yang kompleks. Untuk memudahkan penjabaran proses bisnis saat ini maka proses bisnis akademik yang ada dimodelkan dalam notasi *BPMN*.

1. Proses Bisnis Pengelolaan Data Jurusan

Proses bisnis pengelolaan data jurusan merupakan kegiatan dalam pencatatan informasi-informasi mengenai jurusan yang ada pada SMK Muhammadiyah 1 Kemlagi. Informasi-informasi tersebut meliputi kepala jurusan, jumlah kelas dan jumlah siswa. Pengelolaan meliputi penambahan apabila sekolah membuka jurusan baru, pengubahan pada data jurusan saat ini, dan penghapusan apabila sekolah memutuskan untuk menutup sebuah jurusan. Pendaftaran dilakukan oleh admin akademik dan data jurusan tersebut diperbarui secara berkala dalam sebuah dokumen berformat *Excel*. Pemodelan proses bisnis pengelolaan data jurusan dapat dilihat pada Gambar 4. 1.

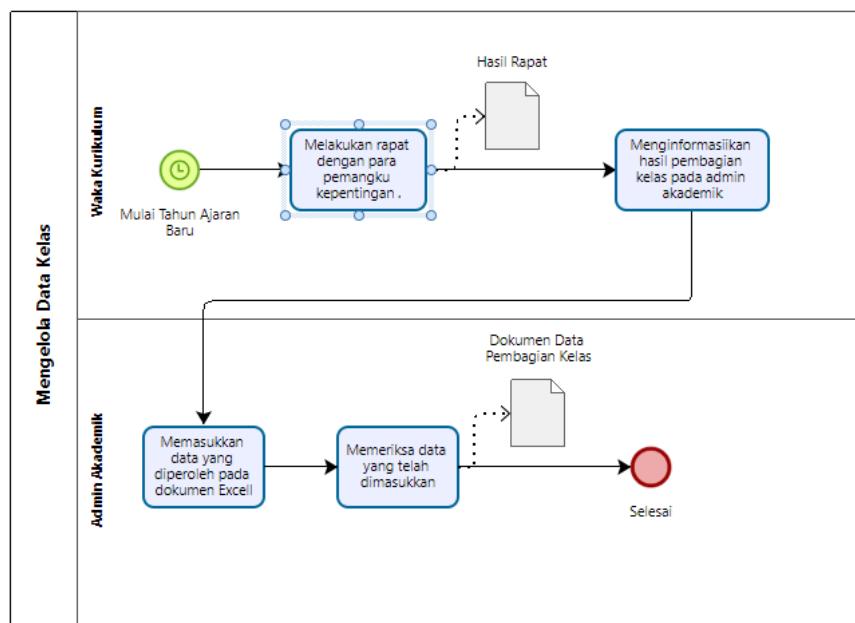


Gambar 4. 1 Pemodelan Proses Bisnis Pengelolaan Data Jurusan

Kegiatan pengelolaan data jurusan dilakukan setahun sekali dan bergantung pada hasil rapat tahunan antara waka kurikulum dengan para pemangku kepentingan. Apabila diputuskan perlu ada perubahan dalam jurusan entah itu kepala jurusan baru, jumlah kelas yang dibuka dalam setiap jurusan, atau kuota siswa dalam setiap jurusan, maka waka kurikulum selanjutnya menginformasikan hal tersebut pada admin akademik yang selanjutnya admin akademik akan memperbarui dokumen informasi jurusan yang saat ini dipakai.

2. Proses Bisnis Pengelolaan Data Kelas

Proses bisnis pengelolaan data kelas merupakan kegiatan dalam pencatatan informasi-informasi mengenai pembagian kelas pada tahun ajaran baru. Informasi-informasi tersebut meliputi penentuan wali kelas dan penempatan siswa dalam kelas tersebut. Pengelolaan meliputi pembuatan record kelas baru sesuai dengan tahun ajaran saat ini, pengubahan apabila terjadi perubahan dalam kelas seperti terdapat murid pindahan atau apabila ada siswa yang dikeluarkan, dan penghapusan apabila terjadi kekeliruan data. Pendataan dilakukan oleh admin akademik dengan membuat dokumen baru berformat Excel. Pemodelan proses bisnis pengelolaan data kelas dapat dilihat pada Gambar 4. 2.



Gambar 4. 2 Pemodelan Proses Bisnis Pengelolaan Data Kelas

Kegiatan pengelolaan data kelas merupakan kegiatan rutin setiap tahun ajaran baru. Berikut merupakan penjelasan dari proses bisnis tersebut:

- A. Waka Kurikulum melakukan koordinasi untuk penetapan pembagian kelas dengan para pemangku kepentingan (Kepala Sekolah, Waka, Kajur).

B. Hasil dari rapat akan diinformasikan pada admin akademik untuk selanjutnya diproses dalam dokumen Excel.

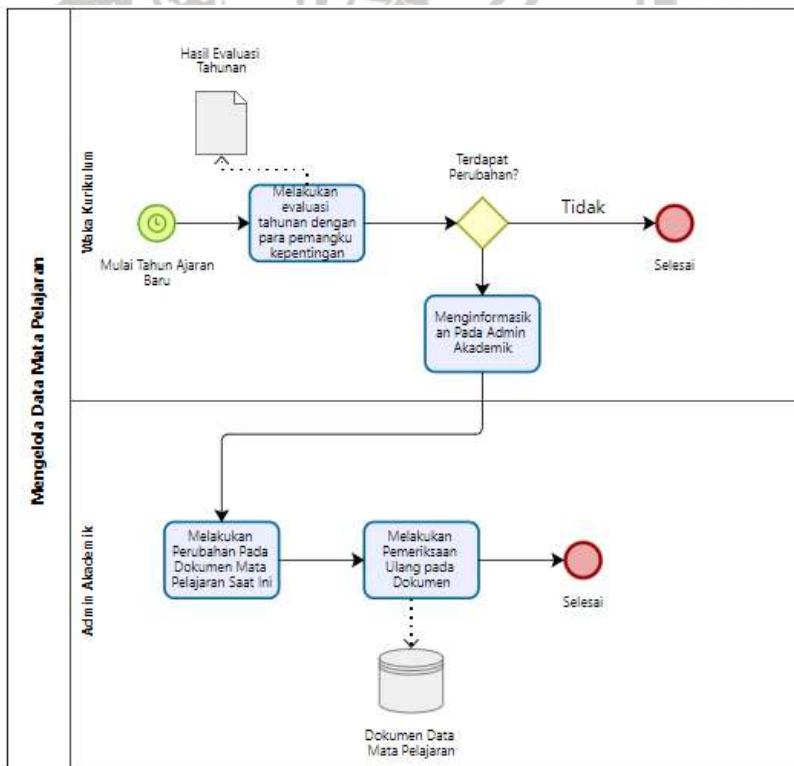
C. Karena merupakan data yang krusial maka admin akademik perlu memastikan bahwa data yang dimasukkan benar-benar valid.

D. Hasil akhir dari proses ini adalah dokumen pembagian kelas yang selanjutnya akan diberikan pada para guru.

3. Proses Bisnis Pengelolaan Data Mata Pelajaran

Proses bisnis pengelolaan data mata pelajaran merupakan kegiatan dalam pencatatan informasi-informasi mengenai informasi mata pelajaran yang akan diajarkan di SMK Muhammadiyah 1 Kemlagi sesuai tahun ajaran saat ini. Informasi-informasi tersebut meliputi guru pengajar, lama jam pelajaran dan jurusan yang memuat mata pelajaran tersebut. Pengelolaan meliputi penambahan mata pelajaran apabila sekolah menambahkan mata pelajaran baru, pengubahan informasi mata pelajaran seperti guru pengajar atau lama jam pelajaran dan penghapusan apabila sekolah memutuskan untuk mengganti/meniadakan suatu mata pelajaran. Pengelolaan dilakukan oleh admin akademik dengan memperbarui secara berkala pada dokumen data mata pelajaran. Pemodelan proses bisnis pengelolaan data kelas dapat dilihat pada

Gambar 4. 3



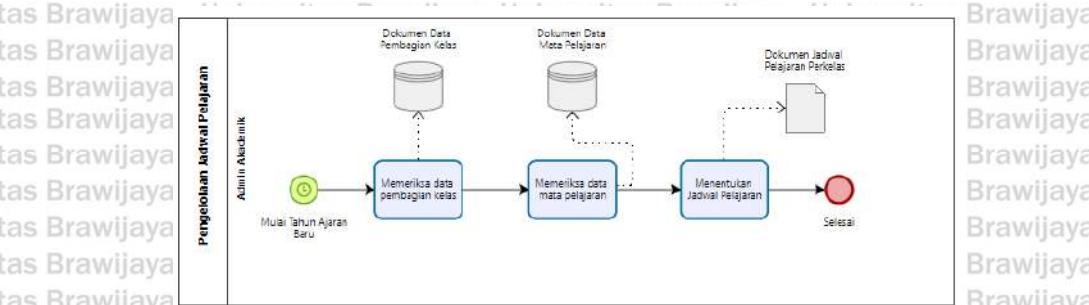
Gambar 4. 3 Pemodelan Proses Bisnis Mengelola Data Mata Pelajaran

Kegiatan pengelolaan data mata pelajaran dilakukan setahun sekali dan bergantung pada hasil evaluasi tahunan antara waka kurikulum dengan para pemangku kepentingan. Penjelasan dari model proses bisnis yang telah dijabarkan sebelumnya adalah sebagai berikut:

- A. Waka Kurikulum melakukan rapat tahunan dengan para pemangku kepentingan untuk evaluasi mata pelajaran. Evaluasi mempertimbangkan apakah suatu mata pelajaran pada suatu jurusan perlu dilakukan perubahan guru pengajarnya, atau sekolah perlu menambah atau menghilangkan suatu mata pelajaran.
- B. Apabila hasil evaluasi menyatakan bahwa tidak diperlukan perubahan pada mata pelajaran yang ada saat ini, maka tidak akan dilakukan perubahan pada informasi mata pelajaran.
- C. Apabila hasil evaluasi menyatakan perlu melakukan perubahan maka waka kurikulum akan menginformasikan hasil rapat pada admin akademik.
- D. Admin akademik melakukan pembaruan pada dokumen data mata pelajaran.
- E. Admin akademik melakukan pengecekan ulang untuk memastikan bahwa tidak ada kekeliruan dalam perubahan data.
- F. Hasil dari proses bisnis ini adalah dokumen data mata pelajaran terbaru.

4. Pengelolaan Data Jadwal Pelajaran

Proses bisnis pengelolaan data jadwal pelajaran merupakan kegiatan rutin tahunan yang dilakukan oleh admin akademik setelah mendapatkan data mata pelajaran terbaru dari proses pengelolaan mata pelajaran. Kegiatan pengelolaan meliputi pembuatan dokumen jadwal pelajaran baru untuk tiap kelas, pengubahan jadwal pelajaran apabila ada perubahan pada jadwal pelajaran dan penghapusan jadwal pelajaran apabila terdapat kesalahan pada jadwal pelajaran. Pemodelan proses bisnis pengelolaan data jadwal pelajaran terdapat pada Gambar 4. 4.



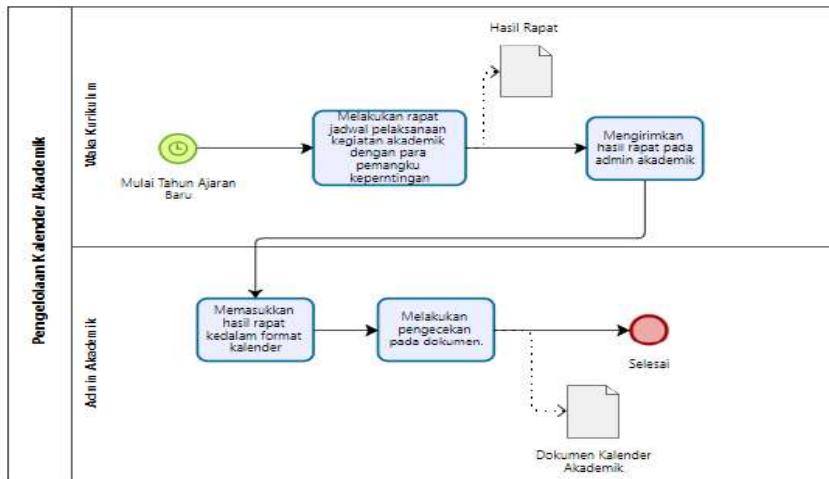
Gambar 4.4 Pemodelan Proses Bisnis Pengelolaan Jadwal Pelajaran

Berikut merupakan penjelasan dari model proses bisnis pada Gambar 4.4 :

- Admin akademik melakukan pemeriksaan pada dokumen pembagian data pembagian kelas yang telah dibuat sebelumnya.
- Admin akademik melakukan pemeriksaan pada dokumen data mata pelajaran yang telah dibuat sebelumnya.
- Admin akademik selanjutnya menentukan jadwal pelajaran menyesuaikan dengan data mata pelajaran yang ada pada tahun ajaran saat ini.
- Akan dihasilkan dokumen jadwal pelajaran seluruh kelas yang dikategorikan berdasarkan kelas-kelas yang ada.

5. Pengelolaan Kalender Akademik

Pengelolaan akademik merupakan kegiatan rutin tahunan yang dilakukan oleh admin akademik. Kegiatan pengelolaan meliputi pembuatan dokumen kalender akademik baru, pengubahan kalender akademik lama, dan penghapusan kalender akademik yang tidak terpakai lagi. Pemodelan proses bisnis pengelolaan kalender akademik dapat dilihat pada **Error! Reference source not found..**



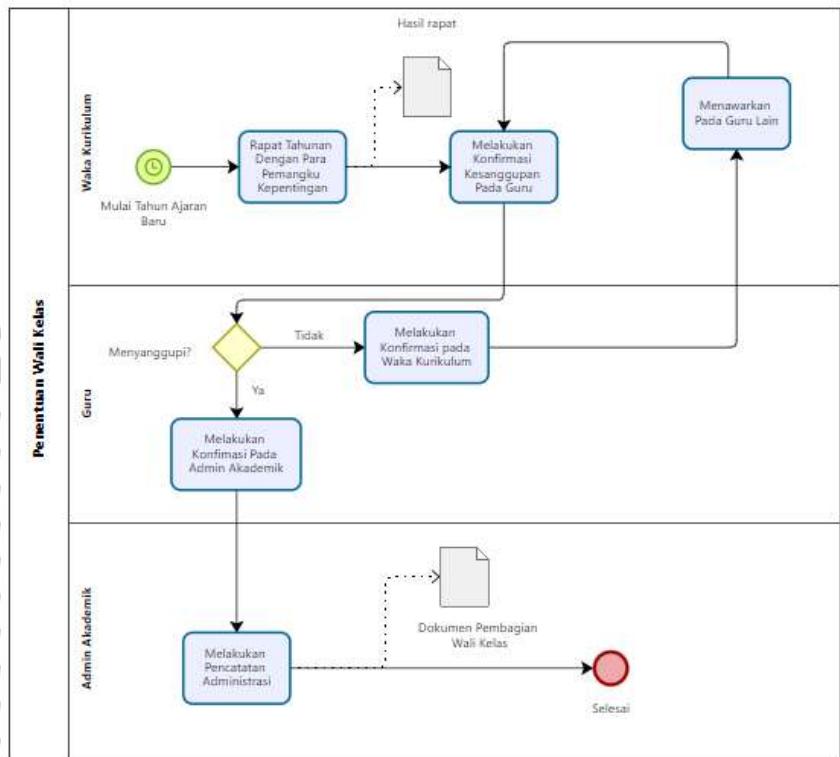
Gambar 4. 5 Pemodelan Proses Bisnis Pengelolaan Kalender Akademik

Berikut penjelasan pemodelan proses bisnis pengelolaan kalender akademik pada [Error! Reference source not found.](#):

- A. Waka kurikulum melakukan rapat dengan para pemangku kepentingan mengenai *timeline* pelaksanaan kegiatan akademik pada tahun ajaran saat ini. Rapat akan menghasilkan hasil mengenai jadwal pelaksanaan kegiatan akademik selama satu tahun.
- B. Waka kurikulum mengirimkan hasil rapat pada admin akademik agar diproses lebih lanjut.
- C. Admin akademik mengkonversi *timeline* jadwal pelaksanaan akademik ke dalam format kalender.
- D. Admin melakukan pengecekan apakah data yang dimasukkan telah sesuai.
- E. Hasil dari proses ini adalah dokumen kalender akademik yang siap untuk diterbitkan.

6. Proses Bisnis Penentuan Wali Kelas

Penentuan wali kelas merupakan kegiatan rutin tahunan yang melibatkan beberapa pihak. Pemodelan proses bisnis penentuan wali kelas terdapat pada [Error! Reference source not found.](#).



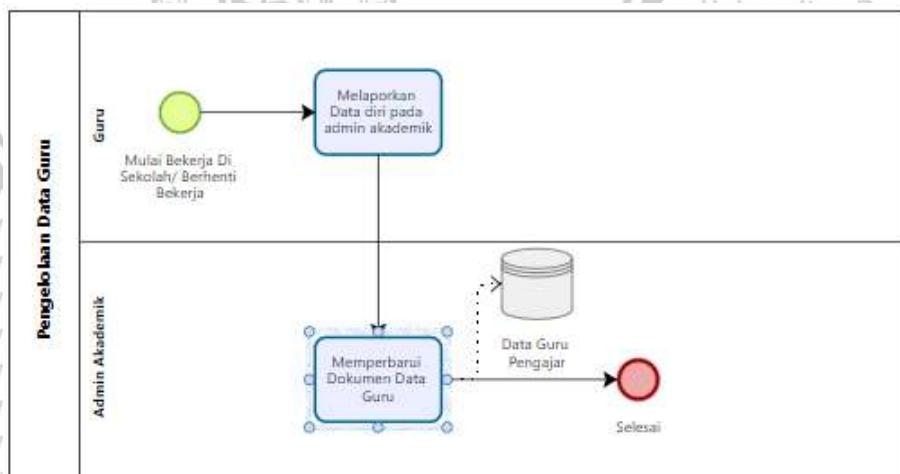
Gambar 4. 6 Pemodelan Proses Bisnis Penentuan Wali Kelas

Berikut penjelasan dari proses bisnis penentuan wali kelas pada **Error! Reference source not found.**:

- A. Waka kurikulum melakukan rapat rutin tahunan dengan para pemangku kepentingan, hasil dari rapat adalah hasil rapat penentuan calon wali kelas.
 - B. Selanjutnya waka kurikulum menanyakan kesanggupan guru yang ditunjuk sebagai calon wali kelas.
 - C. Apabila guru yang ditunjuk menyatakan kesanggupan, maka selanjutnya guru calon wali kelas melakukan konfirmasi pada admin akademik untuk selanjutnya dilakukan pencatatan administrasi.
 - D. Apabila guru yang ditunjuk menyatakan tidak sanggup, maka selanjutnya waka kurikulum akan menawarkan pada guru yang lain, dan melakukan konfirmasi kesanggupan. Proses ini dapat terjadi secara berulang sampai didapati guru yang sanggup.
 - E. Hasil dari proses ini adalah dokumen pembagian peran wali kelas.

7. Pengelolaan Data Guru

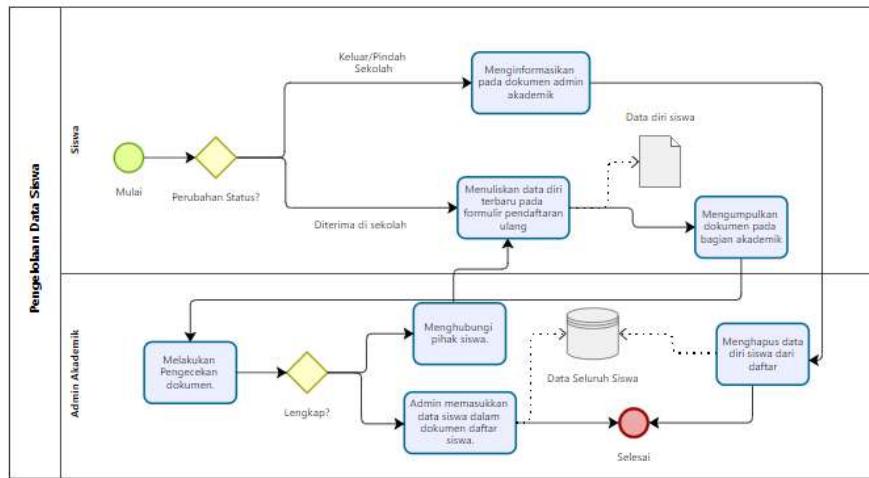
Pengelolaan data guru merupakan kegiatan pencatatan perubahan pada data guru pengajar yang ada di SMK Muhammadiyah 1 Kemlagi. Kegiatan pengelolaan meliputi penambahan data guru apabila terdapat guru baru yang bekerja di sekolah, dan penghapusan apabila ada guru yang sudah tidak aktif lagi mengajar di sekolah. Pemodelan proses bisnis pengelolaan data guru pada Gambar 4. 7.



Gambar 4.7 Pemodelan Proses Bisnis Pengelolaan Data Guru

Berikut penjelasan dari pemodelan proses bisnis pengelolaan data guru pada Gambar 4-7:

- A. Guru yang mulai bekerja atau berhenti bekerja di SMK Muhammadiyah 1 Kemlagi diwajibkan melakukan pelaporan mengenai data diri pada admin akademik.
- B. Admin akademik melakukan pembaruan data guru pada dokumen data guru pengajar.
- 8. Pengelolaan Data Siswa**
- Pengelolaan data siswa merupakan kegiatan yang dilakukan apabila terdapat penambahan siswa baru atau siswa yang keluar/pindah di SMK Muhammadiyah 1 Kemlagi. Kegiatan pengelolaan meliputi penambahan data apabila terdapat murid baru, penghapusan data apabila terdapat murid yang keluar, dikeluarkan, atau pindah sekolah. Pemodelan proses bisnis pengelolaan data siswa terdapat pada Gambar 4. 8.



Gambar 4. 8 Pemodelan Proses Bisnis Pengelolaan Data Siswa

Berikut penjelasan pemodelan proses bisnis pengelolaan data siswa pada Gambar 4. 8:

- Apabila siswa menyatakan ingin pindah sekolah/dikeluarkan, setelah mengurus administrasi yang diperlukan maka siswa perlu melakukan konfirmasi akhir pada admin akademik.
- Admin akademik menghapus data siswa dari dokumen daftar siswa aktif sekolah.
- Apabila siswa dinyatakan diterima sebagai peserta didik di SMK Muhammadiyah 1 Kemlagi, maka siswa perlu menuliskan data diri pada formulir daftar ulang yang selanjutnya diberikan ke admin akademik.
- Admin akademik melakukan pengecekan data diri siswa pada formulir daftar ulang.

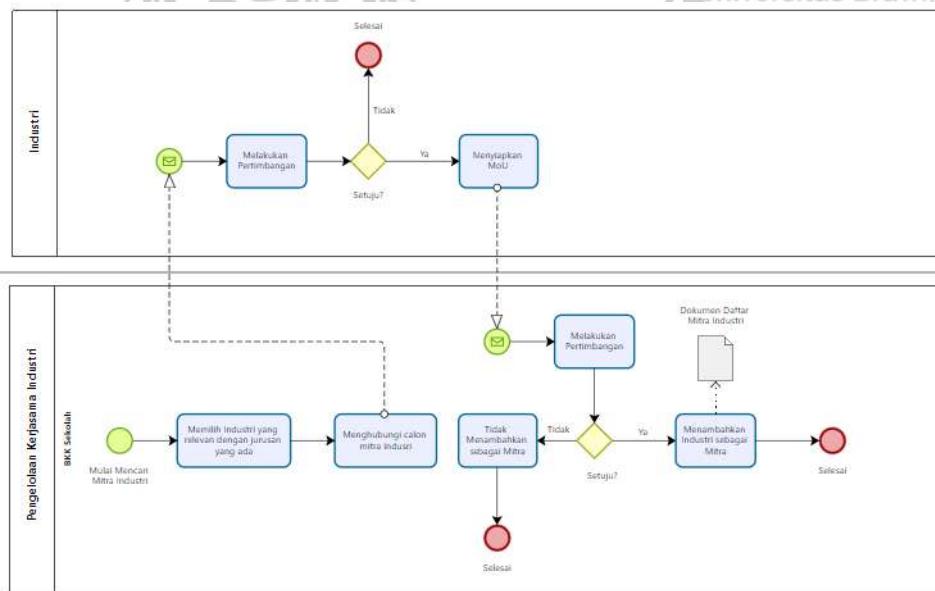


- E. Apabila data siswa dinyatakan valid maka admin akademik memasukkan data diri siswa pada dokumen daftar siswa aktif sekolah.
- F. Apabila data siswa pada formulir dinyatakan tidak valid (ada kesalahan, ada bagian yang belum diisi), maka admin akademik akan menghubungi pihak siswa yang bersangkutan agar segera melakukan perbaikan.
- G. Pihak siswa menuliskan ulang data diri pada formulir pendaftaran dan mengumpulkan kembali pada admin akademik.
- H. Hasil akhir pada proses bisnis ini adalah perubahan pada data daftar siswa aktif sekolah.

9. Pengelolaan Mitra Industri

Pengelolaan Mitra Industri merupakan kegiatan yang dilakukan SMK Muhammadiyah 1 Kemlagi untuk meningkatkan mutu dan pelayanan sekolah. Mitra industri bagi sekolah menengah kejuruan adalah salah satu komponen penting dalam pelaksanaan kegiatan akademik. Dengan bermitra SMK Muhammadiyah 1 Kemlagi dapat menghubungkan lulusan untuk bekerja di industri atau mengarahkan siswa untuk melaksanakan kegiatan prakerin di mitra industri. Pengelolaan mitra industri dilakukan oleh Bursa Kerja Khusus (BKK) SMK Muhammadiyah 1 Kemlagi. Pemodelan proses bisnis pengelolaan mitra industry terdapat pada 4. 9.

Gambar



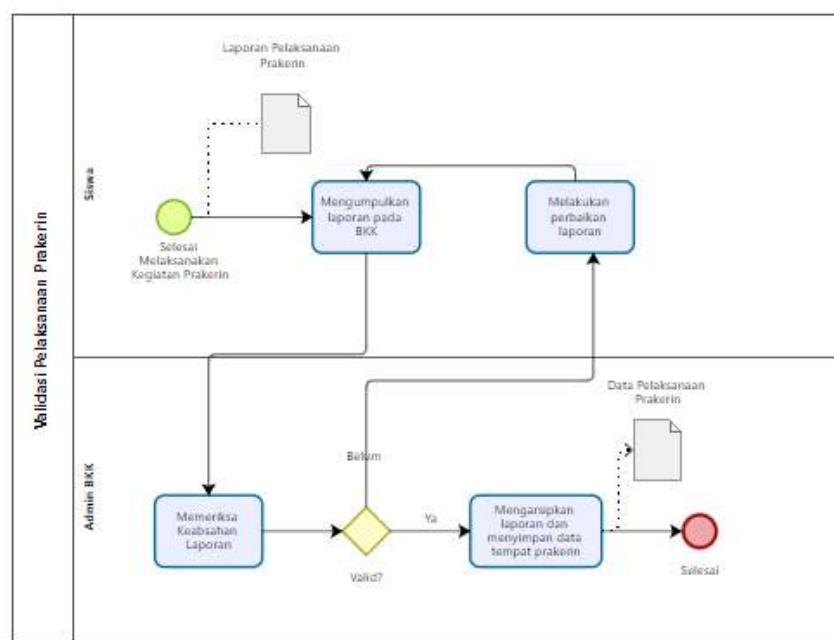
Gambar 4. 9 Pemodelan Proses Bisnis Pengelolaan Mitra Industri

Berikut penjelasan mengenai proses bisnis Pengelolaan mitra industri pada Gambar 4. 9:

- A. Admin BKK memilih industri yang relevan yaitu industri yang memiliki kesesuaian bidang dengan profil lulusan atau jurusan yang ada di SMK Muhammadiyah 1 Kemlagi.
- B. Admin BKK menghubungi pihak industri melalui surat atau media elektronik seperti email atau telpon.
- C. Setelah permohonan terkirim ke pihak industry, pihak industri melakukan pertimbangan terhadap permohonan SMK.
- D. Apabila pihak industri menolak karena suatu hal, maka proses dianggap selesai.
- E. Apabila pihak indutri setuju maka pihak industri akan menyiapkan MoU untuk dibahas dengan pihak SMK Muhammadiyah 1 Kemlagi.
- F. Admin BKK memeriksa dan melakukan pertimbangan terhadap MoU dari pihak industri calon mitra.
- G. Apabila pihak BKK tidak menyetujui maka tidak menambahkan pihak industri sebagai mitra.
- H. Apabila pihak BKK menyetujui maka admin BKK menambahkan pihak industri sebagai mitra industri dan mencatat pada dokumen mitra industri.

10. Validasi Pelaksanaan Prakerin Siswa

Validasi pelaksanaan prakerin siswa merupakan kegiatan yang dilakukan oleh admin BKK untuk melakukan pendataan mengenai tempat prakerin siswa. Siswa yang telah melaksanakan kegiatan prakerin wajib untuk melakukan pelaporan akhir pada BKK untuk pencatatan data industri tempat prakerin siswa. Setelah laporan siswa dinyatakan valid maka siswa dianggap sudah melaksanakan prakerin. Pemodelan proses bisnis validasi pelaksanaan prakerin pada Gambar 4. 10.



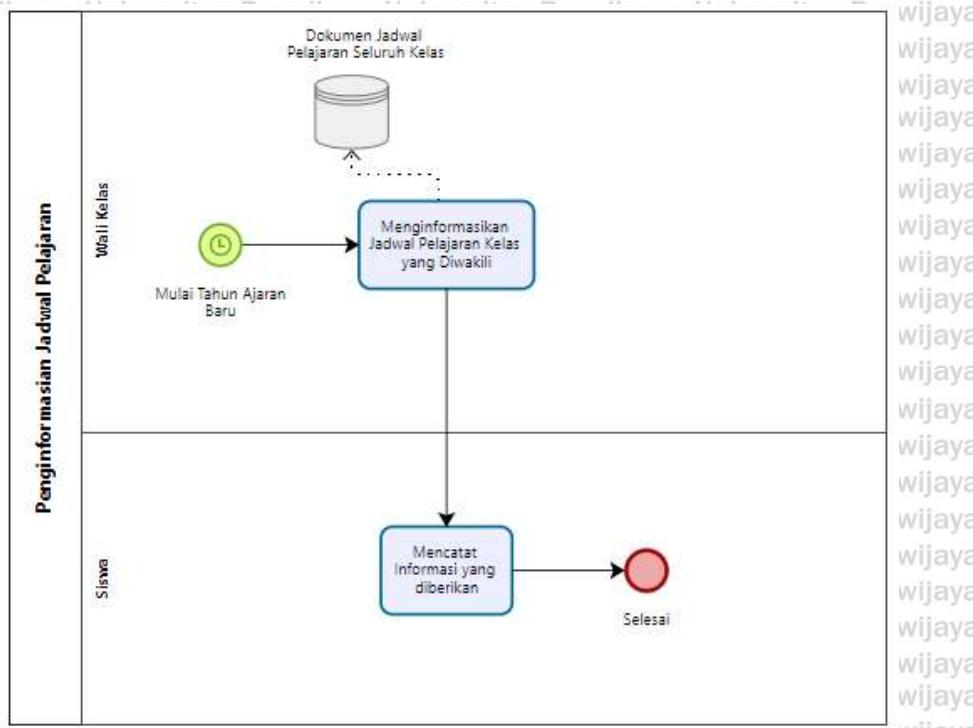
Gambar 4. 10 Pemodelan Proses Bisnis Validasi Pelaksanaan Prakerin

Berikut penjelasan dari pemodelan proses bisnis validasi pelaksanaan prakerin pada Gambar 4. 10:

1. Setelah melaksanakan kegiatan prakerin dan menyelesaikan laporan prakerin, siswa diwajibkan untuk mengumpulkan laporan pada bagian Bursa Kerja Khusus (BKK).
2. Admin BKK melakukan pemeriksaan pada laporan kegiatan prakerin siswa.
3. Apabila laporan prakerin siswa tidak valid maka siswa perlu melakukan perbaikan pada laporan dan mengumpulkan ulang pada BKK.
4. Apabila laporan prakerin siswa telah valid maka selanjutnya laporan siswa akan diarsipkan untuk keperluan lebih lanjut.
5. Hasil dari proses ini adalah dokumen data industri yang dapat diajak bermitra dengan sekolah.

11. Penginformasian Jadwal Pelajaran

Penginformasian jadwal pelajaran merupakan kegiatan yang dilakukan oleh wali kelas pada siswa di kelas yang diwakilinya. Pemodelan proses bisnis penginformasian jadwal pelajaran pada Gambar 4. 11.



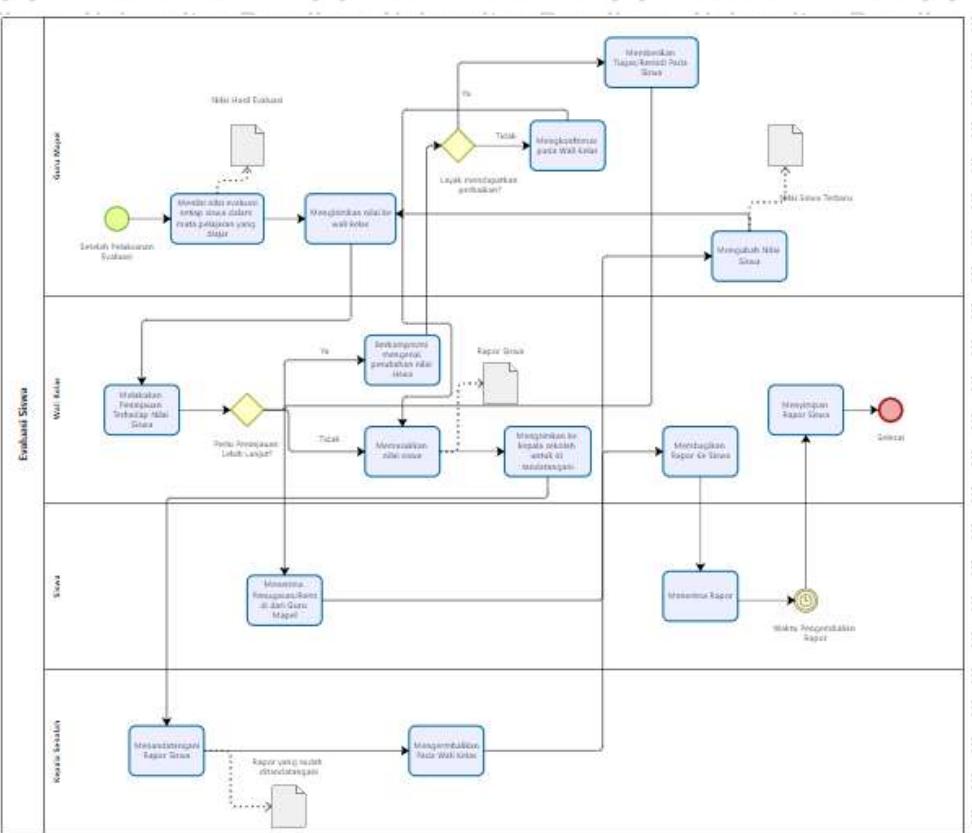
Gambar 4. 11 Pemodelan Proses Bisnis Penginformasian Jadwal Pelajaran

Berikut penjelasan pemodelan proses bisnis penginformasian jadwal pelajaran pada Gambar 4. 11:

1. Wali Kelas mengumpulkan para siswa kelas yang diwakilinya dan membacakan jadwal pelajaran kelas sesuai dengan yang tertulis pada dokumen jadwal pelajaran.
2. Siswa mendengarkan dan mencatat jadwal pelajaran yang dibacakan oleh guru.

12. Evaluasi Siswa

Evaluasi siswa merupakan kegiatan yang dilakukan pada saat tengah semester dan akhir semester. Proses bisnis ini merupakan proses bisnis yang paling kompleks diantara proses bisnis akademik yang lain. Pemodelan proses bisnis evaluasi siswa terdapat pada Gambar 4. 12.



Gambar 4. 12 Pemodelan Proses Bisnis Evaluasi Siswa

Berikut penjelasan pemodelan proses bisnis evaluasi siswa pada Gambar 4. 12:

- A. Guru mata pelajaran melakukan rekapitulasi terhadap nilai hasil evaluasi siswa sesuai dengan mata pelajaran yang diajarnya, dan menyatukan dalam dokumen hasil nilai evaluasi.
 - B. Guru mata pelajaran selanjutnya mengirimkan dokumen nilai hasil evaluasi pada wali kelas.
 - C. Wali kelas selanjutnya melakukan peninjauan nilai yang diperoleh oleh siswa.
 - D. Wali kelas memasukkan nilai pada dokumen rapor siswa.
 - E. Apabila didapati satu atau lebih siswa dalam kelas yang mengalami masalah dalam nilai, maka wali kelas melakukan kompromi dengan guru mata pelajaran mengenai perubahan nilai siswa yang bermasalah.
 - F. Guru mata pelajaran menindaklanjuti permintaan dari wali kelas dan menentukan apakah siswa yang bersangkutan layak mendapatkan perbaikan.
 - G. Apabila tidak layak maka guru mata pelajaran melakukan konfirmasi dan memberi penjelasan pada wali kelas.

H. Apabila layak maka guru mata pelajaran mempersiapkan pelaksanaan remidi/perbaikan nilai siswa.

I. Siswa melaksanakan perbaikan nilai/remidi yang diberikan oleh guru mata pelajaran.

J. Guru mata pelajaran selanjutnya memperbarui nilai siswa sesuai dengan hasil perbaikan/remidi. Lalu mengirimkan nilai terbaru pada wali kelas.

K. Wali kelas mengirimkan rapor nilai siswa ke kepala sekolah untuk ditandatangani.

L. Kepala sekolah menandatangani rapor siswa dan mengembalikannya kepada wali kelas.

M. Wali kelas membagikan rapor pada pihak siswa (wali murid).

N. Setelah menerima rapor siswa diwajibkan mengembalikannya sesuai dengan waktu yang ditentukan.

O. Wali kelas menyimpan rapor siswa.

4.1.2 Analisis Proses Bisnis Saat Ini

Berdasarkan pada model-model proses bisnis pada bagian identifikasi proses bisnis. Tahap analisis proses bisnis mengidentifikasi masalah yang ada dalam proses bisnis saat ini, serta dampak dari masalah tersebut. Sehingga dapat diberikan solusi atas permasalahan yang ada. Hasil pemetaan masalah didapatkan dari analisis dokumen kurikulum dan wawancara dengan pihak sekolah sehingga dapat diperoleh hasil semaksimal mungkin. Hasil analisis proses bisnis saat ini dapat dilihat pada Tabel 4. 1.

Tabel 4. 1 Tabel Analisis Proses Bisnis Saat ini

| No | Masalah | Dampak | Solusi |
|----|--|---|---|
| 1 | Pendataan data-data akademik seperti data siswa, data guru, data jurusan, data mata pelajaran masih dalam dokumen-dokumen yang terpisah. | <ul style="list-style-type: none"> Pencarian data menjadi sulit. Rawan terjadi kesalahan data. Rawan terjadi kehilangan dokumen. | Menerapkan sistem informasi yang dapat membantu pengelolaan data secara terpusat (<i>database</i>). |

Tabel 4. 1 Tabel Analisis Proses Bisnis Saat ini (Lanjutan)

| No | Masalah | Dampak | Solusi |
|----|---|---|--|
| 2 | Penyebaran informasi masih kurang <i>reliable</i> . | <ul style="list-style-type: none"> Rawan terjadi miskomunikasi antar <i>stakeholder</i>. Kurang mendukung peningkatan pelayanan sekolah. | Menerapkan sistem informasi yang memiliki fitur penerbitan informasi secara terpusat untuk seluruh <i>stakeholder</i> . |
| 3 | Masih terdapat proses yang berulang seperti validasi data siswa. | <ul style="list-style-type: none"> Menambah beban kerja aktor. Memperlambat kegiatan akademik. | Menerapkan sistem informasi yang memungkinkan pendeklegasian tugas sesuai dengan tupoksi masing-masing <i>stakeholder</i> dan dapat melakukan pengecekan otomatis terhadap kelengkapan data yang dimasukkan. |
| 4 | Pengolahan dokumen masih terbatas pada dua (admin akademik dan admin BKK) perangkat komputer. | <ul style="list-style-type: none"> Ancaman kehilangan seluruh data akibat virus atau kesalahan sangat besar. Menghambat proses akademik apabila ditemukan masalah pada perangkat. | Menerapkan sistem informasi yang memungkinkan pengaksesan pada banyak perangkat. |

Tabel 4. 1 Tabel Analisis Proses Bisnis Saat ini (Lanjutan)

| No | Masalah | Dampak | Solusi |
|----|--|--|---|
| 5 | Masih belum ada proses pengecekan akhir dari Waka Kurikulum. | <ul style="list-style-type: none"> • Memungkinkan terjadinya kesalahan yang dilakukan oleh admin akademik. • Memungkinkan evaluasi tahunan menjadi kurang efektif karena kesalahan data. | Menerapkan sistem informasi yang memungkinkan Waka Kurikulum dapat melakukan validasi data. |

4.1.3 Deskripsi Sistem Informasi

Sistem Informasi Akademik (SIAKAD) SMK Muhammadiyah 1 Kemlagi merupakan sistem informasi berbasis web yang dikembangkan untuk mengatasi permasalahan kegiatan akademik meliputi pendataan, penyebaran dan efisiensi pekerjaan. Nantinya SIAKAD diharapkan dapat mengatasi permasalahan yang telah dijabarkan sebelumnya. Lebih lanjut harapan fitur-fitur yang ada pada SIAKAD adalah sebagai berikut:

1. Pengolahan data akademik yang meliputi data siswa, data guru, prakerin, bursa kerja, mata pelajaran, jurusan, kelas, dan hasil evaluasi siswa (rapor) menjadi lebih mudah dan cepat. Menggunakan web yang dapat diakses secara *online*.
2. Menerapkan basis data dalam penyimpanan datanya sehingga dapat mengurangi pekerjaan yang bersifat repetitif dan rawan terjadi kesalahan seperti pemindahan file dari komputer satu ke komputer lain.
3. Menerapkan fitur pencarian sehingga data dapat ditemukan dengan mudah dan cepat.
4. Sistem informasi menerapkan tampilan antarmuka yang familiar dengan penggunaan sehari-hari sehingga memudahkan penggunaan sistem informasi, dalam hal ini antarmuka sistem diharapkan bersih dan modern, mengikuti gaya antarmuka yang banyak digunakan pada Android/Windows.
5. Memiliki tingkatan level akses penggunaan, sehingga pengguna hanya dapat melihat atau mengubah hal-hal yang relevan dengan statusnya di sekolah.

4.1.4 Identifikasi Pengguna

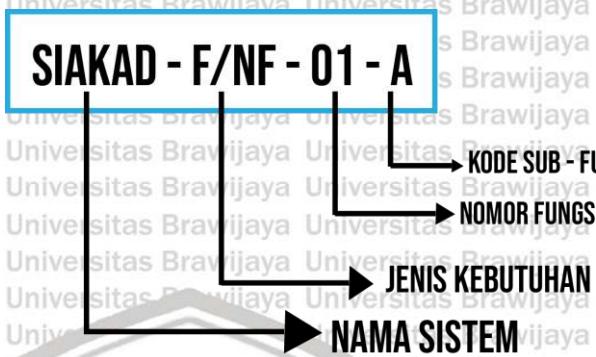
Siklus hidup sistem terdiri dari berbagai kejadian (*event*), salah satunya adalah interaksi antara sistem dengan aktor/pengguna, merujuk pada harapan fitur yang didefinisikan sebelumnya, maka sistem informasi akademik (SIAKAD) SMK Muhammadiyah 1 Kemlagi, menerapkan multi pengguna (*multi user*) yang memiliki tingkatan akses yang berbeda-beda sehingga interaksi dengan sistem juga berbeda. Aktor atau pengguna yang ada pada sistem informasi diidentifikasi pada Tabel 4. 2.

Tabel 4. 2 Identifikasi Pengguna Sistem

| No | Pengguna | Deskripsi |
|----|----------------|--|
| 1 | Admin Sistem | Admin sistem merupakan pengguna yang memiliki hak akses tertinggi terhadap sistem, hak akses tersebut meliputi pengubahan data siswa, pengubahan data guru, pengubahan data jadwal pelajaran, pengubahan data kelas, pengubahan data jurusan, pengubahan data siswa dan lain-lain. |
| 2 | Admin BKK | Admin prakerin merupakan pengguna sistem yang memiliki akses pengelolaan terhadap data-data yang berhubungan dengan industri, yang meliputi mitra industri dan prakerin siswa, |
| 3 | Waka Kurikulum | Waka kurikulum merupakan pengguna sistem yang memiliki akses terhadap beberapa fungsi sistem seperti melihat data mengenai prakerin dan melakukan validasi data kelas dan data kegiatan akademik. |
| 4 | Guru | Guru merupakan pengguna sistem yang memiliki akses terhadap beberapa fungsi seperti melihat jadwal pelajaran, memasukan nilai siswa, melihat laporan mengenai siswa perkelas, melihat detail kelas. |
| 5 | Siswa | Siswa merupakan yang memiliki akses terbatas terhadap sistem informasi, diantaranya memperbarui data diri secara berkala, mengajukan permohonan pelaksanaan prakerin, melihat data rapor, melihat detail kelas, dan memperbarui data diri. |

4.1.5 Identifikasi Kebutuhan

Pada bagian deskripsi sistem telah dijabarkan mengenai harapan fitur yang ada pada aplikasi secara general, pada bagian ini akan dilakukan penspesifikasi mengenai fitur-fitur yang ada pada sistem secara lebih detail dan terperinci meliputi kode fitur, deskripsi fitu, serta pengguna/aktor dari fitur tersebut. Untuk



Gambar 4. 13 Format Penomoran Kebutuhan

Penomoran kebutuhan pada penelitian ini menggunakan beberapa bagian seperti pada Gambar 4. 13. Bagian nama sistem dalam format merupakan singkatan dari Sistem Informasi Akademik yang merupakan sistem yang dikembangkan dalam penelitian ini. Pada bagian jenis kebutuhan terdiri dari 2 (dua) jenis, F untuk menyatakan bahwa kebutuhan yang dimaksud adalah kebutuhan fungsional, sedangkan NF menyatakan bahwa kebutuhan yang dimaksud adalah kebutuhan non fungsional. Bagian nomor fungsional digunakan untuk melakukan penomoran pada kebutuhan, kebutuhan diawali dengan nomor 1 (satu), sedangkan untuk bagian kode sub-fungsional diawali dengan huruf A.

Daftar kebutuhan pengembangan yang telah teridentifikasi dikelompokan menurut interaksi antara sistem dengan aktor/pengguna, pada penelitian ini, fungsionalitas/kebutuhan didefinisikan dengan kode fungsionalitas/fitur, nama fitur serta deskripsi dari fungsi tersebut. Berikut merupakan daftar kebutuhan yang ada dalam penelitian ini.

1. Semua Pengguna

Sistem informasi memiliki fitur yang bersifat *general* atau fitur yang dapat digunakan oleh seluruh pengguna tanpa pandang level akses pengguna tersebut. Fitur tersebut adalah fitur untuk melihat kalender akademik. Selain itu juga terdapat fitur wajib yang perlu dijalankan oleh sistem yaitu fitur *login* dan *logout*. Penjabaran lebih lengkap dapat dilihat pada Tabel 4. 3.

Tabel 4. 3 Identifikasi Fitur Semua Pengguna

| No | Kode Fitur | Nama Fitur | Deskripsi |
|----|-----------------|----------------------------------|---|
| 1 | SIAKAD – F – 01 | Melakukan <i>login</i> ke sistem | Untuk mengakses fungsi-fungsi yang ada pada sistem pengguna perlu melakukan <i>login</i> menggunakan <i>username</i> dan <i>password</i> , proses <i>login</i> mengidentifikasi level akses user. |
| 2 | SIAKAD – F – 02 | Melakukan Log Out | Apabila pengguna telah selesai menggunakan sistem, pengguna dapat melakukan <i>log out</i> untuk menghindari penyalahgunaan akses oleh pihak yang tidak berwenang. |
| 3 | SIAKAD – F-03 | Melihat Kalender Akademik | Pengguna dapat melihat kalender akademik pada menu kalender akademik. |

2. Admin Sistem

Admin sistem diberikan akses-akses krusial terhadap sistem dikarenakan beberapa fungsi yang digunakan oleh pengguna lain hanya bisa diakses setelah dilakukan pengelolaan oleh admin sistem. Dalam sistem informasi admin sistem memiliki akses paling tinggi terhadap fungsionalitas yang ada pada aplikasi. Fungsi-fungsi yang diberikan untuk pengguna admin sistem dijabarkan lebih lanjut pada Tabel 4. 4.

Tabel 4. 4 Identifikasi Fitur Pengguna Admin Sistem

| No | Kode Fitur | Nama Fitur | Deskripsi |
|----|-----------------|------------------------|---|
| 1 | SIAKAD – F - 04 | Mengelola Data Jurusan | Pengguna dapat menambah, menghapus dan mengubah data jurusan yang ada pada sekolah. |

Tabel 4. 4 Identifikasi Fitur Pengguna Admin Sistem (Lanjutan)

| No | Kode Fitur | Nama Fitur | Deskripsi |
|----|-----------------|-------------------------------|--|
| 2 | SIAKAD – F - 05 | Mengelola Kelas | Pengguna dapat menambah dan menghapus kelas, mengatur wali kelas pada setiap kelasnya, dan mengatur penempatan kelas untuk siswa. |
| 3 | SIAKAD – F - 06 | Mengelola data mata pelajaran | Pengguna dapat menambah, menghapus dan mengubah, mata pelajaran pada sistem. |
| 4 | SIAKAD – F - 07 | Mengelola data guru | Pengguna dapat mengelola data guru, dan menentukan mata pelajaran yang diampu |
| 5 | SIAKAD – F - 08 | Mengelola data siswa | Pengguna dapat mengelola data diri siswa, dan pembagian kelas siswa. |
| 6 | SIAKAD – F - 09 | Mengelola jadwal pelajaran. | Pengguna dapat menambah, mengubah dan menghapus jadwal pelajaran pada setiap kelas yang ada serta guru pengampu mata pelajaran tersebut. |
| 7 | SIAKAD – F - 10 | Mengelola kalender akademik | Pengguna dapat mengatur kalender akademik secara berkala. |

3. Admin BKK

Fungsi-fungsinya yang diberikan pada admin BKK terbatas pada pengelolaan praktik kerja industri, dan mitra industri sekolah. Lebih lanjut fungsi-fungsi yang diberikan kepada admin BKK dapat dilihat pada Tabel 4.5.

Tabel 4. 5 Identifikasi Kebutuhan Fungsional Admin BKK

| No | Kode Fitur | Nama Fitur | Deskripsi |
|----|-----------------|--|--|
| 1 | SIAKAD – F - 11 | Melakukan validasi pelaksanaan praktik kerja industri siswa. | Pengguna dapat menerima atau menolak pengajuan pelaksanaan praktik kerja industri siswa. |
| 2 | SIAKAD - F- 12 | Mengelola Data Mitra Industri | Pengguna dapat melakukan baca, buat, ubah dan hapus data-data mengenai industry yang bermitra dengan SMK Muhammadiyah 1 Kemlagi. |
| 3 | SIAKAD – F – 13 | Melihat laporan pelaksanaan prakerin siswa. | Pengguna dapat melihat jumlah siswa yang sudah prakerin, sedang prakerin dan belum prakerin, serta tempat siswa melaksanakan prakerin. |

4. Waka Kurikulum

Pengguna waka kurikulum diberikan fungsi-fungsi yang berhubungan dengan pelaporan data akademik seperti pelaksanaan prakerin dan validasi akademik yang meliputi validasi kalender akademik, validasi penetapan wali kelas dan validasi kegiatan akademik, karena urusan teknis seperti penambahan dan penghapusan dilakukan oleh admin sistem. Fungsi-fungsi yang diberikan untuk waka kurikulum terdapat pada Tabel 4. 6

Tabel 4. 6 Identifikasi Kebutuhan Fungsional Waka Kurikulum

| No | Kode Fitur | Nama Fitur | Deskripsi |
|----|-----------------|---|--|
| 1 | SIAKAD – F - 13 | Melihat laporan pelaksanaan prakerin siswa. | Pengguna dapat melihat jumlah siswa yang sudah prakerin, sedang prakerin dan belum prakerin, serta tempat siswa melaksanakan prakerin. |

Tabel 4. 6 Identifikasi Kebutuhan Fungsional Waka Kurikulum (Lanjutan)

| No | Kode Fitur | Nama Fitur | Deskripsi |
|----|-----------------|-------------------------------|--|
| 2 | SIAKAD – F - 14 | Memvalidasi kegiatan akademik | Pengguna dapat melakukan validasi terhadap penerbitan kalender akademik yang telah dibuat oleh admin sistem. |
| 3 | SIAKAD – F - 15 | Memvalidasi Penetapan Kelas | Pengguna dapat melakukan validasi terhadap penetapan wali kelas yang telah dilakukan oleh admin akademik. |

5. Guru

Pengguna guru diberikan fungsi-fungsi yang berhubungan langsung dengan kegiatan belajar mengajar sehari-hari dan evaluasi pada siswa, lebih lanjut fungsi-fungsi yang diberikan pada pengguna Guru terdapat pada Tabel 4. 7.

Tabel 4. 7 Identifikasi Kebutuhan Fungsionalitas Pengguna Guru

| No | Kode Fitur | Nama Fitur | Deskripsi |
|----|-----------------|---------------------------------------|---|
| 1 | SIAKAD – F - 16 | Memperbarui data diri. | Pengguna dapat mengubah data pribadi dan melakukan pembaruan secara berkala terhadap data tersebut. |
| 2 | SIAKAD-F-17 | Melihat kelas yang diwakili. | Pengguna dapat melihat daftar kelas yang diajar saat ini. |
| 3 | SIAKAD – F – 18 | Memasukan nilai hasil evaluasi siswa. | Pengguna dapat mengelola hasil nilai mata pelajaran yang diampu pada jadwal yang telah ditentukan. |

Tabel 4.7 Kebutuhan Fungsionalitas Pengguna Guru (Lanjutan)

| No | Kode Fitur | Nama Fitur | Deskripsi |
|----|-----------------|--------------------------------------|---|
| 4 | SIAKAD – F - 20 | Mengubah nilai hasil evaluasi siswa. | Pengguna dapat melakukan perubahan pada nilai yang dimasukkan apabila diperlukan. |

6. Siswa

Pengguna siswa diberikan fungsi-fungsi yang berguna untuk pelaksanaan kegiatan akademik di sekolah, fungsi-fungsi tersebut dapat dilihat pada Tabel 4. 8.

Tabel 4.8 Identifikasi Kebutuhan Fungsionalitas Pengguna Siswa

| No | Kode Fitur | Nama Fitur | Deskripsi |
|----|-----------------|--|---|
| 1 | SIAKAD – F – 16 | Memperbarui Data Diri | Pengguna dapat mengubah data pribadi dan melakukan pembaruan secara berkala terhadap data tersebut. |
| 2 | SIAKAD-F- 21 | Melihat Detail Informasi Kelas. | Pengguna dapat melihat informasi kelas secara detail. |
| 3 | SIAKAD – F – 22 | Melihat nilai hasil evaluasi | Pengguna dapat melihat hasil dari pelaksanaan evaluasi tengah semester dan evaluasi akhir semester. |
| 4 | SIAKAD – F – 23 | Mengajukan penuntasan pelaksanaan prakerin | Pengguna dapat mengajukan pelaksanaan praktik kerja industri pada jadwal yang telah ditentukan, |
| 5 | SIAKAD – F – 24 | Mengubah Penuntasan pelaksanaan prakerin. | Pengguna dapat mengubah informasi pengajuan pelaksanaan prakerin apabila diperlukan. |

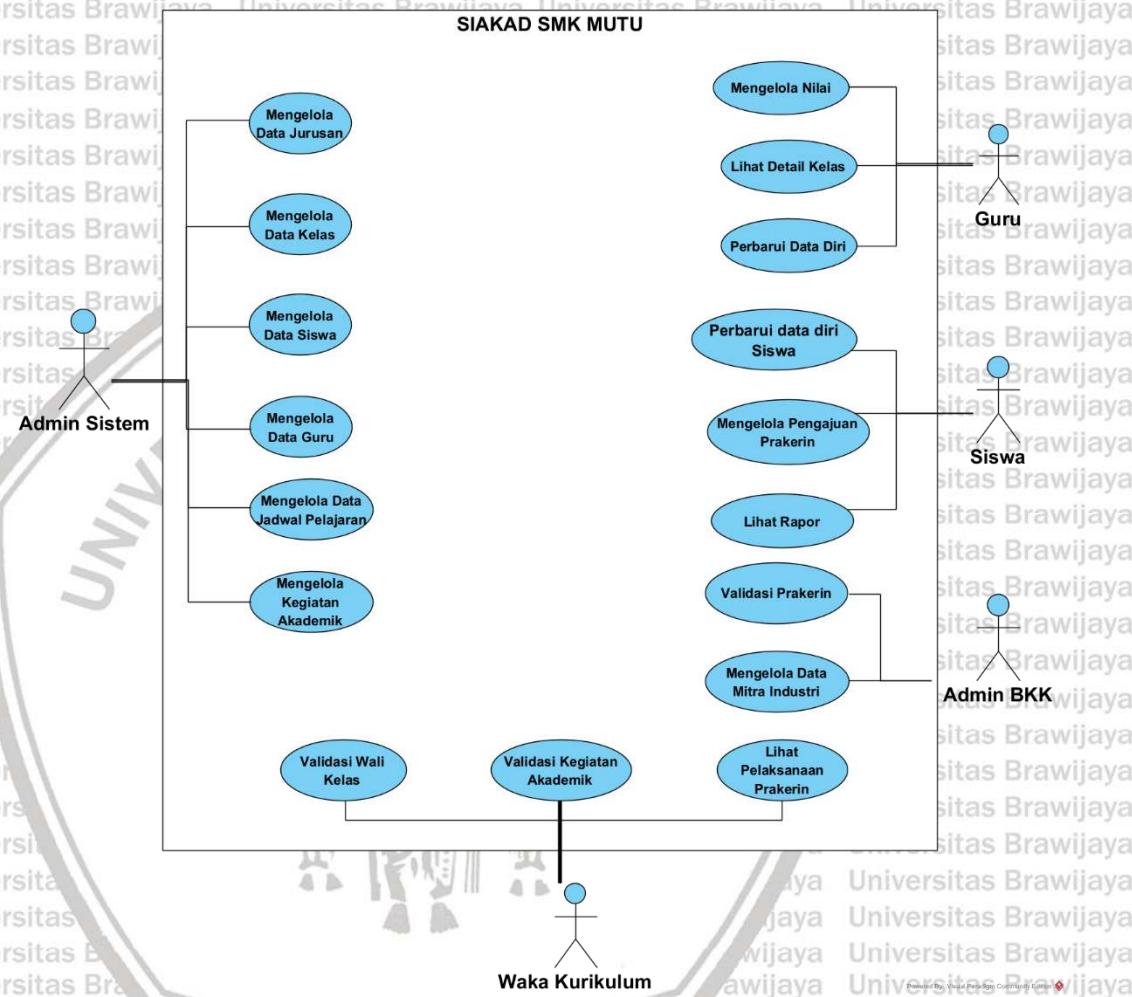
4.2 Perancangan

Pada tahap ini dilakukan pendefinisian terhadap komponen-komponen yang akan digunakan untuk membangun sistem informasi. Perencangan sistem pada penelitian ini meliputi *use case diagram*, *use case scenario*, *activity diagram*, perancangan arsitektur, *sequence diagram*, *class diagram*, perancangan *database (ERD)*, dan perancangan antarmuka sistem informasi. Hasil dalam perancangan

komponen-komponen tersebut akan dijadikan acuan dalam tahap implementasi sistem.

4.2.1 Use Case Diagram

Diagram *use case* pada Gambar 4.14 merepresentasikan bagaimana interaksi aktor yang meliputi admin sistem, admin BKK, Waka Kurikulum, Guru dan siswa dengan sistem informasi akademik SMK Muhammadiyah 1 Kemlagi.



Gambar 4. 14 Use Case Diagram

4.2.2 Use Case Scenario

Bagian ini merupakan penjabaran secara lebih terperinci dari masing-masing kebutuhan yang ada pada *use case* diagram, *use case scenario* dideskripsikan dalam bentuk tabel yang terdiri dari kode fungsional, deskripsi, kondisi awal (*pre-condition*), skenario utama, skenario alternatif, dan kondisi akhir.

1. Melakukan Login Sistem (SIAKAD-F-01)

Login sistem merupakan langkah awal dan wajib bagi semua pengguna sebelum mengakses fungsi-fungsi yang ada pada sistem. Dalam *use case* ini sistem akan melakukan pengecekan input oleh pengguna sehingga dapat diarahkan sesuai hak akses yang diberikan. Skenario dari *use case* ini dapat dilihat pada Tabel 4. 9.

Tabel 4. 9 Use Case Scenario Melakukan Login Sistem

| | |
|----------------------------|--|
| Kode Use Case | SIAKAD-F-01 |
| Deskripsi | Pada <i>use case</i> ini sistem akan melakukan verifikasi pengguna menggunakan <i>username</i> dan <i>password</i> . |
| Aktor | Semua Pengguna |
| Pre-condition | Aktor telah mengakses halaman login |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor memasukkan <i>username</i> pada <i>field</i> yang telah disediakan 2) Aktor memasukkan <i>password</i> pada <i>field</i> yang telah disediakan 3) Aktor menekan tombol <i>login</i> 4) Sistem melakukan pengecekan <i>username</i> dan <i>password</i> yang dimasukkan aktor apakah terdaftar pada <i>database</i>. 5) Sistem mengarahkan aktor pada halaman utama (dashboard). |
| Skenario alternatif | <p>a) <i>Username</i> atau <i>password</i> yang dimasukkan oleh aktor, tidak ditemukan di <i>database</i>.</p> <ul style="list-style-type: none"> • Sistem akan menampilkan informasi bahwa <i>username</i> atau <i>password</i> yang dimasukkan tidak dikenali. |
| Kondisi akhir | Aktor dapat mengakses halaman dashboard sistem. |

2. Melakukan Logout Sistem (SIAKAD-F-02)

Fungsi *logout* berguna untuk menghindari pengaksesan sistem dari pihak yang tidak dikehendaki. Setelah pengguna melakukan *logout* maka pengguna perlu memasukkan kembali data *username* dan *password* untuk masuk ke sistem. Skenario dari *use case* ini dapat dilihat pada Tabel 4. 10.

Tabel 4. 10 Use Case Scenario melakukan Logout Sistem

| | |
|----------------------------|---|
| Kode Use Case | SIAKAD-F-02 |
| Deskripsi | Pada <i>use case ini</i> sistem akan menghapus <i>session</i> dari aktor yang telah login. |
| Aktor | Semua Pengguna |
| Pre-condition | Aktor telah mengakses halaman dashboard. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik menu <i>logout</i> pada sidebar sistem informasi. 2) Sistem menghapus <i>session</i> aktor. 3) Sistem mengalihkan pada halaman <i>login</i> sistem. |
| Skenario alternatif | <p>A. Aktor menutup browser secara langsung.</p> <ul style="list-style-type: none"> • Aktor masih dapat mengakses sistem, selama perangkat yang digunakan tidak dilakukan <i>restart</i>. |
| Kondisi akhir | Aktor tidak dapat mengakses fungsionalitas sistem kembali sebelum melakukan login ulang. |

3. Melihat Kalender Akademik (SIAKAD-F-03)

Use case melihat kalender akademik memungkinkan aktor dapat melihat informasi kalender akademik yang diperbarui secara berkala oleh admin sistem. Skenario dari *use case* ini dapat dilihat pada Tabel 4. 11.

Tabel 4. 11 Use Case Scenario Melihat Kalender Akademik

| | |
|----------------------------|---|
| Kode Use Case | SIAKAD-F-03 |
| Deskripsi | Pada <i>use case ini</i> sistem akan menampilkan informasi kalender akademik pada aktor. |
| Aktor | Semua Pengguna |
| Pre-condition | Aktor telah mengakses halaman dashboard. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik menu kalender akademik yang ada pada sidebar sistem informasi. 2) Sistem menampilkan informasi kalender akademik |
| Skenario alternatif | Tidak ada |

| | |
|----------------------|--|
| Kondisi akhir | Aktor dapat melihat informasi kalender akademik pada sistem informasi. |
|----------------------|--|

4. Mengelola Data Jurusan (SIAKAD-F-04)

Pada *use case* ini memungkinkan aktor untuk melakukan kegiatan pengelolaan yang meliputi tambah, ubah, hapus dan lihat dari data jurusan dalam sistem. Fungsionalitas dispesifikasi lebih lanjut dalam sub-fungsional, hal tersebut dilakukan mengingat masing-masing sub-fungsional tersebut memiliki skenario yang berbeda-beda. Berikut penjelasan dari masing-masing sub-fungsionalitas dari *use case* mengelola data jurusan.

A. Melihat Data Jurusan (SIAKAD-F-04-A)

Sub-fungsional melihat data jurusan memungkinkan aktor untuk melihat data jurusan yang disimpan dalam *database* sistem informasi.

Skenario dapat dilihat pada Tabel 4. 12.

Tabel 4. 12 Use Case Scenario Melihat Data Jurusan

| | |
|----------------------------|--|
| Kode Use Case | SIAKAD-F-04-A |
| Deskripsi | Pada <i>use case</i> ini sistem akan menampilkan informasi jurusan yang tersimpan di <i>database</i> pada aktor. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman dashboard dan login sebagai admin sistem. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik menu jurusan yang ada pada sidebar sistem informasi. 2) Sistem mengarahkan user ke halaman jurusan. 3) Sistem menampilkan data jurusan yang tersimpan di <i>database</i> |
| Skenario alternatif | Tidak ada |
| Kondisi akhir | Aktor dapat melihat informasi jurusan yang tersimpan pada <i>database</i> . |

B. Menambah Data Jurusan (SIAKAD-F-04-B)

Sub-fungsional menambah data jurusan memungkinkan aktor untuk menambahkan data jurusan baru yang akan disimpan dalam *database* sistem informasi. Skenario dapat dilihat pada Tabel 4. 13.

Tabel 4. 13 Use Case Scenario Menambah Data Jurusan

| | |
|----------------------------|---|
| Kode Use Case | SIAKAD-F-04-B |
| Deskripsi | Pada <i>use case ini</i> sistem akan melakukan <i>query</i> untuk menambahkan data jurusan yang tersimpan di database. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman jurusan dan login sebagai admin sistem. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik tombol tambah yang terdapat atas tabel jurusan. 2) Sistem mengarahkan aktor pada halaman tambah jurusan. 3) Aktor mengisi data jurusan yang akan ditambahkan sesuai dengan <i>field</i> yang disediakan. 4) Aktor menekan tombol "simpan". 5) Sistem melakukan penambahan data ke dalam <i>database</i> lalu menampilkan pesan sukses. 6) Sistem mengarahkan aktor ke halaman jurusan. |
| Skenario alternatif | <p>a) Terdapat <i>field</i> yang kosong.</p> <ul style="list-style-type: none"> • Sistem akan menampilkan informasi bahwa terdapat <i>field</i> yang perlu diisi. <p>b) Data yang dimasukan tidak valid</p> <ul style="list-style-type: none"> • Sistem akan menampilkan informasi bahwa data yang dimasukkan tidak valid. |
| Kondisi akhir | Aktor dapat melihat informasi jurusan telah bertambah sesuai dengan data yang dimasukkan. |

C. Mengubah Data Jurusan (SIAKAD-F-04-C)

Sub-fungsional mengubah data jurusan memungkinkan aktor untuk melakukan perubahan terhadap data jurusan yang sudah ada. Skenario dapat dilihat pada Tabel 4. 14.

Tabel 4. 14 Use Case Scenario Mengubah Data Jurusan

| | |
|----------------------------|---|
| Kode Use Case | SIAKAD-F-04-C |
| Deskripsi | Pada <i>use case</i> ini sistem akan melakukan <i>query</i> untuk melakukan perubahan terhadap data jurusan yang tersimpan di database sesuai dengan <i>id</i> yang dipilih oleh aktor. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman jurusan dan login sebagai admin sistem. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik ikon bergambar “pensil” yang ada pada baris data jurusan yang akan diubah datanya. 2) Sistem mengarahkan aktor pada halaman ubah jurusan. 3) Aktor mengisi data jurusan yang akan diubah sesuai dengan field yang disediakan. 4) Aktor menekan tombol “ubah”. 5) Sistem melakukan penambahan data ke dalam database lalu menampilkan pesan sukses. 6) Sistem mengarahkan aktor ke halaman jurusan. |
| Skenario alternatif | <ul style="list-style-type: none"> a) Data yang dimasukkan tidak valid <ul style="list-style-type: none"> • Sistem akan menampilkan keterangan bahwa data yang dimasukkan tidak valid. b) Aktor menekan tombol batal <ul style="list-style-type: none"> • Sistem mengarahkan ke halaman jurusan awal. |
| Kondisi akhir | Aktor dapat melihat informasi jurusan telah berubah sesuai dengan data yang dimasukkan. |

D. Menghapus Data Jurusan (SIAKAD-F-04-D)

Sub-fungsional mengubah data jurusan memungkinkan aktor untuk melakukan penghapusan terhadap data jurusan yang sudah ada. Skenario dapat dilihat pada Tabel 4. 15.

Tabel 4. 15 Use Case Scenario Menghapus Data Jurusan

| | |
|----------------------------|--|
| Kode Use Case | SIAKAD-F-04-D |
| Deskripsi | Pada <i>use case</i> ini sistem akan melakukan <i>query</i> untuk melakukan penghapusan terhadap data jurusan yang tersimpan di database sesuai dengan <i>id</i> yang dipilih oleh aktor. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman jurusan dan login sebagai admin sistem. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik ikon bergambar “tong sampah” yang ada pada baris data jurusan yang akan diubah datanya. 2) Sistem mengarahkan aktor pada halaman konfirmasi penghapusan. 3) Aktor mengklik tombol “ya”. 4) Sistem melakukan penghapusan sesuai dengan <i>id</i> di database lalu menampilkan pesan sukses. 5) Sistem mengarahkan aktor ke halaman jurusan. |
| Skenario alternatif | <p>a) Aktor menekan tombol batal</p> <ul style="list-style-type: none"> • Sistem mengarahkan ke halaman jurusan awal. |
| Kondisi akhir | Aktor dapat melihat informasi jurusan telah berubah sesuai dengan tindakan penghapusan. |

5. Mengelola Data Kelas (SIAKAD-F-05)

Pada *use case* ini memungkinkan aktor untuk melakukan kegiatan pengelolaan yang meliputi tambah, ubah, hapus dan lihat dari data kelas dalam sistem. Fungsionalitas dispesifikasi lebih lanjut dalam sub-fungsional, hal tersebut dilakukan mengingat masing-masing sub-fungsional tersebut memiliki skenario yang berbeda-beda. Berikut penjelasan dari masing-masing sub-fungsionalitas dari *use case* mengelola data kelas.

A. Melihat Data Kelas (SIAKAD-F-05-A)

Sub-fungsional melihat data kelas memungkinkan aktor untuk melihat data kelas yang disimpan dalam *database* sistem informasi. Skenario dapat dilihat pada Tabel 4. 16.

| Kode Use Case | SIAKAD-F-05-A |
|---------------------|---|
| Deskripsi | Pada <i>use case ini</i> sistem akan menampilkan informasi kelas yang tersimpan di database pada aktor. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman dashboard dan login sebagai admin sistem. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik menu kelas yang ada pada sidebar sistem informasi. 2) Sistem mengarahkan user ke halaman kelas. 3) Sistem menampilkan data kelas yang tersimpan di <i>database</i>. |
| Skenario alternatif | Tidak ada |
| Kondisi akhir | Aktor dapat melihat informasi kelas yang tersimpan pada database. |

B. Menambah Data Kelas(SIAKAD-F-05-B)

Sub-fungsional menambah data kelas memungkinkan aktor untuk menambahkan data kelas baru yang akan disimpan dalam database sistem informasi. Skenario dapat dilihat pada Tabel 4. 17.

Tabel 4. 17 Use Case Scenario Menambah Data Kelas

| Kode Use Case | SIAKAD-F-05-B |
|----------------|---|
| Deskripsi | Pada <i>use case ini</i> sistem akan melakukan <i>query</i> untuk menambahkan data kelas yang tersimpan di database. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman kelas, dan login sebagai admin sistem. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik tombol tambah yang terdapat atas tabel kelas. 2) Sistem mengarahkan aktor pada halaman tambah kelas. |

| | | |
|----------------------------|-------------------------------------|---|
| | | <p>3) Aktor mengisi data Mekelas yang akan ditambahkan sesuai dengan field yang disediakan.</p> <p>4) Aktor menekan tombol “simpan”.</p> <p>5) Sistem melakukan penambahan data ke dalam database lalu menampilkan pesan sukses.</p> <p>6) Sistem mengarahkan aktor ke halaman kelas.</p> |
| Skenario alternatif | a) Form Tidak Lengkap | <ul style="list-style-type: none"> Sistem menampilkan keterangan bahwa <i>field</i> perlu diisi. |
| Kondisi akhir | b) Data yang dimasukkan tidak valid | <ul style="list-style-type: none"> Sistem akan menampilkan keterangan bahwa data yang dimasukkan tidak valid. |

C. Mengubah Data Kelas (SIAKAD-F-05-C)

Sub-fungsional mengubah data kelas memungkinkan aktor untuk melakukan perubahan terhadap data kelas yang sudah ada. Skenario dapat dilihat pada Tabel 4. 18.

Tabel 4. 18 Use Case Scenario Mengubah Data Kelas

| | |
|-----------------------|---|
| Kode Use Case | SIAKAD-F-05-C |
| Deskripsi | Pada <i>use case</i> ini sistem akan melakukan <i>query</i> untuk melakukan perubahan terhadap data kelas yang tersimpan di database sesuai dengan <i>id</i> yang dipilih oleh aktor. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman kelas dan login sebagai admin sistem. |
| Skenario Utama | <p>1) Aktor mengklik ikon bergambar “pensil” yang ada pada baris data kelas yang akan diubah datanya.</p> <p>2) Sistem mengarahkan aktor pada halaman ubah kelas.</p> |

| | | |
|----------------------------|--|---|
| | | <ul style="list-style-type: none"> 3) Aktor mengisi data kelas yang akan diubah sesuai dengan field yang disediakan. 4) Aktor menekan tombol “ubah”. 5) Sistem melakukan pengubahan data ke dalam database lalu menampilkan pesan sukses. 6) Sistem mengarahkan aktor ke halaman kelas. |
| Skenario alternatif | | <ul style="list-style-type: none"> a) Data yang dimasukkan tidak valid <ul style="list-style-type: none"> • Sistem akan menampilkan keterangan bahwa data yang dimasukkan tidak valid. b) Aktor menekan tombol batal <ul style="list-style-type: none"> • Sistem mengarahkan ke halaman kelas awal. |
| Kondisi akhir | | Aktor dapat melihat informasi kelas telah berubah sesuai dengan data yang dimasukkan. |

D. Menghapus Data Kelas (SIAKAD-F-05-D)

Sub-fungsional menghapus data kelas memungkinkan aktor untuk melakukan penghapusan terhadap data kelas yang sudah ada. Skenario dapat dilihat pada Tabel 4. 19.

Tabel 4. 19 Use Case Scenario Menghapus Data Kelas

| | |
|-----------------------|---|
| Kode Use Case | SIAKAD-F-05-D |
| Deskripsi | Pada <i>use case ini</i> sistem akan melakukan <i>query</i> untuk melakukan penghapusan terhadap data kelas yang tersimpan di database sesuai dengan <i>id</i> yang dipilih oleh aktor. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman kelas dan login sebagai admin sistem. |
| Skenario Utama | <ul style="list-style-type: none"> 1) Aktor mengklik ikon bergambar “tong sampah” yang ada pada baris data kelas yang akan diubah datanya. 2) Sistem mengarahkan aktor pada halaman konfirmasi penghapusan. |

| | |
|----------------------------|---|
| | <p>3) Aktor mengklik tombol “ya”.</p> <p>4) Sistem melakukan penghapusan sesuai dengan id di database lalu menampilkan pesan sukses.</p> <p>5) Sistem mengarahkan aktor ke halaman kelas.</p> |
| Skenario alternatif | <p>a) Aktor menekan tombol batal</p> <ul style="list-style-type: none"> • Sistem mengarahkan ke halaman kelas awal. |
| Kondisi akhir | Aktor dapat melihat informasi kelas telah berubah sesuai dengan tindakan penghapusan. |

6. Mengelola Data Mata Pelajaran (SIAKAD-F-06)

Pada *use case* ini memungkinkan aktor untuk melakukan kegiatan pengelolaan yang meliputi tambah, ubah, hapus dan lihat data mata pelajaran dalam sistem. Fungsionalitas dispesifikasi lebih lanjut dalam sub-fungsional, hal tersebut dilakukan mengingat masing-masing sub-fungsional tersebut memiliki skenario yang berbeda-beda. Berikut penjelasan dari masing-masing sub-fungsionalitas dari *use case* mengelola data mata pelajaran.

A. Melihat Data Mata Pelajaran Berdasarkan Jurusan (SIAKAD-F-06-A)

Sub-fungsional melihat data kelas memungkinkan aktor untuk melihat data mata pelajaran yang disimpan dalam *database* sistem informasi sesuai dengan jurusan yang dipilih. Skenario dapat dilihat pada Tabel 4. 20.

Tabel 4. 20 Use Case Scenario Melihat Data Mata Pelajaran Berdasarkan Jurusan

| | |
|-----------------------|--|
| Kode Use Case | SIAKAD-F-06-A |
| Deskripsi | Pada <i>use case</i> ini, sistem akan menampilkan informasi mata pelajaran yang tersimpan di database pada aktor. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman dashboard dan login sebagai admin sistem. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik menu mata pelajaran yang ada pada sidebar sistem informasi. 2) Sistem mengarahkan user ke halaman mata pelajaran. |

| | |
|----------------------------|---|
| | <p>3) Aktor mengklik pilihan jurusan yang ada diatas tabel mata pelajaran.</p> <p>4) Aktor mengklik tombol “submit”.</p> <p>5) Sistem menampilkan daftar data mata pelajaran berdasarkan jurusan dipilih.</p> |
| Skenario alternatif | Tidak ada |
| Kondisi akhir | Aktor dapat melihat informasi mata pelajaran yang tersimpan pada database berdasarkan jurusan yang dipilih. |

B. Menambah Data Mata Pelajaran (SIAKAD-F-06-B)

Sub-fungsional menambah data mata pelajaran memungkinkan aktor untuk menambahkan data mata pelajaran baru yang akan disimpan dalam database sistem informasi. Skenario dapat dilihat pada Tabel 4. 21.

Tabel 4. 21 Use Case Scenario Menambah Data Mata Pelajaran

| | |
|-----------------------|---|
| Kode Use Case | SIAKAD-F-06-B |
| Deskripsi | Pada <i>use case ini</i> sistem akan melakukan <i>query</i> untuk menambahkan data mata pelajaran yang tersimpan di database. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman kelas, dan login sebagai admin sistem. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik tombol tambah yang terdapat diatas tabel mata pelajaran. 2) Sistem mengarahkan aktor pada halaman tambah mata pelajaran. 3) Aktor mengisi data mata pelajaran yang akan ditambahkan sesuai dengan field yang disediakan. 4) Aktor menekan tombol “simpan”. 5) Sistem melakukan penambahan data ke dalam database lalu menampilkan pesan sukses. 6) Sistem mengarahkan aktor ke halaman mata pelajaran. |

| | |
|----------------------------|---|
| Skenario alternatif | <p>a) Form Tidak Lengkap</p> <ul style="list-style-type: none"> • Sistem akan menampilkan keterangan bahwa <i>field</i> perlu diisi. <p>b) Data yang dimasukkan tidak valid</p> <ul style="list-style-type: none"> • Sistem akan menampilkan keterangan bahwa data yang dimasukkan tidak valid. |
| Kondisi akhir | Aktor dapat melihat informasi mata pelajaran telah bertambah sesuai dengan data yang dimasukkan. |

C. Mengubah Data Mata Pelajaran (SIAKAD-F-06-C)

Sub-fungsional mengubah data mata pelajaran memungkinkan aktor untuk melakukan perubahan terhadap data mata pelajaran yang sudah ada. Skenario dapat dilihat pada Tabel 4. 22.

Tabel 4. 22 Use Case Scenario Mengubah Data Mata Pelajaran

| | |
|-----------------------|--|
| Kode Use Case | SIAKAD-F-06-C |
| Deskripsi | Pada <i>use case ini</i> sistem akan melakukan <i>query</i> untuk melakukan perubahan terhadap data mata pelajaran yang tersimpan di database sesuai dengan <i>id</i> yang dipilih oleh aktor. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman kelas dan login sebagai admin sistem. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik ikon bergambar "pensil" yang ada pada baris data mata pelajaran yang akan diubah datanya. 2) Sistem mengarahkan aktor pada halaman ubah mata pelajaran. 3) Aktor mengisi data mata pelajaran yang akan diubah sesuai dengan <i>field</i> yang disediakan. 4) Aktor menekan tombol "ubah". 5) Sistem melakukan pengubahan data ke dalam database lalu menampilkan pesan sukses. |

| | | | | | |
|--|--|----------------------------|--|--|---|
| | | | | | 6) Sistem mengarahkan aktor ke halaman kelas. |
| | | Skenario alternatif | a) Data yang dimasukkan tidak valid Sistem akan menampilkan keterangan bahwa data yang dimasukkan tidak valid. b) Aktor menekan tombol batal Sistem mengarahkan ke halaman mata pelajaran awal. | | |
| | | Kondisi akhir | Aktor dapat melihat informasi jurusan telah berubah sesuai dengan data yang dimasukkan. | | |

D. Menghapus Data Mata Pelajaran (SIAKAD-F-06-D)

Sub-fungsional menghapus data mata pelajaran memungkinkan aktor untuk melakukan penghapusan terhadap data mata pelajaran yang sudah ada. Skenario dapat dilihat pada Tabel 4. 23.

Tabel 4. 23 Use Case Scenario Menghapus Data Mata Pelajaran

| | |
|----------------------------|--|
| Kode Use Case | SIAKAD-F-06-D |
| Deskripsi | Pada <i>use case ini</i> sistem akan melakukan <i>query</i> untuk melakukan penghapusan terhadap data mata pelajaran yang tersimpan di database sesuai dengan <i>id</i> yang dipilih oleh aktor. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman mata pelajaran dan login sebagai admin sistem. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik ikon bergambar “tong sampah” yang ada pada baris data mata pelajaran yang akan diubah datanya. 2) Sistem mengarahkan aktor pada halaman konfirmasi penghapusan. 3) Aktor mengklik tombol “ya”. 4) Sistem melakukan penghapusan sesuai dengan <i>id</i> di database lalu menampilkan pesan sukses. 5) Sistem mengarahkan aktor ke halaman mata pelajaran. |
| Skenario alternatif | <ol style="list-style-type: none"> a) Aktor menekan tombol batal |

| | |
|----------------------|--|
| Kondisi awal | <ul style="list-style-type: none"> Sistem mengarahkan ke halaman mata pelajaran awal. |
| Kondisi akhir | Aktor dapat melihat informasi mata pelajaran telah berubah sesuai dengan tindakan penghapusan. |

7. Mengelola Data Guru (SIAKAD-F-07)

Pada *use case* ini memungkinkan aktor untuk melakukan kegiatan pengelolaan yang meliputi tambah, ubah, hapus dan lihat dari data guru dalam sistem. Fungsionalitas dispesifikasi lebih lanjut dalam sub-fungsional, hal tersebut dilakukan mengingat masing-masing sub-fungsional tersebut memiliki skenario yang berbeda-beda. Berikut penjelasan dari masing-masing sub-fungsionalitas dari *use case* mengelola data guru.

A. Melihat Data Guru (SIAKAD-F-07-A)

Sub-fungsional melihat data guru memungkinkan aktor untuk melihat data guru yang disimpan dalam *database* sistem informasi. Skenario dapat dilihat pada Tabel 4. 24.

Tabel 4. 24 *Use Case Scenario* Melihat Data Guru

| | |
|----------------------------|--|
| Kode Use Case | SIAKAD-F-07-A |
| Deskripsi | Pada <i>use case</i> ini sistem akan menampilkan informasi guru yang tersimpan di <i>database</i> pada aktor. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman dashboard dan login sebagai admin sistem. |
| Skenario Utama | <ol style="list-style-type: none"> Aktor mengklik menu guru yang ada pada sidebar sistem informasi. Sistem mengarahkan user ke halaman guru. Sistem menampilkan daftar data guru yang tersimpan pada <i>database</i>. |
| Skenario alternatif | Tidak ada |
| Kondisi akhir | Aktor dapat melihat daftar guru yang tersimpan di <i>database</i> . |

B. Melihat Detail Guru (SIAKAD-F-07-B)

Sub-fungsional melihat detail guru memungkinkan aktor untuk melihat informasi lebih lanjut mengenai guru secara spesifik. Skenario dapat dilihat pada Tabel 4. 25.

Tabel 4. 25 Use Scenario Melihat Detail Guru

| | |
|----------------------------|---|
| Kode Use Case | SIAKAD-F-07-B |
| Deskripsi | Pada <i>use case ini</i> sistem akan melakukan <i>query</i> untuk menampilkan data lengkap guru di database sesuai id yang dipilih. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman guru dan login sebagai admin sistem. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik tombol “detail” yang ada pada baris data guru yang akan dilihat data lengkapnya. 2) Sistem mengarahkan aktor pada halaman detail guru. 3) Sistem menampilkan data lengkap guru sesuai dengan <i>id</i>. |
| Skenario alternatif | Tidak ada |
| Kondisi akhir | Aktor dapat melihat detail informasi guru sesuai dengan <i>id</i> yang dipilih. |

C. Menambah Data Guru (SIAKAD-F-07-C)

Sub-fungsional menambah data guru memungkinkan aktor untuk menambahkan data guru baru yang akan disimpan dalam database sistem informasi. Skenario dapat dilihat pada Tabel 4. 26.

Tabel 4. 26 Use Case Scenario Menambah Data Guru

| | |
|-----------------------|---|
| Kode Use Case | SIAKAD-F-07-C |
| Deskripsi | Pada <i>use case ini</i> sistem akan melakukan <i>query</i> untuk menambahkan data guru yang tersimpan di <i>database</i> . |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman guru, dan login sebagai admin sistem. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik tombol tambah yang terdapat diatas tabel daftar guru. |

| | |
|----------------------------|--|
| | <p>2) Sistem mengarahkan aktor pada halaman tambah guru.</p> <p>3) Aktor mengisi data guru yang akan ditambahkan sesuai dengan field yang disediakan.</p> <p>4) Aktor menekan tombol “simpan”.</p> <p>5) Sistem melakukan penambahan data ke dalam database lalu menampilkan pesan sukses.</p> <p>6) Sistem mengarahkan aktor ke halaman guru.</p> |
| Skenario alternatif | <p>a) Form Tidak Lengkap</p> <ul style="list-style-type: none"> • Sistem menampilkan keterangan bahwa field perlu diisi. <p>b) Data yang dimasukkan tidak valid</p> <ul style="list-style-type: none"> • Sistem akan menampilkan keterangan bahwa data yang dimasukkan tidak valid. |
| Kondisi akhir | Aktor dapat melihat daftar guru telah bertambah sesuai dengan data yang dimasukkan. |

D. Mengubah Data Guru (SIAKAD-F-07-D)

Sub-fungsional mengubah data guru memungkinkan aktor untuk melakukan perubahan terhadap data guru yang sudah ada. Skenario dapat dilihat pada Tabel 4. 27.

Tabel 4. 27 Use Case Scenario Mengubah Data Guru

| | |
|-----------------------|--|
| Kode Use Case | SIAKAD-F-07-D |
| Deskripsi | Pada <i>use case ini</i> sistem akan melakukan <i>query</i> untuk melakukan perubahan terhadap data guru yang tersimpan di database sesuai dengan <i>id</i> yang dipilih oleh aktor. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman guru dan login sebagai admin sistem. |
| Skenario Utama | <p>1) Aktor mengklik ikon bergambar “pensil” yang ada pada baris data guru yang akan diubah datanya.</p> |

| | |
|----------------------------|---|
| | <p>2) Sistem mengarahkan aktor pada halaman ubah data guru.</p> <p>3) Aktor mengisi data guru yang akan diubah sesuai dengan field yang disediakan.</p> <p>4) Aktor menekan tombol “ubah”.</p> <p>5) Sistem melakukan pengubahan data ke dalam database lalu menampilkan pesan sukses.</p> <p>6) Sistem mengarahkan aktor ke halaman data guru.</p> |
| Skenario alternatif | <p>a) Data yang dimasukkan tidak valid Sistem akan menampilkan keterangan bahwa data yang dimasukkan tidak valid.</p> <p>b) Aktor menekan tombol batal Sistem mengarahkan ke halaman data guru awal.</p> |
| Kondisi akhir | Aktor dapat melihat informasi guru telah berubah sesuai dengan data yang dimasukkan. |

E. Menghapus Data Guru (SIAKAD-F-07-E)

Sub-fungsional menghapus data guru memungkinkan aktor untuk melakukan penghapusan terhadap data guru yang sudah ada. Skenario dapat dilihat pada Tabel 4. 28.

Tabel 4. 28 Use Case Scenario Menghapus Data Guru

| | |
|-----------------------|--|
| Kode Use Case | SIAKAD-F-07-E |
| Deskripsi | Pada <i>use case</i> ini sistem akan melakukan <i>query</i> untuk melakukan penghapusan terhadap data guru yang tersimpan di database sesuai dengan <i>id</i> yang dipilih oleh aktor. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman guru dan login sebagai admin sistem. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik ikon bergambar “tong sampah” yang ada pada baris data guru yang akan diubah datanya. 2) Sistem mengarahkan aktor pada halaman konfirmasi penghapusan. |

| | |
|----------------------------|--|
| | <p>3) Aktor mengklik tombol “ya”.</p> <p>4) Sistem melakukan penghapusan sesuai dengan id di database lalu menampilkan pesan sukses.</p> <p>5) Sistem mengarahkan aktor ke halaman guru.</p> |
| Skenario alternatif | <p>a) Aktor menekan tombol batal</p> <ul style="list-style-type: none"> • Sistem mengarahkan ke halaman guru awal. |
| Kondisi akhir | Aktor dapat melihat informasi data guru telah berubah sesuai dengan tindakan penghapusan. |

8. Mengelola Data Siswa (SIAKAD-F-08)

Pada *use case* ini memungkinkan aktor untuk melakukan kegiatan pengelolaan yang meliputi tambah, ubah, hapus dan lihat dari data siswa dalam sistem. Fungsionalitas dispesifikasiakan lebih lanjut dalam sub-fungsional, hal tersebut dilakukan mengingat masing-masing sub-fungsional tersebut memiliki skenario yang berbeda-beda. Berikut penjelasan dari masing-masing sub-fungsionalitas dari *use case* mengelola data siswa.

A. Melihat Data Siswa (SIAKAD-F-08-A)

Sub-fungsional melihat data siswa memungkinkan aktor untuk melihat data siswa yang disimpan dalam *database* sistem informasi. Skenario dapat dilihat pada Tabel 4. 29.

Tabel 4. 29 Use Case Scenario Melihat Data Siswa

| | |
|-----------------------|--|
| Kode Use Case | SIAKAD-F-08-A |
| Deskripsi | Pada <i>use case ini</i> sistem akan menampilkan informasi siswa yang tersimpan di database pada aktor. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman dashboard dan login sebagai admin sistem. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik menu siswa yang ada pada sidebar sistem informasi. 2) Sistem mengarahkan user ke halaman siswa. 3) Sistem menampilkan daftar data siswa. |

| | |
|----------------------------|--|
| Skenario alternatif | Tidak ada |
| Kondisi akhir | Aktor dapat melihat daftar siswa yang tersimpan di <i>database</i> . |

B. Melihat Detail Siswa (SIAKAD-F-08-B)

Sub-fungsional melihat detail siswa memungkinkan aktor untuk melihat informasi lebih lanjut mengenai siswa secara spesifik. Skenario dapat dilihat pada Tabel 4. 30.

Tabel 4. 30 Use Case Scenario Melihat Detail Siswa

| | |
|----------------------------|--|
| Kode Use Case | SIAKAD-F-08-B |
| Deskripsi | Pada <i>use case ini</i> sistem akan melakukan <i>query</i> untuk menampilkan data lengkap guru di database sesuai id yang dipilih. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman siswa dan login sebagai admin sistem. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik tombol “detail” yang ada pada baris data siswa yang akan dilihat data lengkapnya. 2) Sistem mengarahkan aktor pada halaman detail siswa. 3) Sistem menampilkan data lengkap siswa sesuai dengan <i>id</i>. |
| Skenario alternatif | Tidak ada |
| Kondisi akhir | Aktor dapat melihat detail informasi siswa sesuai dengan <i>id</i> yang dipilih. |

C. Menambah Data Siswa (SIAKAD-F-08-C)

Sub-fungsional menambah data siswa memungkinkan aktor untuk

menambahkan data siswa baru yang akan disimpan dalam database sistem informasi. Skenario dapat dilihat pada Tabel 4. 31.

Tabel 4. 31 Use Case Scenario Menambah Data Siswa

| | |
|----------------------------|--|
| Kode Use Case | SIAKAD-F-08-C |
| Deskripsi | Pada <i>use case ini</i> sistem akan melakukan <i>query</i> untuk menambahkan data siswa yang tersimpan di database. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman siswa, dan login sebagai admin sistem. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik tombol tambah yang terdapat diatas tabel daftar siswa. 2) Sistem mengarahkan aktor pada halaman tambah siswa. 3) Aktor mengisi data siswa yang akan ditambahkan sesuai dengan field yang disediakan. 4) Aktor menekan tombol "simpan". 5) Sistem melakukan penambahan data ke dalam database lalu menampilkan pesan sukses. 6) Sistem mengarahkan aktor ke halaman siswa. |
| Skenario alternatif | <ol style="list-style-type: none"> a) Form Tidak Lengkap <ul style="list-style-type: none"> • Sistem menampilkan keterangan bahwa field perlu diisi. b) Data yang dimasukkan tidak valid <ul style="list-style-type: none"> • Sistem akan menampilkan keterangan bahwa data yang dimasukkan tidak valid. |
| Kondisi akhir | Aktor dapat melihat daftar siswa telah bertambah sesuai dengan data yang dimasukkan. |

D. Mengubah Data Siswa (SIAKAD-F-08-D)

Sub-fungsional mengubah data siswa memungkinkan aktor untuk melakukan perubahan terhadap data siswa yang sudah ada. Skenario dapat dilihat pada Tabel 4. 32.

Tabel 4. 32 Use Case Scenario Mengubah Data Siswa

| | |
|----------------------------|---|
| Kode Use Case | SIAKAD-F-07-D |
| Deskripsi | Pada <i>use case</i> ini sistem akan melakukan <i>query</i> untuk melakukan perubahan terhadap data siswa yang tersimpan di database sesuai dengan <i>id</i> yang dipilih oleh aktor. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman siswa dan login sebagai admin sistem. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik ikon bergambar “pensil” yang ada pada baris data siswa yang akan diubah datanya. 2) Sistem mengarahkan aktor pada halaman ubah data siswa. 3) Aktor mengisi data siswa yang akan diubah sesuai dengan field yang disediakan. 4) Aktor menekan tombol “ubah”. 5) Sistem melakukan pengubahan data ke dalam database lalu menampilkan pesan sukses. 6) Sistem mengarahkan aktor ke halaman data siswa. |
| Skenario alternatif | <ol style="list-style-type: none"> a) Data yang dimasukkan tidak valid Sistem akan menampilkan keterangan bahwa data yang dimasukkan tidak valid. b) Aktor menekan tombol batal Sistem mengarahkan ke halaman data siswa awal. |
| Kondisi akhir | Aktor dapat melihat informasi siswa telah berubah sesuai dengan data yang dimasukkan. |

E. Menghapus Data Siswa (SIAKAD-F-08-E)

Sub-fungsional menghapus data siswa memungkinkan aktor untuk melakukan penghapusan terhadap data siswa yang sudah ada. Skenario dapat dilihat pada Tabel 4. 33.

| Kode Use Case | SIAKAD-F-08-E |
|---------------------|---|
| Deskripsi | Pada <i>use case</i> ini sistem akan melakukan <i>query</i> untuk melakukan penghapusan terhadap data siswa yang tersimpan di database sesuai dengan <i>id</i> yang dipilih oleh aktor. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman siswa dan login sebagai admin sistem. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik ikon bergambar “tong sampah” yang ada pada baris data siswa yang akan diubah datanya. 2) Sistem mengarahkan aktor pada halaman konfirmasi penghapusan. 3) Aktor mengklik tombol “ya” 4) Sistem melakukan penghapusan sesuai dengan <i>id</i> di database lalu menampilkan pesan sukses. 5) Sistem mengarahkan aktor ke halaman siswa. |
| Skenario alternatif | <ol style="list-style-type: none"> a) Aktor menekan tombol batal <ul style="list-style-type: none"> • Sistem mengarahkan ke halaman siswa awal. |
| Kondisi akhir | Aktor dapat melihat informasi data siswa telah berubah sesuai dengan tindakan penghapusan. |

F. Mencari Data Siswa (SIAKAD-F-08-F)

Sub-fungsional mencari data siswa memungkinkan aktor untuk melakukan pencarian terhadap data siswa berdasarkan kode unik (NISN) siswa. Skenario dapat dilihat pada Tabel 4. 34.

Tabel 4. 34 Use Case Scenario Mencari Data Siswa

| Kode Use Case | SIAKAD-F-08-F |
|---------------|---|
| Deskripsi | Pada <i>use case</i> ini sistem akan melakukan <i>query</i> untuk melakukan pencarian terhadap data siswa yang tersimpan di database sesuai dengan <i>id</i> yang dipilih oleh aktor. |
| Aktor | Admin Sistem |

| | |
|----------------------------|---|
| Pre-condition | Aktor telah mengakses halaman siswa dan login sebagai admin sistem. |
| Skenario Utama | <p>1) Aktor mengisi field cari data yang ada diatas tabel daftar siswa.</p> <p>2) Aktor mengklik tombol cari.</p> <p>3) Sistem melakukan query ke database untuk melakukan pencarian data.</p> <p>4) Sistem menampilkan data siswa yang dicari.</p> |
| Skenario alternatif | <p>a) Data yang dicari tidak ada</p> <ul style="list-style-type: none"> Sistem menampilkan keterangan bahwa data yang dicari tidak ditemukan. |
| Kondisi akhir | Aktor dapat melihat data siswa sesuai dengan nama yang dimasukkan. |

9. Mengelola Data Jadwal Pelajaran (SIAKAD-F-09)

Pada *use case* ini memungkinkan aktor untuk melakukan kegiatan pengelolaan yang meliputi tambah, ubah, hapus dan lihat data jadwal pelajaran dalam sistem. Fungsionalitas dispesifikasikan lebih lanjut dalam sub-fungsional, hal tersebut dilakukan mengingat masing-masing sub-fungsional tersebut memiliki skenario yang berbeda-beda. Berikut penjelasan dari masing-masing sub-fungsionalitas dari *use case* mengelola data jadwal pelajaran.

A. Melihat Data Jadwal Pelajaran Berdasarkan Kelas (SIAKAD-F-09-A)

Sub-fungsional melihat data jadwal pelajaran berdasarkan kelas memungkinkan aktor untuk melihat data mata pelajaran yang disimpan dalam *database* sistem informasi sesuai dengan kelas yang dipilih. Skenario dapat dilihat pada Tabel 4.35.

Tabel 4.35 Use Case Scenario Melihat Data Jadwal Pelajaran Berdasarkan Kelas

| | |
|----------------------|---|
| Kode Use Case | SIAKAD-F-09-A |
| Deskripsi | Pada <i>use case</i> ini sistem akan menampilkan informasi jadwal pelajaran yang tersimpan di <i>database</i> pada aktor. |
| Aktor | Admin Sistem. |

| | |
|----------------------------|--|
| Pre-condition | Aktor telah mengakses halaman dashboard dan login sebagai admin sistem. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik menu jadwal pelajaran yang ada pada sidebar sistem informasi. 2) Sistem mengarahkan user ke halaman jadwal pelajaran. 3) Aktor mengklik pilihan kelas yang ada pada tabel daftar jadwal pelajaran. 4) Aktor mengklik tombol “detail” 5) Sistem menampilkan daftar data jadwal pelajaran berdasarkan kelas dipilih. |
| Skenario alternatif | Tidak ada |
| Kondisi akhir | Aktor dapat melihat informasi jadwal pelajaran yang tersimpan pada database berdasarkan kelas yang dipilih. |

B. Menambah Data Jadwal Pelajaran (SIAKAD-F-09-B)

Sub-fungsional menambah data jadwal pelajaran memungkinkan aktor untuk menambahkan data jadwal pelajaran baru yang akan disimpan dalam database sistem informasi. Skenario dapat dilihat pada Tabel 4. 36.

Tabel 4. 36 Use Case Scenario Menambah Data Jadwal Pelajaran

| | |
|-----------------------|--|
| Kode Use Case | SIAKAD-F-09-B |
| Deskripsi | Pada <i>use case ini</i> sistem akan melakukan <i>query</i> untuk menambahkan data jadwal pelajaran yang tersimpan di database. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman jadwal pelajaran, dan login sebagai admin sistem. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik tombol tambah yang terdapat diatas tabel daftar jadwal pelajaran. 2) Sistem mengarahkan aktor pada halaman tambah jadwal pelajaran. |

| | | |
|----------------------------|----|---|
| | | <p>3) Aktor mengisi data jadwal pelajaran yang akan ditambahkan sesuai dengan field yang disediakan.</p> <p>4) Aktor menekan tombol “simpan”.</p> <p>5) Sistem melakukan penambahan data ke dalam database lalu menampilkan pesan sukses.</p> <p>6) Sistem mengarahkan aktor ke halaman jadwal pelajaran.</p> |
| Skenario alternatif | a) | <p>Form Tidak Lengkap</p> <ul style="list-style-type: none"> Sistem menampilkan keterangan bahwa field perlu diisi. <p>b) Data yang dimasukkan tidak valid</p> <ul style="list-style-type: none"> Sistem akan menampilkan keterangan bahwa data yang dimasukkan tidak valid. |
| Kondisi akhir | | Aktor dapat melihat informasi jadwal pelajaran telah bertambah sesuai dengan data yang dimasukkan. |

C. Mengubah Data Jadwal Pelajaran (SIAKAD-F-09-C)

Sub-fungsional mengubah data jadwal pelajaran memungkinkan aktor untuk melakukan perubahan terhadap jadwal mata pelajaran yang sudah ada. Skenario dapat dilihat pada Tabel 4. 37.

Tabel 4. 37 Use Case Scenario Mengubah Data Jadwal Pelajaran

| | |
|-----------------------|--|
| Kode Use Case | SIAKAD-F-09-C |
| Deskripsi | Pada <i>use case ini</i> sistem akan melakukan <i>query</i> untuk melakukan perubahan terhadap data jadwal pelajaran yang tersimpan di database sesuai dengan <i>id</i> yang dipilih oleh aktor. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman jadwal pelajaran dan login sebagai admin sistem. |
| Skenario Utama | <p>1) Aktor mengklik ikon bergambar “pensil” yang ada pada baris data jadwal pelajaran yang akan diubah datanya.</p> |

| | | |
|--|----------------------------|--|
| | | <p>2) Sistem mengarahkan aktor pada halaman ubah mata pelajaran.</p> <p>3) Aktor mengisi data jadwal pelajaran yang akan diubah sesuai dengan field yang disediakan.</p> <p>4) Aktor menekan tombol “ubah”.</p> <p>5) Sistem melakukan pengubahan data ke dalam database lalu menampilkan pesan sukses.</p> <p>6) Sistem mengarahkan aktor ke halaman kelas.</p> |
| | Skenario alternatif | <p>a) Data yang dimasukkan tidak valid</p> <p>Sistem akan menampilkan keterangan bahwa data yang dimasukkan tidak valid.</p> <p>b) Aktor menekan tombol batal</p> <p>Sistem mengarahkan ke halaman jadwal pelajaran awal.</p> |
| | Kondisi akhir | Aktor dapat melihat informasi jadwal pelajaran telah berubah sesuai dengan data yang dimasukkan. |

D. Menghapus Data Jadwal Pelajaran (SIAKAD-F-09-D)

Sub-fungsional menghapus data jadwal pelajaran memungkinkan aktor untuk melakukan penghapusan terhadap data jadwal pelajaran yang sudah ada. Skenario dapat dilihat pada Tabel 4. 38.

Tabel 4. 38 Use Case Scenario Menghapus Data Jadwal Pelajaran

| | |
|-----------------------|--|
| Kode Use Case | SIAKAD-F-09-D |
| Deskripsi | Pada <i>use case ini</i> sistem akan melakukan <i>query</i> untuk melakukan penghapusan terhadap data jadwal pelajaran yang tersimpan di database sesuai dengan <i>id</i> yang dipilih oleh aktor. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman jadwal pelajaran dan login sebagai admin sistem. |
| Skenario Utama | <p>1) Aktor mengklik ikon bergambar “tong sampah” yang ada pada baris data mata pelajaran yang akan diubah datanya.</p> |

| | |
|----------------------------|--|
| | <p>2) Sistem mengarahkan aktor pada halaman konfirmasi penghapusan.</p> <p>3) Aktor mengklik tombol “ya”.</p> <p>4) Sistem melakukan penghapusan sesuai dengan id di database lalu menampilkan pesan sukses.</p> <p>5) Sistem mengarahkan aktor ke halaman jadwal pelajaran.</p> |
| Skenario alternatif | <p>a) Aktor menekan tombol batal</p> <ul style="list-style-type: none"> • Sistem mengarahkan ke halaman jadwal pelajaran awal. |
| Kondisi akhir | Aktor dapat melihat informasi mata pelajaran telah berubah sesuai dengan tindakan penghapusan. |

10. Mengelola Kalender Akademik (SIAKAD-F-10)

Pada *use case* ini memungkinkan aktor untuk melakukan kegiatan pengelolaan yang meliputi tambah, ubah, hapus dan lihat kalender akademik dalam sistem. Fungsionalitas dispesifikasikan lebih lanjut dalam sub-fungsional, hal tersebut dilakukan mengingat masing-masing sub-fungsional tersebut memiliki skenario yang berbeda-beda. Berikut penjelasan dari masing-masing sub-fungsionalitas dari *use case* mengelola kalender akademik.

A. Membuat Kegiatan Akademik (SIAKAD-F-10-A)

Sub-fungsional membuat kalender akademik memungkinkan aktor untuk membuat kalender akademik yang akan disimpan dalam database sistem informasi. Skenario dapat dilihat pada Tabel 4. 39.

Tabel 4. 39 Use Case Scenario Membuat Kegiatan Akademik

| | |
|-----------------------|---|
| Kode Use Case | SIAKAD-F-10-A |
| Deskripsi | Pada <i>use case ini</i> sistem akan melakukan <i>query</i> untuk membuat kalender akademik dan disimpan di database. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman dashboard, dan login sebagai admin sistem. |
| Skenario Utama | 1) Aktor mengklik tombol buat yang terdapat diatas tabel daftar kegiatan akademik. |

| | | |
|--|----------------------------|---|
| | | <p>2) Sistem mengarahkan aktor pada halaman buat kegiatan akademik.</p> <p>3) Aktor mengisi data kegiatan akademik yang akan dibuat sesuai dengan <i>field</i> yang disediakan.</p> <p>4) Aktor menekan tombol “tambah”.</p> <p>5) Sistem melakukan penambahan data ke dalam database lalu menampilkan pesan sukses.</p> <p>6) Sistem mengarahkan aktor ke halaman kalender akademik.</p> |
| | Skenario alternatif | <p>a) Form Tidak Lengkap</p> <ul style="list-style-type: none"> • Sistem menampilkan keterangan bahwa <i>field</i> perlu diisi. <p>b) Data yang dimasukkan tidak valid</p> <ul style="list-style-type: none"> • Sistem akan menampilkan keterangan bahwa data yang dimasukkan tidak valid. |
| | Kondisi akhir | Aktor dapat melihat informasi kegiatan akademik telah dibuat dan menunggu verifikasi dari waka kurikulum. |

B. Mengubah Kegiatan Akademik (SIAKAD-F-10-B)

Sub-fungsional mengubah data kalender akademik memungkinkan aktor untuk melakukan perubahan terhadap kalender akademik yang sudah ada. Skenario dapat dilihat pada Tabel 4. 40.

Tabel 4. 40 Use Case Scenario Mengubah Kalender Akademik

| | |
|----------------------|--|
| Kode Use Case | SIAKAD-F-10-B |
| Deskripsi | Pada <i>use case ini</i> sistem akan melakukan <i>query</i> untuk melakukan perubahan terhadap kalender akademik yang tersimpan di database sesuai dengan <i>id</i> yang dipilih oleh aktor. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman kalender akademik dan login sebagai admin sistem. |

| | |
|----------------------------|---|
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik ikon bergambar "pensil" yang ada pada baris kalender akademik yang akan diubah datanya. 2) Sistem mengarahkan aktor pada halaman ubah kalender akademik. 3) Aktor mengisi kalender akademik yang akan diubah sesuai dengan field yang disediakan. 4) Aktor menekan tombol "ubah". 5) Sistem melakukan pengubahan data ke dalam database lalu menampilkan pesan sukses. 6) Sistem mengarahkan aktor ke halaman kalender akademik. |
| Skenario alternatif | <ol style="list-style-type: none"> a) Data yang dimasukkan tidak valid Sistem akan menampilkan keterangan bahwa data yang dimasukkan tidak valid. b) Aktor menekan tombol batal Sistem mengarahkan ke halaman kalender akademik awal. |
| Kondisi akhir | Aktor dapat melihat kalender akademik telah berubah sesuai dengan data yang dimasukkan. |

C. Menghapus Data Kalender Akademik (SIAKAD-F-10-C)

Sub-fungsional menghapus data jadwal pelajaran memungkinkan aktor untuk melakukan penghapusan terhadap kalender akademik yang sudah ada. Skenario dapat dilihat pada Tabel 4. 41.

Tabel 4. 41 Use Case Scenario Menghapus Kalender Akademik

| | |
|----------------------|---|
| Kode Use Case | SIAKAD-F-10-C |
| Deskripsi | Pada <i>use case ini</i> sistem akan melakukan <i>query</i> untuk melakukan penghapusan terhadap data kalender akademik yang tersimpan di database sesuai dengan <i>id</i> yang dipilih oleh aktor. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman kalender akademik dan login sebagai admin sistem. |

| | |
|----------------------------|--|
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik ikon bergambar “tong sampah” yang ada pada baris data kalender akademik yang akan diubah datanya. 2) Sistem mengarahkan aktor pada halaman konfirmasi penghapusan. 3) Aktor mengklik tombol “ya” 4) Sistem melakukan penghapusan sesuai dengan id di database lalu menampilkan pesan sukses. 5) Sistem mengarahkan aktor ke halaman kalender akademik. |
| Skenario alternatif | <ol style="list-style-type: none"> a) Aktor menekan tombol batal <ul style="list-style-type: none"> • Sistem mengarahkan ke halaman kalender akademik awal. |
| Kondisi akhir | Aktor dapat melihat kalender akademik telah berubah sesuai dengan tindakan penghapusan. |

11. Memvalidasi Pelaksanaan Prakerin Siswa (SIAKAD-F-11)

Proses ini merupakan tindak lanjut dari proses pengajuan penyelesaian kegiatan praktik kerja industri yang diajukan oleh siswa. Pada *use case* ini aktor yang bertindak adalah admin BKK. Skenario dari *use case* ini dapat dilihat pada Tabel 4. 42.

Tabel 4. 42 Use Case Scenario Memvalidasi Pelaksanaan Prakerin Siswa

| | |
|-----------------------|--|
| Kode Use Case | SIAKAD-F-11 |
| Deskripsi | Pada <i>use case</i> ini aktor akan menindaklanjuti proses perubahan data prakerin siswa. |
| Aktor | Admin BKK |
| Pre-condition | Aktor telah masuk ke halam dashboard, dan login sebagai admin BKK. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik menu prakerin yang ada pada sidebar sistem informasi. 2) Sistem mengarahkan aktor pada halaman yang dituju. 3) Aktor mengklik data pengajuan prakerin. 4) Sistem mengarahkan aktor pada detail pengajuan. |

| | | |
|----------------------------|----|--|
| | | <p>5) Aktor melakukan pengecekan data pengajuan dengan data lapangan.</p> <p>6) Aktor mengklik tombol “acc”.</p> <p>7) Sistem mengubah status pelaksanaan prakerin yang diajukan siswa menjadi “sudah prakerin”.</p> |
| Skenario alternatif | a) | <p>Data pengajuan tidak cocok dengan data lapangan.</p> <ul style="list-style-type: none"> • Aktor menekan tombol “tidak diterima”. • Status pelaksanaan prakerin tetap tidak diubah. |
| Kondisi akhir | | Sistem mengarahkan aktor pada halaman prakerin. |

12. Mengelola Data Mitra Industri (SIAKAD-F-12)

Pada *use case* ini memungkinkan aktor untuk melakukan kegiatan pengelolaan yang meliputi tambah, ubah, hapus dan lihat data mitra industry yang bekerja sama dengan SMK Muhammadiyah 1 Kemlagi. Fungsionalitas dispesifikasi lebih lanjut dalam sub-fungsional, hal tersebut dilakukan mengingat masing-masing sub-fungsional tersebut memiliki skenario yang berbeda-beda. Berikut penjelasan dari masing-masing sub-fungsionalitas dari *use case* mengelola data mitra industri.

A. Melihat Data Mitra Industri (SIAKAD-F-12-A)

Sub-fungsional melihat mitra industri memungkinkan aktor untuk melihat data mitra industri yang tersimpan dalam *database* sistem informasi. Skenario dapat dilihat pada Tabel 4. 43.

Tabel 4. 43 Use Case Scenario Melihat Data Mitra Industri

| | |
|-----------------------|--|
| Kode Use Case | SIAKAD-F-12-A |
| Deskripsi | Pada <i>use case</i> ini sistem akan menampilkan informasi mitra industri yang tersimpan di database pada aktor. |
| Aktor | Admin BKK |
| Pre-condition | Aktor telah mengakses halaman dashboard dan login sebagai admin BKK. |
| Skenario Utama | <p>1) Aktor mengklik menu mitra industri yang ada pada sidebar sistem informasi.</p> |

| | | | | |
|--|----------------------------|-----------|---|--|
| | | | 2) Sistem mengarahkan user ke halaman mitra industri. | |
| | | | 3) Sistem menampilkan daftar data mitra industri. | |
| | Skenario alternatif | Tidak ada | | |

B. Menambah Data Mitra Industri (SIAKAD-F-12-B)

Sub-fungsional menambah data mitra industri memungkinkan aktor untuk menambahkan data mitra industri baru yang akan disimpan dalam database sistem informasi. Skenario dapat dilihat pada Tabel 4. 44.

Tabel 4. 44 Use Case Scenario Menambah Data Mitra Industri

| | |
|----------------------------|--|
| Kode Use Case | SIAKAD-F-12-B |
| Deskripsi | Pada <i>use case ini</i> sistem akan melakukan <i>query</i> untuk menambahkan data mitra industri yang akan disimpan di database. |
| Aktor | Admin BKK |
| Pre-condition | Aktor telah mengakses halaman mitra industri, dan login sebagai admin BKK. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik tombol tambah yang terdapat diatas tabel daftar mitra industri. 2) Sistem mengarahkan aktor pada halaman tambah mitra industri. 3) Aktor mengisi data mitra industri yang akan ditambahkan sesuai dengan field yang disediakan. 4) Aktor menekan tombol "simpan". 5) Sistem melakukan penambahan data ke dalam database lalu menampilkan pesan sukses. 6) Sistem mengarahkan aktor ke halaman mitra industri. |
| Skenario alternatif | a) Form Tidak Lengkap |

| | | |
|--|----------------------|---|
| | | <ul style="list-style-type: none"> Sistem menampilkan keterangan bahwa <i>field</i> perlu diisi. |
| | b) | <ul style="list-style-type: none"> Data yang dimasukkan tidak valid |
| | Kondisi akhir | Aktor dapat melihat data mitra industri telah bertambah sesuai dengan data yang dimasukkan. |

C. Mengubah Data Mitra Industri (SIAKAD-F-12-C)

Sub-fungsional mengubah data mitra industro memungkinkan aktor untuk melakukan perubahan terhadap data mitra industri yang sudah ada. Skenario dapat dilihat pada Tabel 4. 45.

Tabel 4. 45 Use Case Scenario Mengubah Data Mitra Industri

| | |
|-----------------------|--|
| Kode Use Case | SIAKAD-F-12-C |
| Deskripsi | Pada <i>use case</i> ini sistem akan melakukan <i>query</i> untuk melakukan perubahan terhadap data mitra industri yang tersimpan di database sesuai dengan <i>id</i> yang dipilih oleh aktor. |
| Aktor | Admin BKK |
| Pre-condition | Aktor telah mengakses halaman mitra industri dan login sebagai admin BKK. |
| Skenario Utama | <ol style="list-style-type: none"> Aktor mengklik ikon bergambar “pensil” yang ada pada baris data mitra industri yang akan diubah datanya. Sistem mengarahkan aktor pada halaman ubah mitra industri. Aktor mengisi data mitra industri yang akan diubah sesuai dengan <i>field</i> yang disediakan. Aktor menekan tombol “ubah”. Sistem melakukan pengubahan data ke dalam database lalu menampilkan pesan sukses. Sistem mengarahkan aktor ke mitra industri. |

| | |
|----------------------------|---|
| Skenario alternatif | <p>a) Data yang dimasukkan tidak valid Sistem akan menampilkan keterangan bahwa data yang dimasukkan tidak valid.</p> <p>b) Aktor menekan tombol batal Sistem mengarahkan ke halaman mitra industri awal.</p> |
| Kondisi akhir | Aktor dapat melihat data mitra industri telah berubah sesuai dengan data yang dimasukkan. |

D. Menghapus Data Mitra Industri (SIAKAD-F-12-D)

Sub-fungsional menghapus data mitra industri memungkinkan aktor untuk melakukan penghapusan terhadap data mitra industri yang sudah ada. Skenario dapat dilihat pada Tabel 4. 46.

Tabel 4. 46 Use Case Scenario Menghapus Data Mitra Industri

| | |
|----------------------------|--|
| Kode Use Case | SIAKAD-F-12-D |
| Deskripsi | Pada <i>use case ini</i> sistem akan melakukan <i>query</i> untuk melakukan penghapusan terhadap data mitra industri yang tersimpan di database sesuai dengan <i>id</i> yang dipilih oleh aktor. |
| Aktor | Admin Sistem |
| Pre-condition | Aktor telah mengakses halaman jadwal pelajaran dan login sebagai admin BKK. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik ikon bergambar “tong sampah” yang ada pada baris data mata pelajaran yang akan diubah datanya. 2) Sistem mengarahkan aktor pada halaman konfirmasi penghapusan. 3) Aktor mengklik tombol “ya”. 4) Sistem melakukan penghapusan sesuai dengan <i>id</i> di database lalu menampilkan pesan sukses. 5) Sistem mengarahkan aktor ke halaman mitra industri. |
| Skenario alternatif | <p>a) Aktor menekan tombol batal</p> <ul style="list-style-type: none"> • Sistem mengarahkan ke halaman mitra industri awal. |

| | |
|----------------------|--|
| Kondisi akhir | Aktor dapat melihat informasi mata pelajaran telah berubah sesuai dengan tindakan penghapusan. |
|----------------------|--|

13. Melihat Data Pelaksanaan Prakerin (SIAKAD-F-13)

Proses ini memungkinkan aktor untuk melihat data-data pelaksanaan prakerin oleh siswa, data-data yang dimaksud adalah tempat prakerin, jenis industri, dan narahubung. Pada *use case* ini aktor yang bertindak adalah admin BKK. Skenario dari *use case* ini dapat dilihat pada Tabel 4. 47.

Tabel 4. 47 Use Case Scenario Melihat Data Pelaksanaan Prakerin

| | |
|----------------------------|--|
| Kode Use Case | SIAKAD-F-13 |
| Deskripsi | Pada <i>use case</i> ini aktor dapat melihat data-data pelaksanaan prakerin. |
| Aktor | Admin Sistem, Admin BKK, Waka Kurikulum. |
| Pre-condition | Aktor telah masuk ke halaman dashboard. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik menu laporan prakerin yang ada pada sidebar sistem informasi. 2) Sistem mengarahkan aktor pada halaman yang dituju. 3) Sistem menampilkan data pelaksanaan prakerin siswa. |
| Skenario alternatif | Tidak ada. |
| Kondisi akhir | Aktor dapat melihat data pelaksanaan prakerin. |

14. Memvalidasi Kalender Akademik (SIAKAD-F-14)

Proses ini merupakan tindak lanjut dari *use case* pembuatan kalender akademik yang dilakukan oleh admin sistem. Skenario dari *use case* ini dapat dilihat pada Tabel 4. 48.

Tabel 4. 48 Use Case Scenario Memvalidasi Kalender Akademik

| | |
|----------------------|--|
| Kode Use Case | SIAKAD-F-14 |
| Deskripsi | Pada <i>use case</i> ini aktor akan menindaklanjuti proses penerbitan kalender akademik. |
| Aktor | Waka Kurikulum |
| Pre-condition | Aktor telah masuk ke halaman dashboard. |

| | |
|----------------------------|---|
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik menu validasi yang ada pada sidebar sistem informasi. 2) Aktor mengklik sub-menu kalender akademik. 3) Sistem menampilkan halaman yang dituju. 4) Aktor melakukan pengecekan data kalender yang akan diterbitkan. 5) Aktor mengklik tombol “acc”. 6) Sistem menerbitkan kalender akademik. |
| Skenario alternatif | <p>a) Data kalender ditemukan kesalahan.</p> <ul style="list-style-type: none"> • Aktor menekan tombol “belum acc”. • Status kalender akademik “perlu direvisi”. |
| Kondisi akhir | Sistem mengarahkan aktor pada halaman validasi kalender akademik. |

15. Memvalidasi Penetapan Wali Kelas (SIAKAD-F-15)

Proses ini merupakan tindak lanjut dari *use case* penentuan wali kelas yang dilakukan oleh admin sistem. Skenario dari *use case* ini dapat dilihat pada Tabel 4. 49.

Tabel 4. 49 Use Case Scenario Memvalidasi Penetapan Wali Kelas

| | |
|-----------------------|--|
| Kode Use Case | SIAKAD-F-15 |
| Deskripsi | Pada <i>use case</i> ini aktor akan menindaklanjuti proses penetapan wali kelas. |
| Aktor | Waka Kurikulum |
| Pre-condition | Aktor telah masuk ke halaman dashboard. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik menu validasi yang ada pada sidebar sistem informasi. 2) Aktor mengklik sub-menu wali kelas. 3) Sistem menampilkan halaman yang dituju. 4) Aktor melakukan pengecekan kebenaran data. 5) Aktor mengklik tombol “acc”. 6) Sistem menetapkan wali kelas pada kelas yang dituju. |

| | |
|----------------------------|---|
| Skenario alternatif | a) Data salah <ul style="list-style-type: none"> • Aktor menekan tombol “tidak diterima”. • Status kelas “menjadi perlu perbaikan”. |
| Kondisi akhir | Sistem mengarahkan aktor pada halaman validasi wali kelas. |

16. Memperbarui Data Diri (SIAKAD-F-16)

Use case ini memungkinkan aktor untuk melakukan pembaruan data pribadi secara berkala apabila dibutuhkan. Skenario dari use case ini dapat dilihat pada Tabel 4. 50.

Tabel 4. 50 Use Case Scenario Memperbarui Data Diri

| | |
|----------------------------|---|
| Kode Use Case | SIAKAD-F-16 |
| Deskripsi | Pada use case ini aktor dapat memperbarui data diri. |
| Aktor | Guru, Siswa. |
| Pre-condition | Aktor telah masuk ke halaman dashboard. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik menu ubah profile yang ada pada sidebar sistem informasi. 2) Sistem mengarahkan aktor pada halaman yang dituju. 3) Aktor memperbarui data sesuai dengan field yang disediakan. 4) Aktor menekan tombol “submit”. 5) Sistem menyimpan perubahan data. 6) Sistem mengarahkan aktor pada halaman profile. |
| Skenario alternatif | A. Data yang dimasukkan tidak valid. <ul style="list-style-type: none"> • Sistem menampilkan keterangan bahwa data yang dimasukkan tidak valid. |
| Kondisi akhir | Aktor dapat melihat data diri yang telah diubah pada halaman profile. |

17. Melihat Kelas yang Diwakili (SIAKAD-F-17)

Use case ini memungkinkan aktor untuk melihat informasi kelas yang diajar. Skenario dari use case ini dapat dilihat pada Tabel 4. 51.

Tabel 4. 51 Use Case Scenario Melihat Kelas Yang Diwakili

| | |
|----------------------------|---|
| Kode Use Case | SIAKAD-F-17 |
| Deskripsi | Pada <i>use case ini</i> aktor dapat melihat kelas yang saat ini diwakili. |
| Aktor | Guru. |
| Pre-condition | Aktor telah masuk ke halaman dashboard. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik menu kelas yang diajar yang ada pada sidebar sistem informasi. 2) Sistem menampilkan daftar kelas. |
| Skenario alternatif | Tidak ada |
| Kondisi akhir | Aktor dapat melihat daftar kelas yang diajar pada halaman kelas yang diajar. |

18. Memasukan Nilai Hasil Evaluasi Siswa (SIAKAD-F-18)

*Use case ini memungkinkan aktor untuk melakukan *input* nilai hasil evaluasi tengah semester dan evaluasi akhir semester siswa. Skenario dari use case ini dapat dilihat pada Tabel 4. 52.*

Tabel 4. 52 Use Case Scenario Memasukkan Nilai Hasil Evaluasi Siswa

| | |
|-----------------------|--|
| Kode Use Case | SIAKAD-F-18 |
| Deskripsi | Pada <i>use case ini</i> aktor dapat memasukkan nilai hasil evaluasi tengah semester dan akhir semester siswa. |
| Aktor | Guru. |
| Pre-condition | Aktor telah masuk ke halaman kelas yang diajar. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik salah satu kelas yang ada pada daftar kelas yang diajar. 2) Sistem menampilkan detail informasi kelas yang diajar. 3) Aktor memilih siswa yang akan dimasukkan nilainya. 4) Aktor memasukkan nilai sesuai dengan field yang disediakan. |

| | | |
|----------------------------|---|---|
| | | 5) Aktor mengklik tombol “submit” 6) Sistem menyimpan nilai yang dimasukkan. |
| Skenario alternatif | A. Data yang dimasukkan tidak valid. • Sistem menampilkan keterangan bahwa data yang dimasukkan tidak valid. | |
| Kondisi akhir | Nilai yang dimasukkan akan menunggu validasi dari wali kelas. | |

19. Mengubah Nilai Hasil Evaluasi Siswa(SIAKAD-F-20)

Use case ini dilakukan apabila data hasil evaluasi tidak di “acc” oleh guru yang menjadi wali kelas. Wali kelas akan menginformasikan pada guru melalui proses diluar sistem, untuk melakukan kompromi nilai. Skenario dari use case ini dapat dilihat pada Tabel 4. 53.

Tabel 4. 53 Use Case Scenario Mengubah Nilai Hasil Evaluasi Siswa

| | |
|----------------------------|--|
| Kode Use Case | SIAKAD-F-20 |
| Deskripsi | Pada use case ini aktor dapat mengubah nilai hasil evaluasi tengah semester dan akhir semester siswa. |
| Aktor | Guru. |
| Pre-condition | Aktor telah masuk ke halaman kelas yang diajar. |
| Skenario Utama | 1) Aktor mengklik salah satu kelas yang ada pada daftar kelas yang diajar. 2) Sistem menampilkan detail informasi kelas yang diajar. 3) Aktor memilih siswa yang akan dirubah nilainya. 4) Aktor memasukkan nilai sesuai dengan field yang disediakan. 5) Aktor mengklik tombol “submit” 6) Sistem menyimpan nilai yang dimasukkan. |
| Skenario alternatif | a) Data yang dimasukkan tidak valid. • Sistem menampilkan keterangan bahwa data yang dimasukkan tidak valid. |

| | |
|----------------------|---|
| Kondisi akhir | Nilai yang dimasukkan akan menunggu validasi dari wali kelas. |
|----------------------|---|

20. Melihat Detail Informasi Kelas (SIAKAD-F-21)

Use case ini memungkinkan aktor untuk melihat informasi riwayat kelas. Skenario dari use case ini dapat dilihat pada Tabel 4. 54.

Tabel 4. 54 Use Case Scenario Melihat Informasi Kelas Saat Ini

| | |
|----------------------------|--|
| Kode Use Case | SIAKAD-F-21 |
| Deskripsi | Pada use case ini aktor dapat melihat informasi-informasi yang berhubungan dengan kelas yang didukti, seperti daftar siswa, wali kelas dan jadwal pelajaran. |
| Aktor | Siswa |
| Pre-condition | Aktor telah masuk dashboard. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik menu kelas yang ada pada sidebar sistem informasi. 2) Sistem menampilkan daftar informasi kelas. 3) Aktor mengklik kelas yang ingin dilihat detailnya. |
| Skenario alternatif | Tidak ada |
| Kondisi akhir | Aktor dapat melihat informasi kelas. |

21. Melihat Nilai Hasil Evaluasi (SIAKAD-F-22)

Use case ini memungkinkan aktor untuk melihat nilai hasil evaluasi tengah semester dan evaluasi akhir semester. Skenario dari use case ini dapat dilihat pada Tabel 4. 55.

Tabel 4. 55 Use Case Scenario Melihat Nilai Hasil Evaluasi

| | |
|----------------------|---|
| Kode Use Case | SIAKAD-F-22 |
| Deskripsi | Pada use case ini aktor dapat melihat nilai hasil evaluasi tengah semester dan evaluasi akhir semester yang telah dilalui oleh siswa. |
| Aktor | Siswa |
| Pre-condition | Aktor telah masuk dashboard. |

| | |
|----------------------------|---|
| Skenario Utama | 1) Aktor mengklik menu rapor yang ada pada sidebar sistem informasi. 2) Sistem menampilkan halaman nilai hasil evaluasi siswa. |
| Skenario alternatif | Tidak ada |
| Kondisi akhir | Aktor dapat melihat nilai hasil evaluasi. |

22. Mengajukan Penuntasan Pelaksanaan Prakerin (SIAKAD-F-23)

Use case ini memungkinkan aktor untuk melakukan pengajuan penuntasan pelaksanaan prakerin. Skenario dari *use case* ini dapat dilihat pada Tabel 4. 56.

Tabel 4. 56 Use Case Scenario Mengajukan Penuntasan Pelaksanaan Prakerin

| | |
|----------------------------|---|
| Kode Use Case | SIAKAD-F-23 |
| Deskripsi | Pada <i>use case</i> ini aktor dapat mengajukan penuntasan pelaksanaan prakerin, yang nantinya akan divalidasi oleh admin BKK. |
| Aktor | Siswa |
| Pre-condition | <ul style="list-style-type: none"> - Aktor telah masuk dashboard. - Aktor tidak memiliki <i>record</i> data prakerin yang telah tervalidasi di sistem. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik menu prakerin yang ada pada sidebar sistem informasi. 2) Sistem menampilkan halaman prakerin. 3) Aktor mengisi data informasi prakerin dengan benar. 4) Aktor mengklik tombol “ajukan”. 5) Sistem menyimpan data yang dimasukkan. |
| Skenario alternatif | <ol style="list-style-type: none"> a) Data yang dimasukkan tidak valid. <ul style="list-style-type: none"> • Sistem menampilkan keterangan bahwa data yang dimasukkan tidak valid. |
| Kondisi akhir | Sistem menyimpan data yang dimasukkan dan menunggu validasi oleh admin BKK. |

23. Mengubah Data Penuntasan Pelaksanaan Prakerin (SIAKAD-F-24)

Use case ini dilakukan apabila data yang diajukan sebelumnya dinyatakan ditolak oleh admin BKK. Skenario dari use case ini dapat dilihat pada Tabel 4. 57.

Tabel 4. 57 Use Case Scenario Mengubah Data Penuntasan Pelaksanaan Prakerin

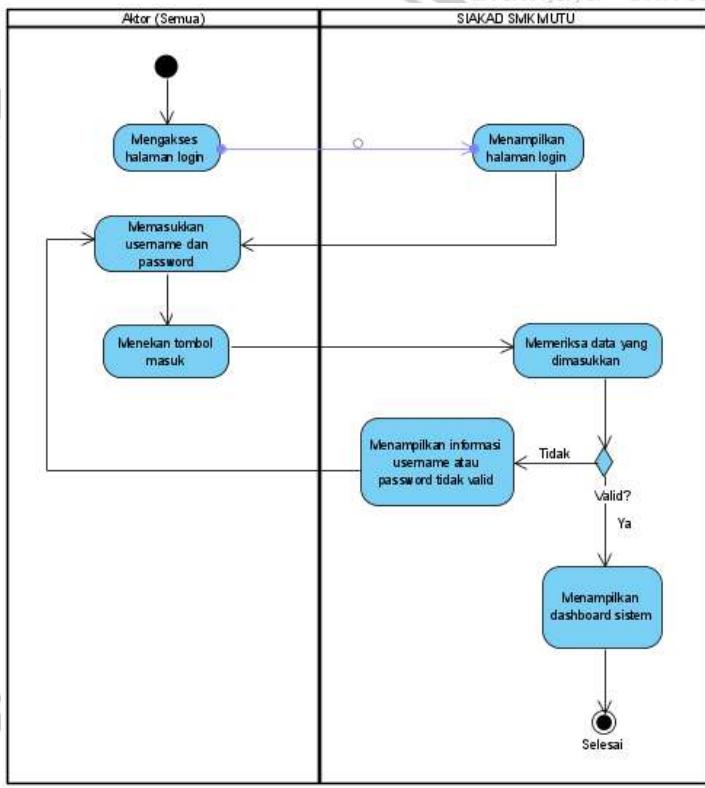
| | |
|----------------------------|---|
| Kode Use Case | SIAKAD-F-24 |
| Deskripsi | Pada <i>use case ini</i> aktor dapat mengubah data pengajuan penuntasan prakerin. |
| Aktor | Siswa |
| Pre-condition | <ul style="list-style-type: none"> - Aktor telah masuk dashboard. - Aktor tidak memiliki <i>record</i> data prakerin yang telah tervalidasi di sistem. |
| Skenario Utama | <ol style="list-style-type: none"> 1) Aktor mengklik menu prakerin yang ada pada <i>sidebar</i> sistem informasi. 2) Sistem menampilkan halaman prakerin. 3) Aktor mengklik ubah pengajuan prakerin. 4) Aktor mengisi data sesuai field yang ada. 5) Mengklik tombol submit. 6) Sistem menyimpan data. 7) Sisem mengarahkan pada halaman prakerin. |
| Skenario alternatif | <p>A. Data yang dimasukkan tidak valid.</p> <p>A. Sistem menampilkan keterangan bahwa data yang dimasukkan tidak valid.</p> |
| Kondisi akhir | Sistem menyimpan data yang dimasukkan dan menunggu validasi oleh admin BKK. |

4.2.3 Activity Diagram

Penggunaan *activity diagram* bertujuan untuk menggambarkan siklus hidup sistem secara representatif. Dengan *activity diagram* tahap *use case scenario* yang telah dibuat akan dibuat representasi grafisnya. Berikut adalah *activity diagram* dari *use case* yang ada.

1. Melakukan Login Ke Sistem

Activity diagram login ke sistem merupakan sebuah aktivitas ketika aktor ingin mengakses fungsi-fungsi yang ada pada sistem, *login* merupakan sebuah langkah awal dan wajib dilakukan oleh aktor sebelum bisa menggunakan sistem. Proses autentifikasi menggunakan username dan password, setelah aktor memasukan username dan password maka, sistem akan melakukan pengecekan apakah *username* dan *password* yang dimasukkan oleh aktor valid (terdaftar pada database). Apabila *username* atau *password* tidak terdaftar maka Sistem akan menampilkan informasi bahwa data yang dimasukkan tidak valid. Apabila *username* dan *password* yang dimasukkan terdaftar maka sistem akan mengalihkan aktor pada halaman dashboard sistem informasi. *Activity diagram* untuk proses login ke sistem dapat dilihat pada

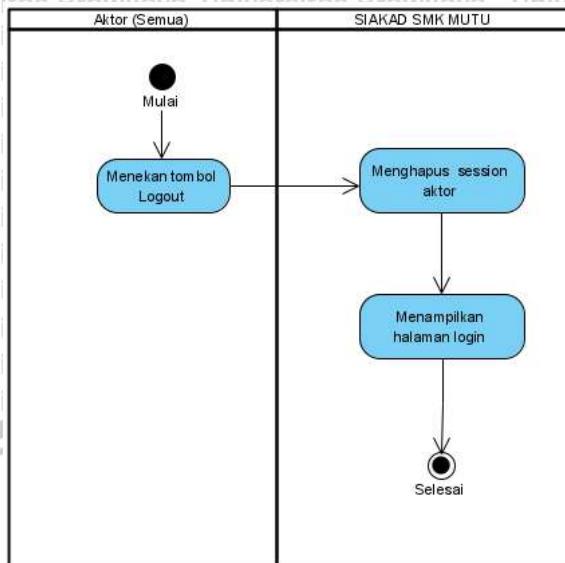


Gambar 4. 15 Activity Diagram Proses Login

2. Melakukan Logout Dari Sistem

Activity diagram *logout* dari sistem berguna untuk melakukan pengamanan terhadap akses yang tidak dikehendaki pada sistem informasi. Untuk melakukan logout aktor hanya perlu menekan tombol logout yang ada pada sidebar sistem informasi. Setelah itu sistem akan menghapus session aktor dan mengarahkan aktor pada halaman *login* awal. Untuk mengakses kembali sistem informasi aktor perlu memasukkan

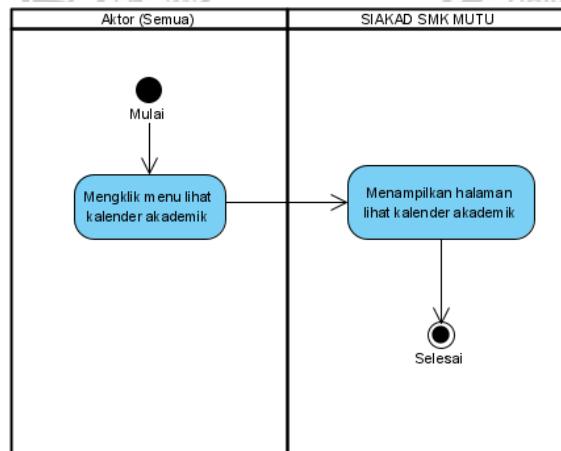
kembali *username* dan *password* dan menekan tombol masuk. *Activity diagram* untuk proses logout dari sistem dapat dilihat pada Gambar 4. 16.



Gambar 4. 16 *Activity Diagram Proses Logout*

3. Melihat kalender Akademik

Activity diagram melihat kalender akademik memungkinkan seluruh aktor dapat melihat informasi kalender akademik dari sumber yang terpusat. Untuk melihat kalender akademik aktor perlu melakukan klik pada menu lihat kalender akademik yang ada pada *sidebar* sistem informasi. Setelah itu sistem akan menampilkan halaman kalender akademik. *Activity diagram* melihat kalender akademik dapat dilihat pada Gambar 4. 17.



Gambar 4.17 Activity Diagram Proses Melihat Kalender Akademik

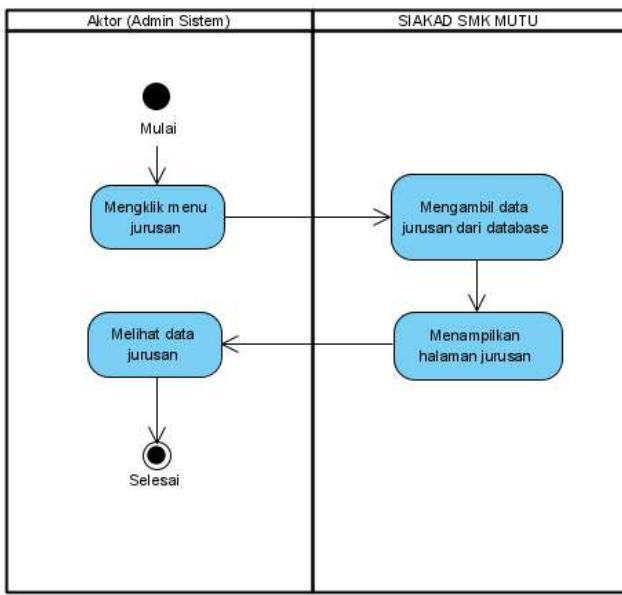
4. Mengelola Data Jurusan

Activity diagram mengelola data jurusan terdiri dari 4 (empat) *sub-activity* yang meliputi penambahan, pengubahan, pembacaan, dan

penghapusan. Aktor pada *activity* ini adalah admin sistem. Masing-masing *activity* dalam pengelolaan data jurusan adalah sebagai berikut:

A. Melihat Data Jurusan

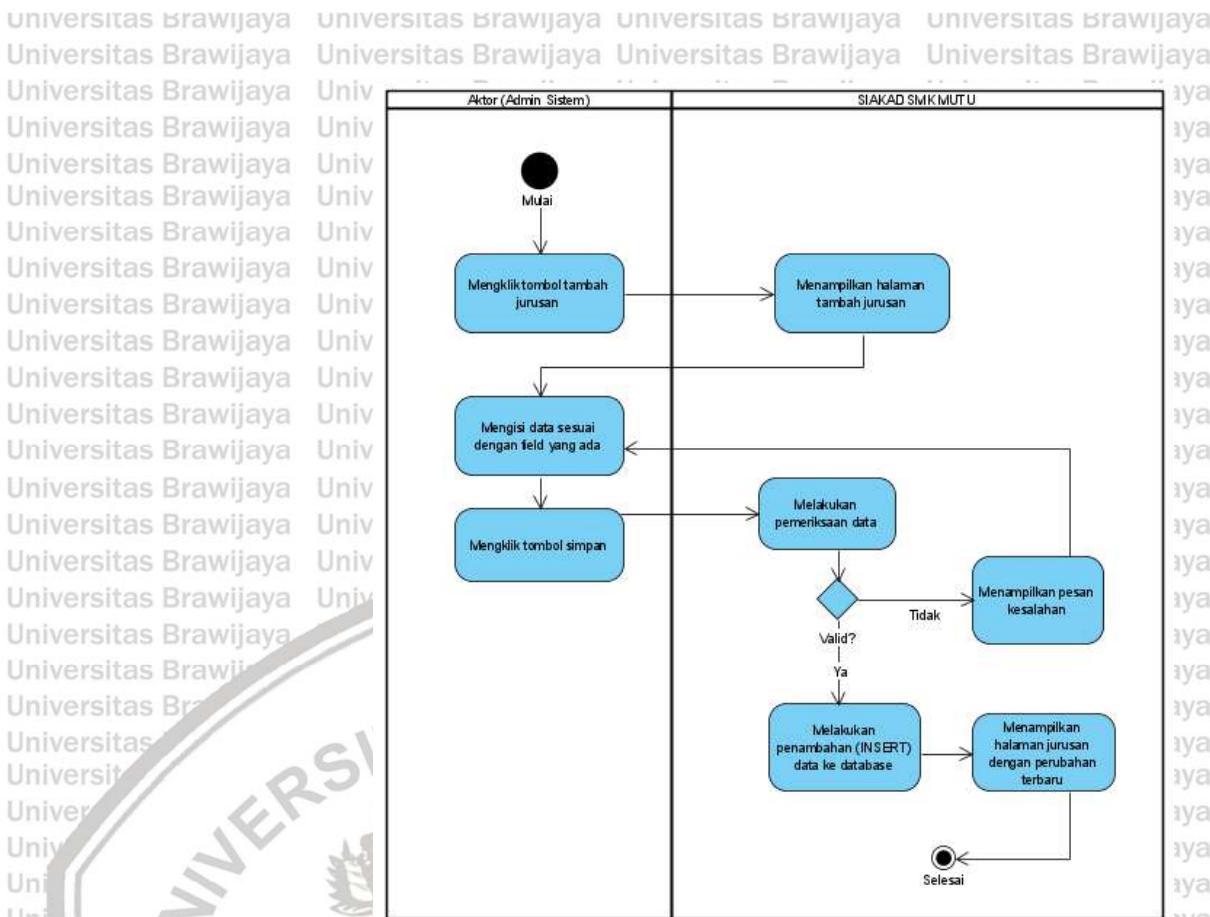
Activity melihat data jurusan memungkinkan aktor untuk dapat melihat data-data mengenai jurusan yang saat ini ada di SMK Muhammadiyah 1 Kemlagi. *Activity diagram* untuk kegiatan melihat data jurusan pada Gambar 4. 18.



Gambar 4. 18 *Activity Diagram* Melihat Data Jurusan

B. Menambah Data Jurusan

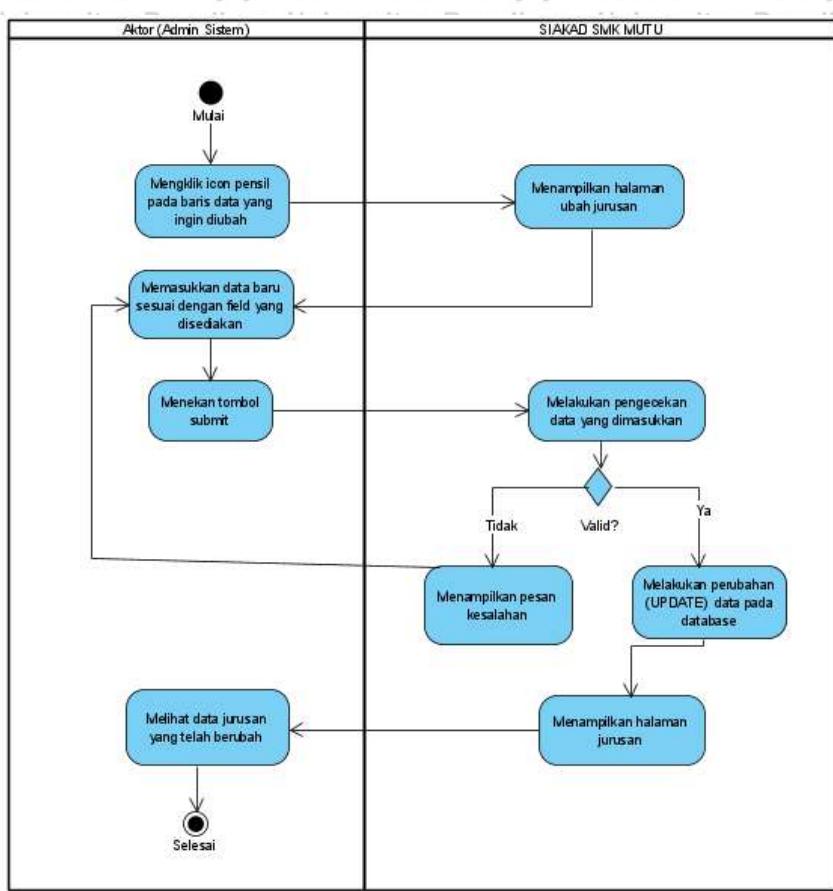
Activity diagram menambah data jurusan merupakan model diagram untuk memasukkan data jurusan baru pada database melalui sistem informasi. Aktor pada *activity* ini adalah admin sistem. *Activity diagram* untuk kegiatan menambah data jurusan pada Gambar 4. 19.



Gambar 4. 19 Activity Diagram Menambah Data Jurusan

C. Mengubah Data Jurusan

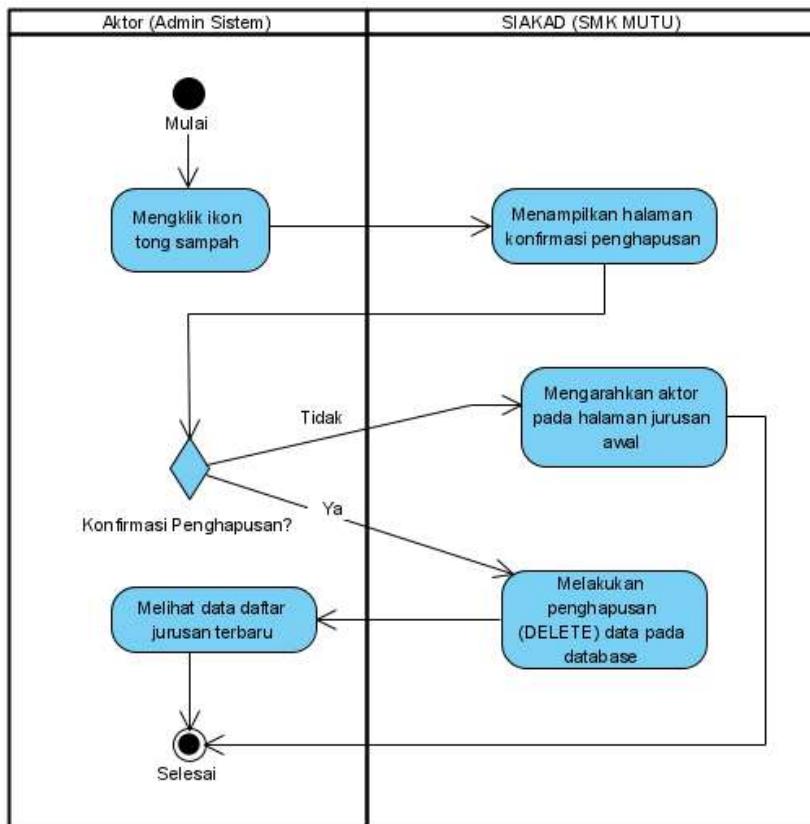
Activity diagram mengubah data jurusan merupakan model diagram untuk melakukan perubahan informasi pada suatu jurusan yang sudah ada pada database melalui sistem informasi. Aktor pada *activity* ini adalah admin sistem. *Activity diagram* untuk kegiatan mengubah data jurusan pada Gambar 4. 20.



Gambar 4. 20 Activity Diagram Mengubah Data Jurusan

D. Menghapus Data Jurusan

Activity diagram menghapus data jurusan merupakan model diagram untuk melakukan penghapusan suatu jurusan yang sudah ada pada *database* melalui sistem informasi. Aktor pada *activity* ini adalah admin sistem. *Activity diagram* untuk kegiatan menghapus data jurusan pada Gambar 4. 21.



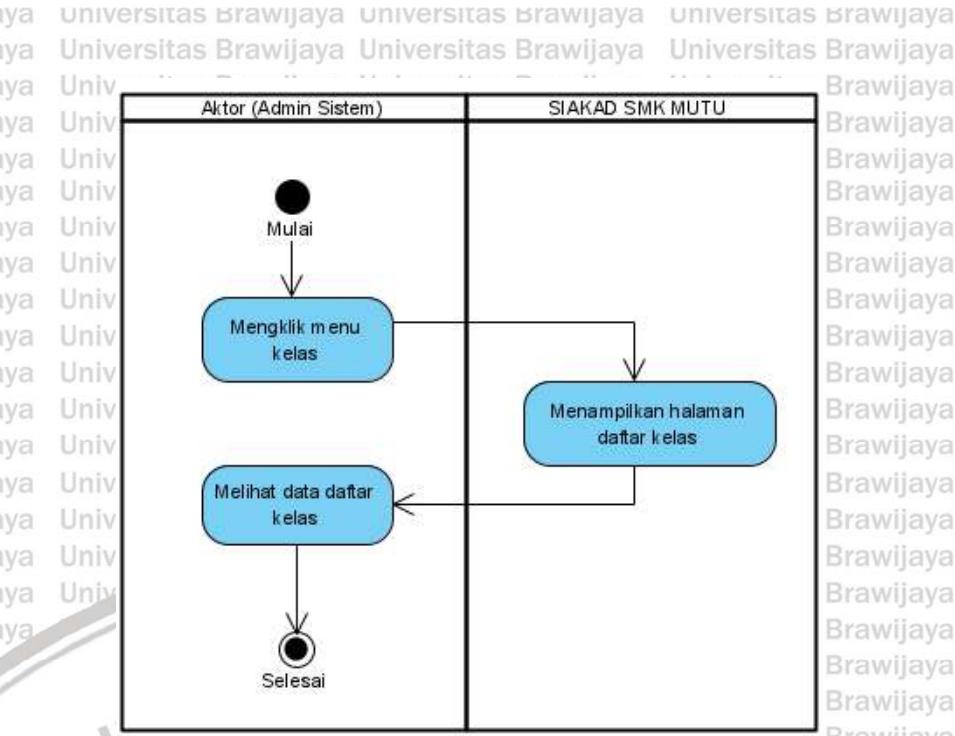
Gambar 4. 21 Activity Diagram Menghapus Data Jurusan

5. Mengelola Data Kelas

Activity diagram mengelola data kelas terdiri dari 5 (lima) *sub-activity* yang meliputi penambahan data kelas, pengubahan informasi kelas, pembacaan data kelas, penghapusan kelas dan penentuan wali kelas. Aktor pada *activity* ini adalah admin sistem. Masing-masing *activity* dalam pengelolaan data kelas adalah sebagai berikut:

A. Melihat Data Kelas

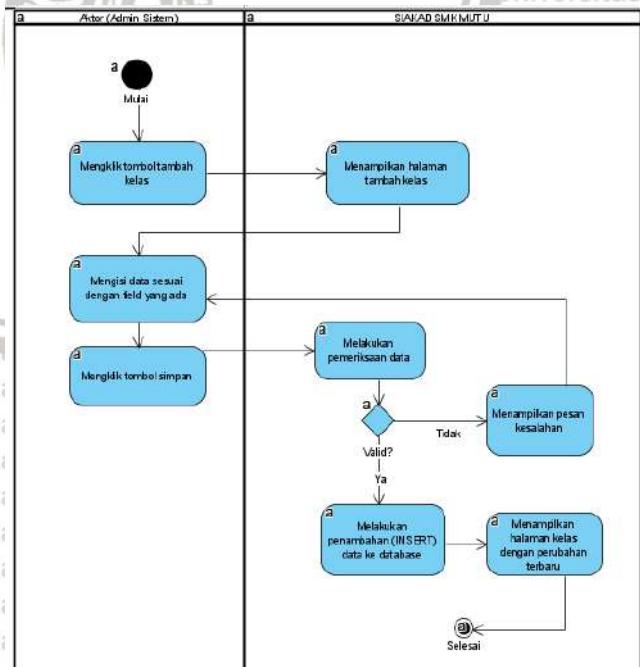
Activity melihat data kelas memungkinkan aktor untuk dapat melihat data-data mengenai kelas yang saat ini ada di SMK Muhammadiyah 1 Kemlagi. *Activity diagram* untuk kegiatan melihat data kelas pada Gambar 4. 22.



Gambar 4. 22 Activity Diagram Melihat Data Kelas

B. Menambah Data Kelas

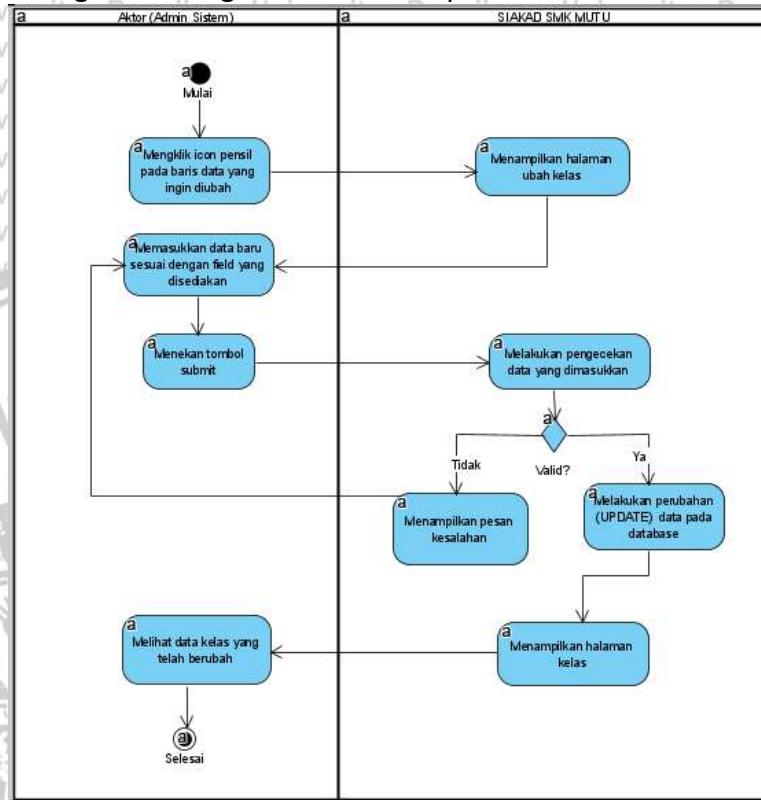
Activity diagram menambah data kelas merupakan model diagram untuk memasukkan data kelas baru pada *database* melalui sistem informasi. Aktor pada *activity* ini adalah admin sistem. *Activity diagram* untuk kegiatan menambah data kelas pada Gambar 4. 23.



Gambar 4. 23 Activity Diagram Menambah Data Kelas

C. Mengubah Data Kelas

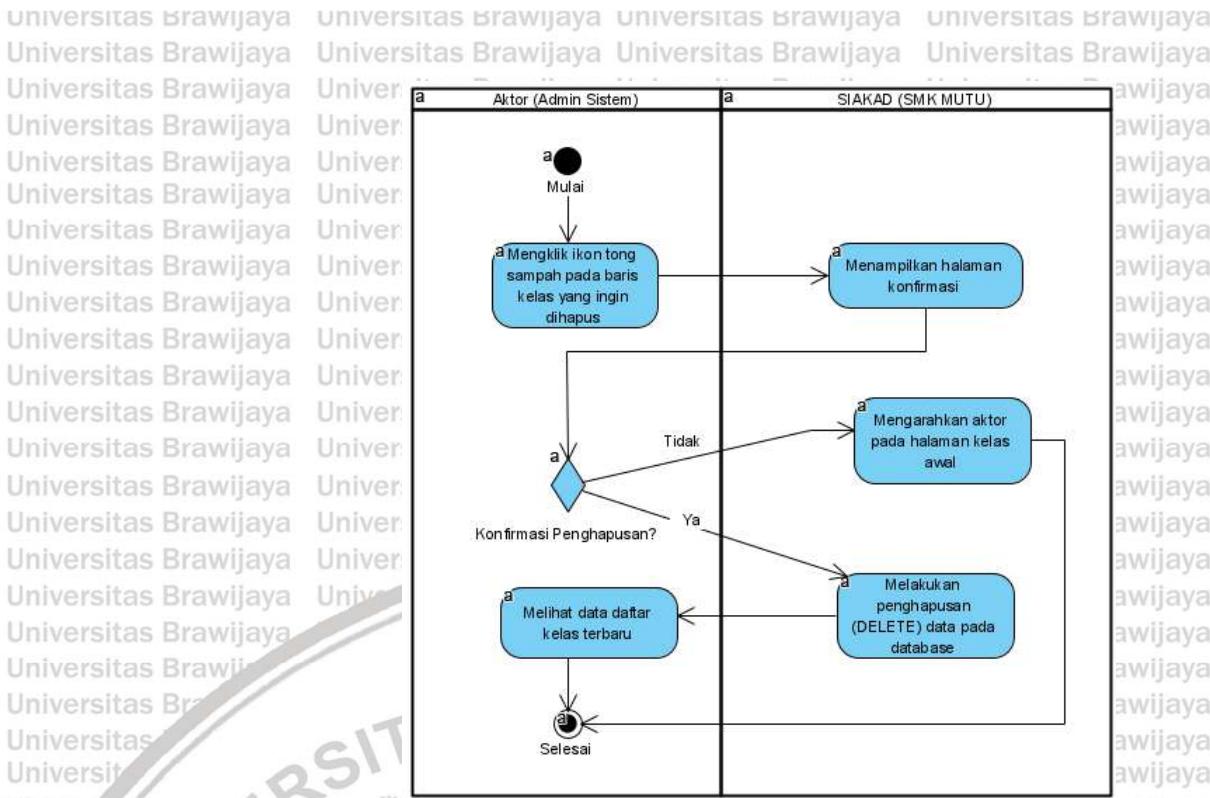
Activity diagram mengubah data kelas merupakan model diagram untuk melakukan perubahan informasi pada suatu kelas yang sudah ada pada database melalui sistem informasi. Aktor pada *activity* ini adalah admin sistem. *Activity diagram* untuk kegiatan mengubah data kelas pada Gambar 4. 24.



Gambar 4. 24 *Activity Diagram* mengubah Data Kelas

D. Menghapus Data Kelas

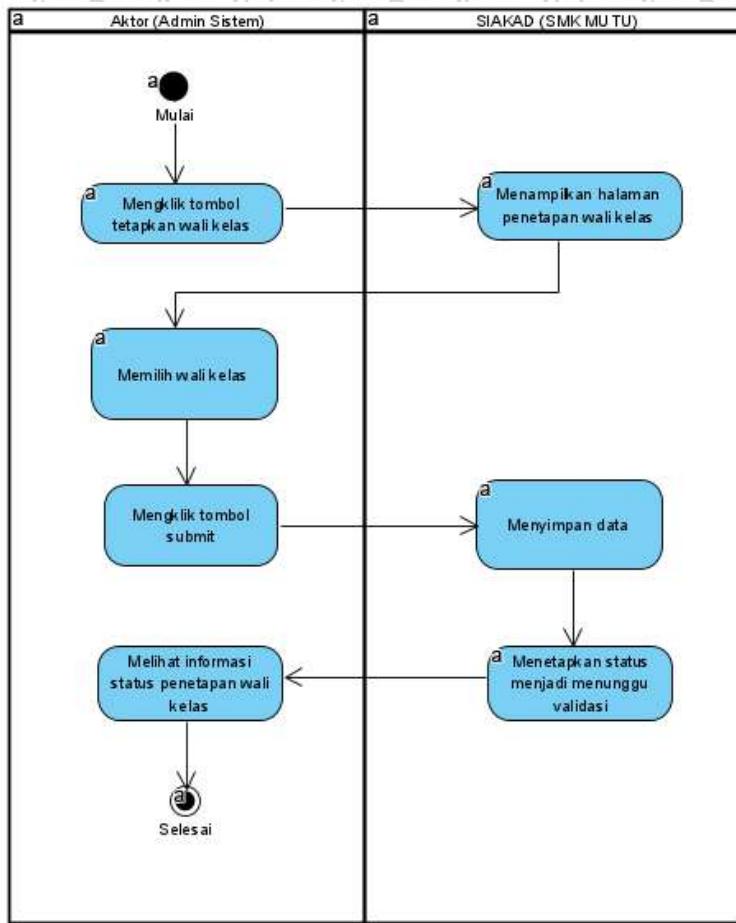
Activity diagram menghapus data kelas merupakan model diagram untuk melakukan penghapusan suatu kelas yang sudah ada pada database melalui sistem informasi. Aktor pada *activity* ini adalah admin sistem. *Activity diagram* untuk kegiatan menghapus data kelas pada Gambar 4. 25.



Gambar 4. 25 Activity Diagram Menghapus Data Kelas

E. Menetapkan Wali Kelas

Activity menetapkan wali kelas memungkinkan aktor untuk mengatur penempatan wali kelas untuk setiap kelas yang ada pada tahun ajaran baru melalui sistem informasi, activity ini dilakukan oleh admin sistem. *Activity diagram* menetapkan wali kelas dapat dilihat pada Gambar 4. 26.



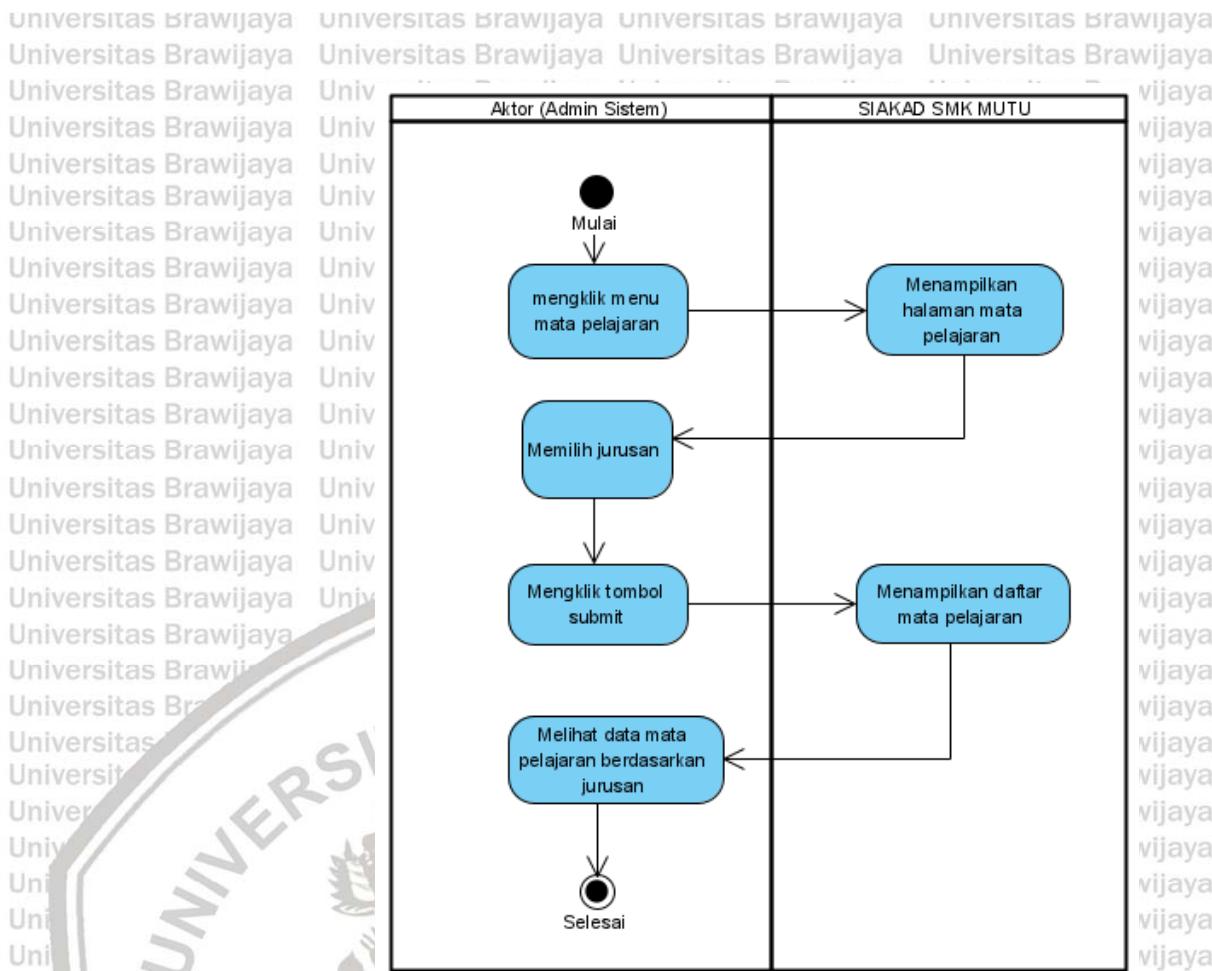
Gambar 4. 26 Activity Diagram Menetapkan Wali Kelas

6. Mengelola Data Mata Pelajaran

Activity diagram mengelola data mata pelajaran terdiri dari 4 (empat) *sub-activity* yang meliputi penambahan data mata pelajaran, pengubahan informasi suatu mata pelajaran, pembacaan data mata pelajaran, dan penghapusan data mata pelajaran. Aktor pada *activity* ini adalah admin sistem. Masing-masing *activity* dalam pengelolaan data kelas adalah sebagai berikut:

A. Melihat Data Mata Pelajaran Berdasarkan Jurusan

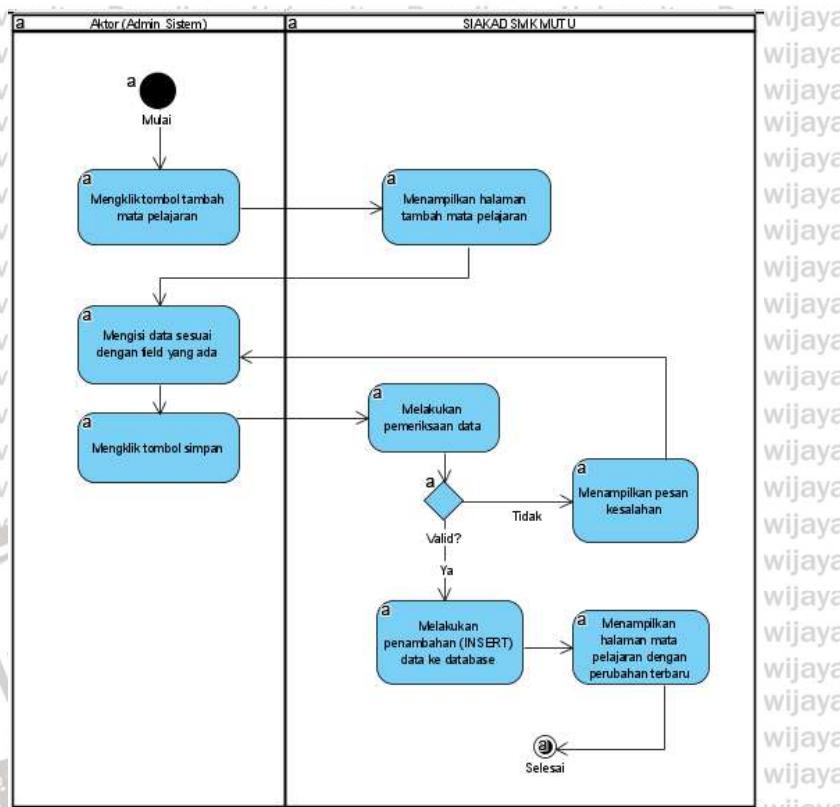
Activity melihat data kelas memungkinkan aktor untuk dapat melihat data-data mengenai mata pelajaran di setiap jurusan yang saat ini ada di SMK Muhammadiyah 1 Kemlagi. *Activity diagram* untuk kegiatan melihat data mata pelajaran berdasarkan jurusan dapat dilihat pada Gambar 4. 27.



Gambar 4. 27 Activity Diagram Melihat Data Mata Pelajaran Berdasarkan Jurusan

B. Menambah Data Mata Pelajaran

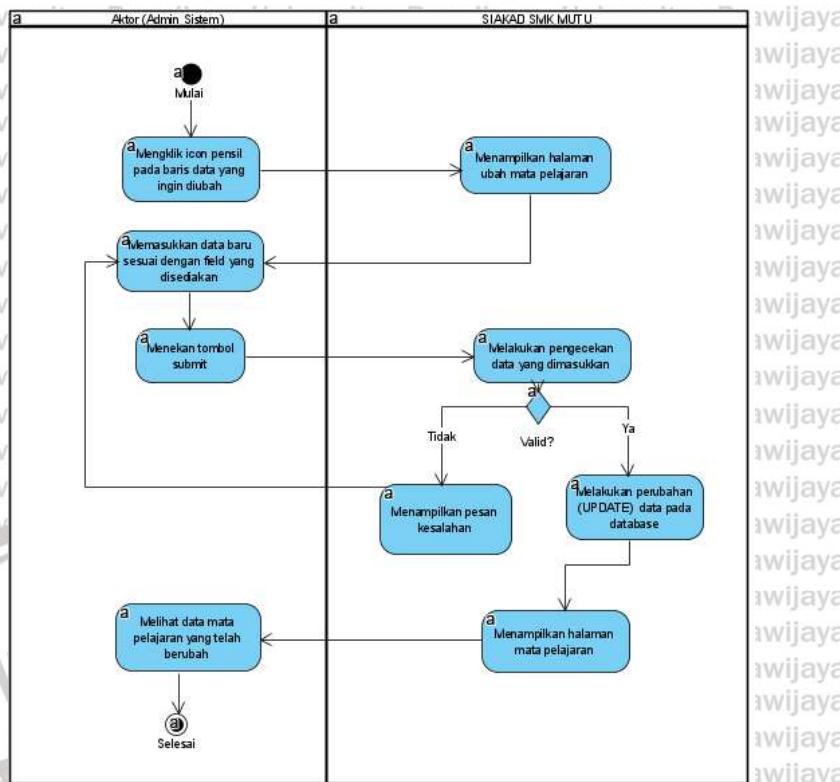
Activity diagram menambah data mata pelajaran merupakan model diagram untuk memasukkan data mata pelajaran baru pada database melalui sistem informasi. Aktor pada activity ini adalah admin sistem. Activity diagram untuk kegiatan menambah data kelas pada Gambar 4. 28.



Gambar 4. 28 Activity Diagram Menambah Data Mata Pelajaran.

C. Mengubah Data Mata Pelajaran

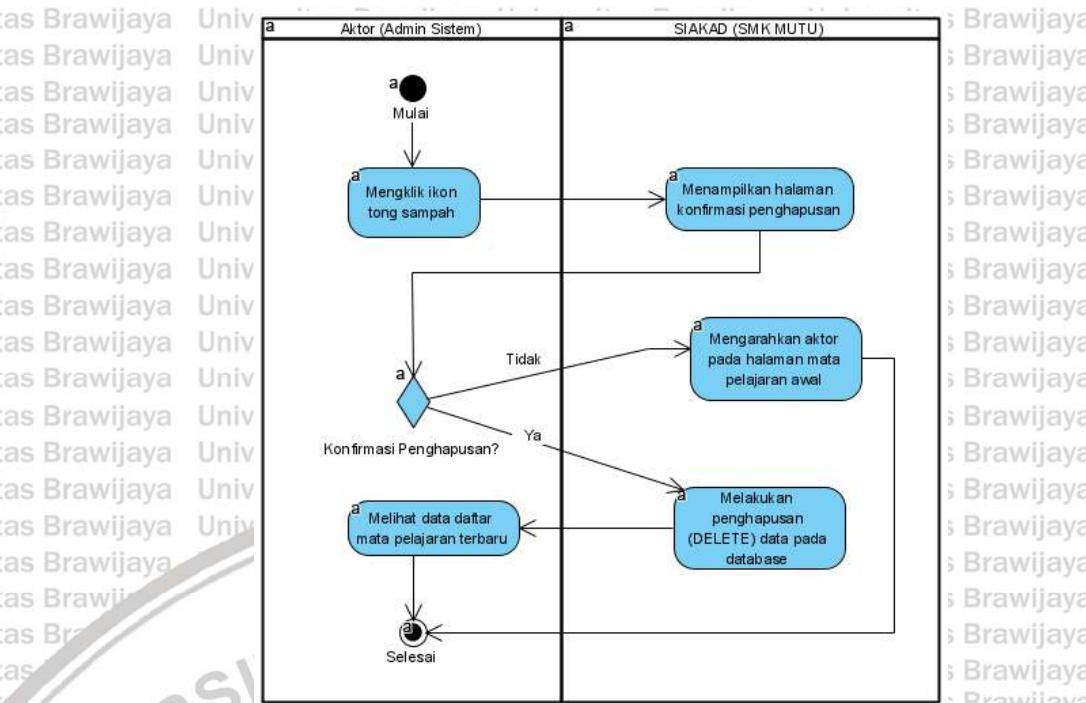
Activity diagram mengubah data mata pelajaran merupakan model diagram untuk melakukan perubahan informasi pada suatu kelas yang sudah ada pada database melalui sistem informasi. Aktor pada activity ini adalah admin sistem. Activity diagram untuk kegiatan mengubah data kelas dapat dilihat pada Gambar 4. 29.



Gambar 4. 29 Activity Diagram Mengubah Data Mata Pelajaran

D. Menghapus Data Mata Pelajaran

Activity diagram menghapus data kelas merupakan model diagram untuk melakukan penghapusan suatu mata pelajaran yang sudah ada pada *database* melalui sistem informasi. Aktor pada *activity* ini adalah admin sistem. *Activity diagram* untuk kegiatan menghapus data mata pelajaran dapat dilihat pada Gambar 4. 30.



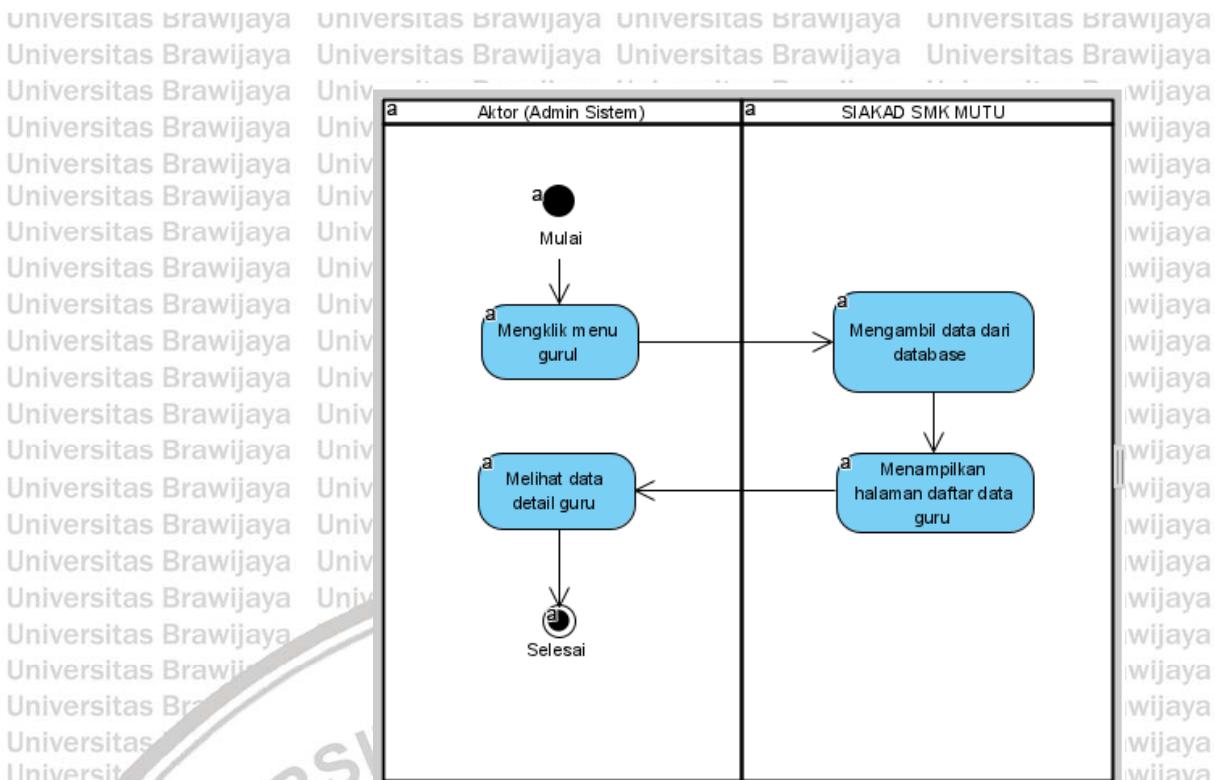
Gambar 4. 30 Activity Diagram Menghapus Mata Pelajaran

7. Mengelola Data Guru

Activity diagram mengelola data guru terdiri dari 5 (lima) *sub-activity* yang meliputi penambahan guru, pengubahan informasi guru, pembacaan data guru, pembacaan data lengkap guru, dan penghapusan data guru. Aktor pada *activity* ini adalah admin sistem. Masing-masing *activity* dalam pengelolaan data guru adalah sebagai berikut:

A. Melihat Data Guru

Activity melihat data guru memungkinkan aktor untuk dapat melihat data-data mengenai daftar guru yang saat ini mengajar di SMK Muhammadiyah 1 Kemlagi. *Activity diagram* untuk kegiatan melihat data guru dapat dilihat pada Gambar 4. 31.

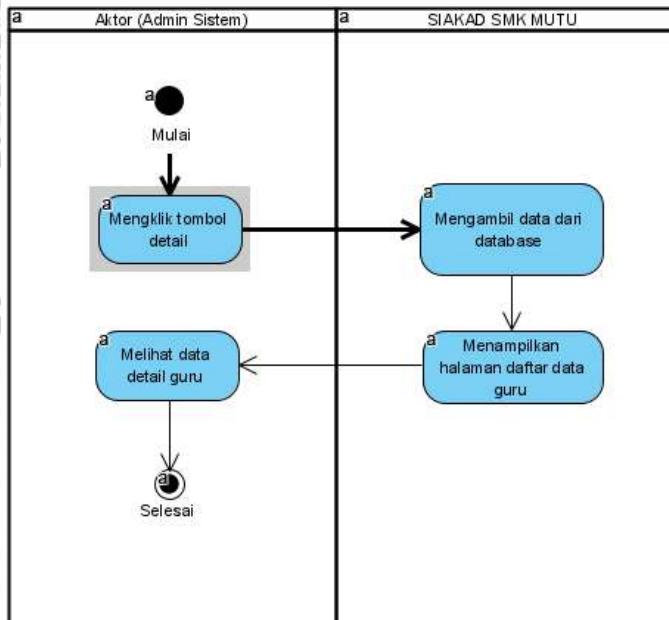


Gambar 4. 31 Activity Diagram Melihat Data Guru

B. Melihat Detail Guru

Activity melihat detail data guru memungkinkan aktor untuk dapat melihat data keseluruhan mengenai guru yang dituju. *Activity diagram* untuk kegiatan melihat detail data guru dapat dilihat pada

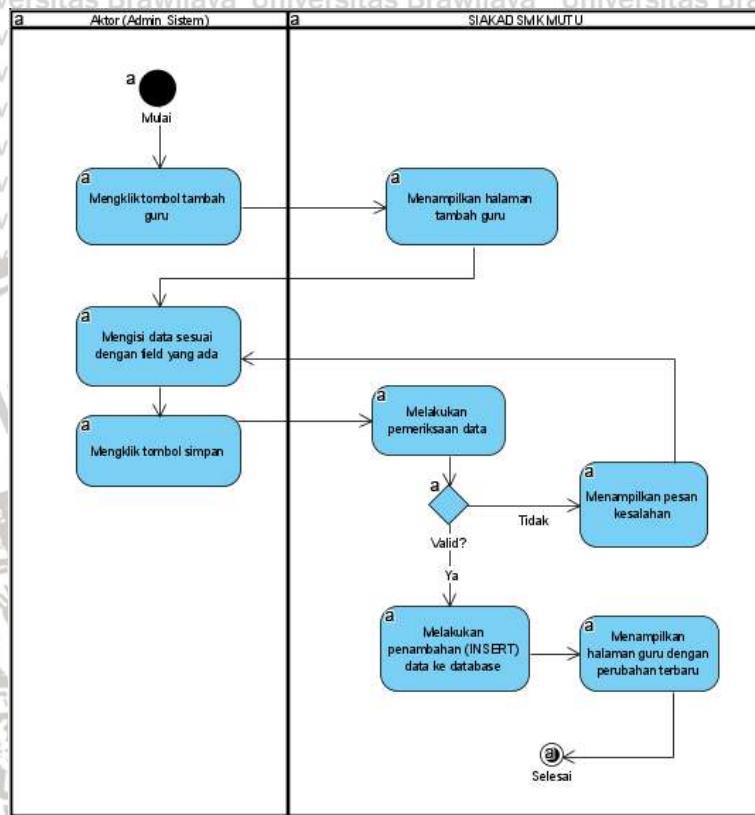
Gambar 4. 32.



Gambar 4. 32 Activity Diagram Melihat Detail Data Guru

C. Menambah Data Guru

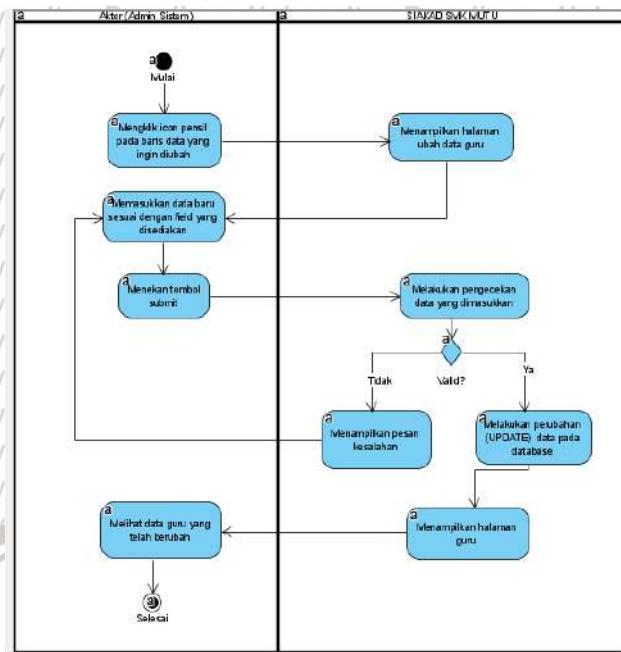
Activity diagram menambah data guru merupakan model diagram untuk memasukkan data guru baru pada *database* melalui sistem informasi. Aktor pada *activity* ini adalah admin sistem. *Activity diagram* untuk kegiatan menambah data guru dapat dilihat pada Gambar 4. 33.



Gambar 4. 33 *Activity Diagram* menambah Data Guru

D. Mengubah Data Guru

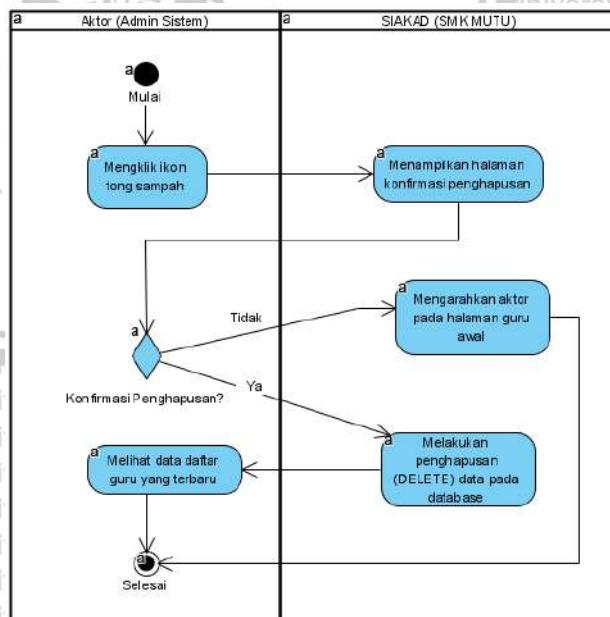
Activity diagram mengubah data guru merupakan model diagram untuk melakukan perubahan informasi pada salah satu guru yang sudah ada pada database melalui sistem informasi. Aktor pada *activity* ini adalah admin sistem. *Activity diagram* untuk kegiatan mengubah data guru dapat dilihat pada Gambar 4. 34.



Gambar 4. 34 Activity Diagram Mengubah Data Guru

E. Menghapus Data Guru

Activity diagram menghapus data guru merupakan model diagram untuk melakukan penghapusan data guru yang sudah ada pada *database* melalui sistem informasi. Aktor pada *activity* ini adalah admin sistem. *Activity diagram* untuk kegiatan menghapus data guru dapat dilihat pada Gambar 4.35.



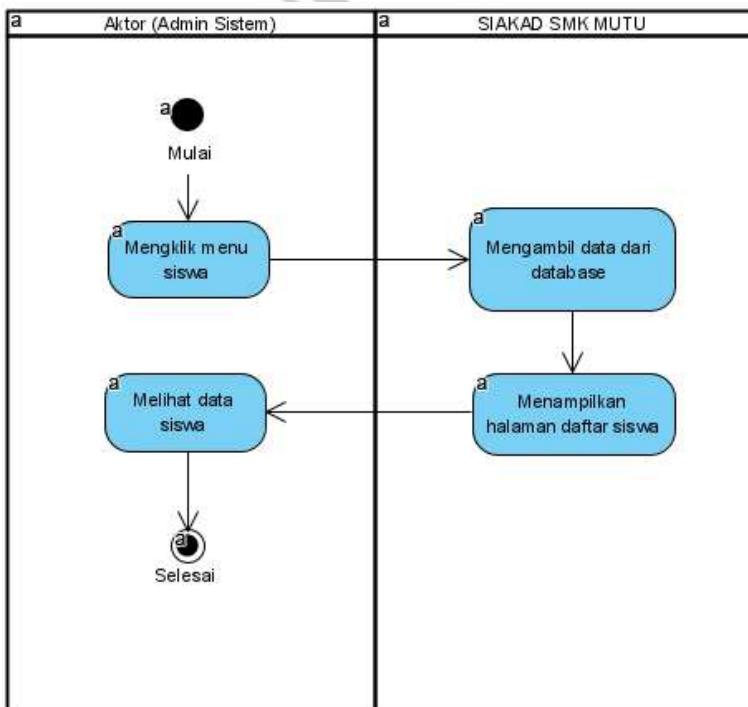
Gambar 4. 35 Activity Diagram Menghapus Data Guru

8. Mengelola Data Siswa

Activity diagram mengelola data siswa terdiri dari 7 (tujuh) *sub-activity* yang meliputi penambahan data siswa, pengubahan informasi siswa, pembacaan data siswa, pembacaan data lengkap siswa, pencarian data siswa, pengubahan status siswa dan penghapusan data siswa. Aktor pada *activity* ini adalah admin sistem. Masing-masing *activity* dalam pengelolaan data siswa adalah sebagai berikut:

A. Melihat Data Siswa

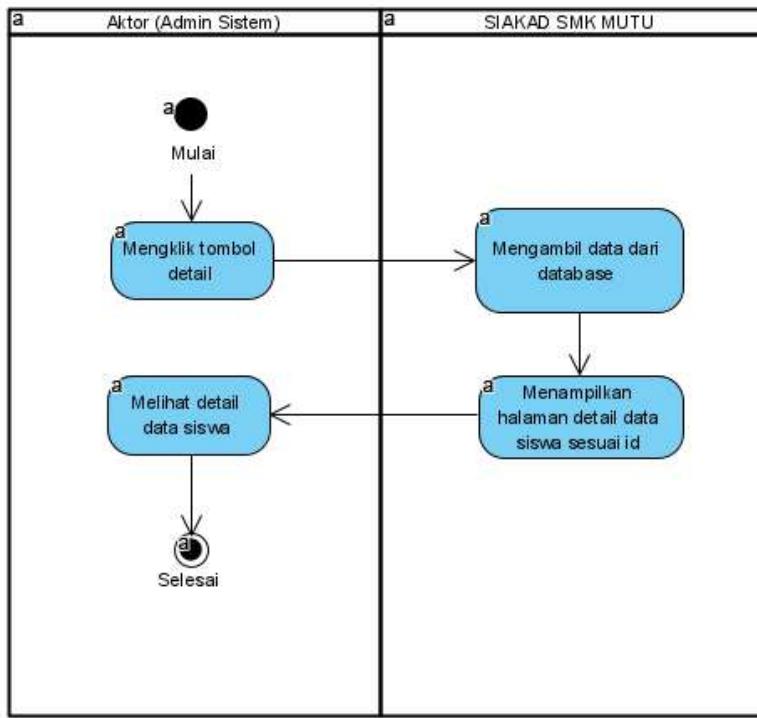
Activity melihat data siswa memungkinkan aktor untuk dapat melihat data-data mengenai daftar siswa yang saat ini belajar di SMK Muhammadiyah 1 Kemlagi. *Activity diagram* untuk kegiatan melihat data siswa dapat dilihat pada Gambar 4.36.



Gambar 4. 36 Activity Diagram Melihat Data Siswa

B. Melihat Detail Data Siswa

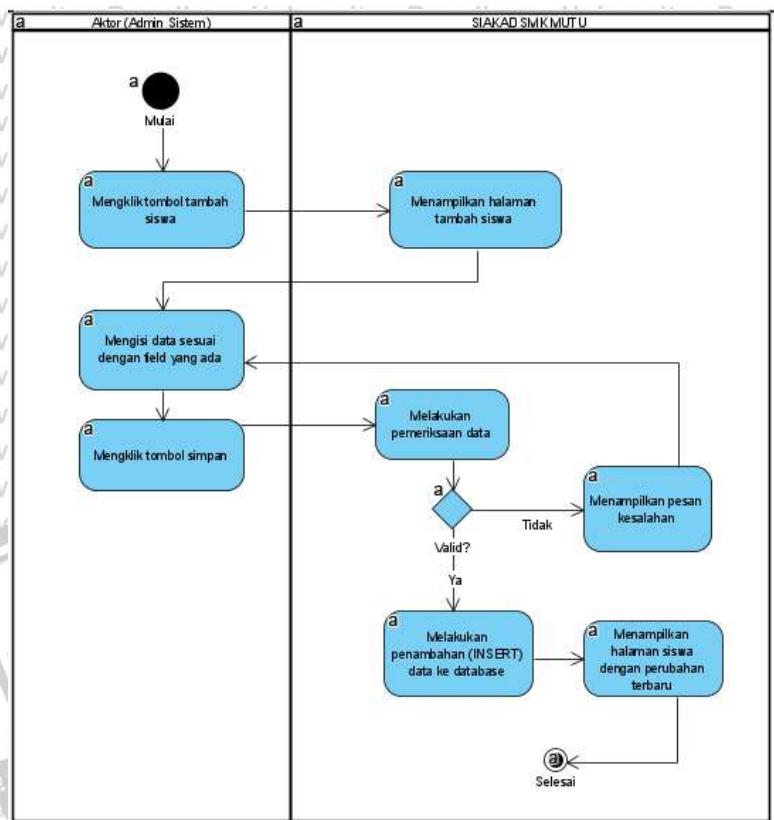
Activity melihat detail data siswa memungkinkan aktor untuk dapat melihat data keseluruhan mengenai siswa yang dituju. *Activity diagram* untuk kegiatan melihat detail data siswa dapat dilihat pada Gambar 4.37.



Gambar 4. 37 Activity Diagram Melihat Detail Siswa

C. Menambah Data Siswa

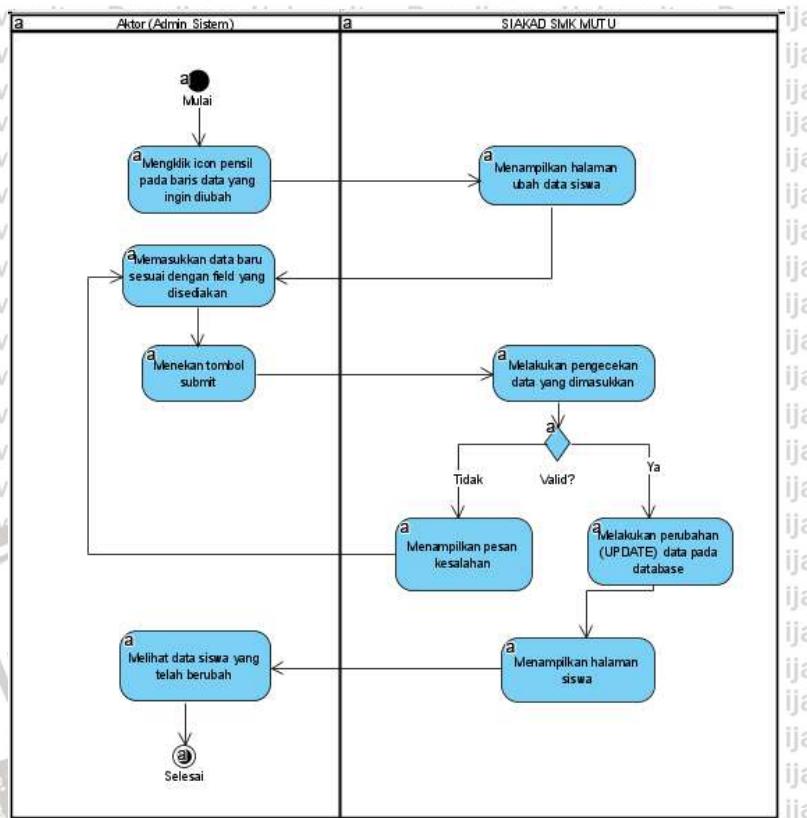
Activity diagram menambah data siswa merupakan model diagram untuk memasukkan data siswa baru pada *database* melalui sistem informasi. Aktor pada *activity* ini adalah admin sistem. *Activity diagram* untuk kegiatan menambah data siswa dapat dilihat pada Gambar 4. 38.



Gambar 4. 38 Activity Diagram Menambah Data Siswa

D. Mengubah Data Siswa

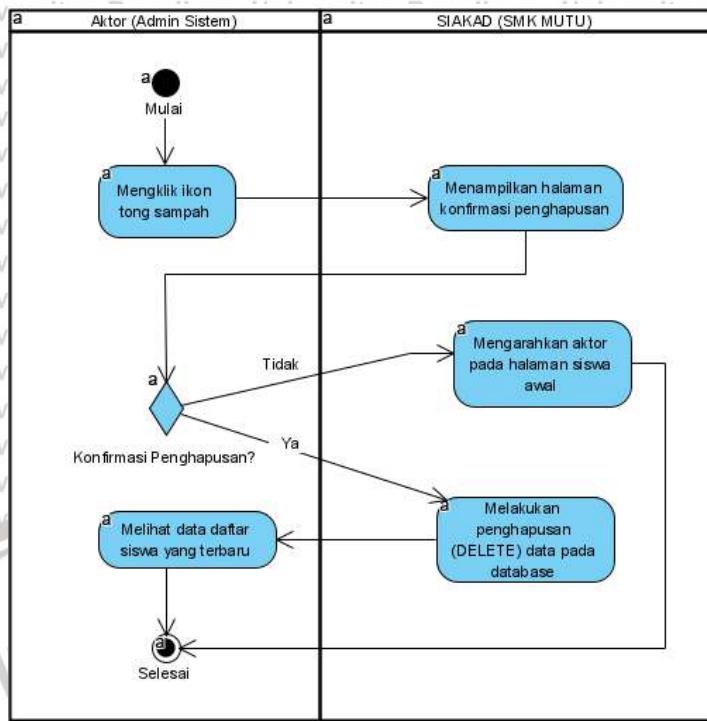
Activity diagram mengubah data siswa merupakan model diagram untuk melakukan perubahan informasi pada salah satu siswa yang sudah ada pada database melalui sistem informasi. Aktor pada *activity* ini adalah admin sistem. *Activity diagram* untuk kegiatan mengubah data siswa dapat dilihat pada Gambar 4. 39.



Gambar 4. 39 Activity Diagram Mengubah Data Siswa

E. Menghapus Data Siswa

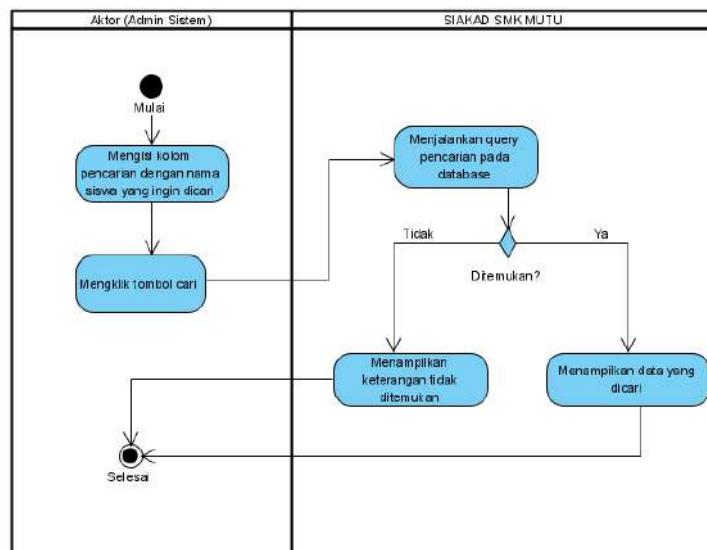
Activity diagram menghapus data siswa merupakan model diagram untuk melakukan penghapusan data siswa yang sudah ada pada *database* melalui sistem informasi. Aktor pada *activity* ini adalah admin sistem. *Activity diagram* untuk kegiatan menghapus data siswa dapat dilihat pada Gambar 4. 40.



Gambar 4. 40 Activity Diagram Menghapus Data Siswa

F. Mencari Data Siswa

Activity diagram mencari data siswa merupakan model diagram untuk melakukan pencarian data siswa yang sudah ada pada *database* melalui sistem informasi dengan memasukkan nama siswa yang dicari. Aktor pada *activity* ini adalah admin sistem. *Activity diagram* untuk kegiatan mencari data siswa dapat dilihat pada pada Gambar 4. 41.



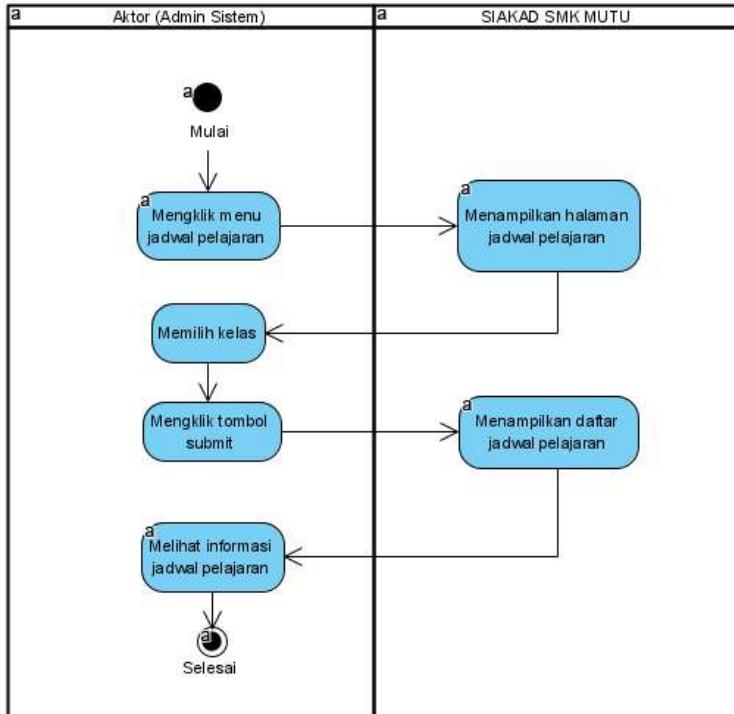
Gambar 4. 41 Activity Diagram Mencari Data Siswa

9. Mengelola Data Jadwal Pelajaran

Activity diagram mengelola data jadwal pelajaran terdiri dari 4 (empat) *sub-activity* yang meliputi penambahan data jadwal pelajaran, pengubahan informasi suatu jadwal pelajaran, pembacaan data jadwal pelajaran, dan penghapusan data jadwal pelajaran. Aktor pada *activity* ini adalah admin sistem. Masing-masing *activity* dalam pengelolaan jadwal pelajaran adalah sebagai berikut:

A. Melihat Jadwal Pelajaran Berdasarkan Kelas

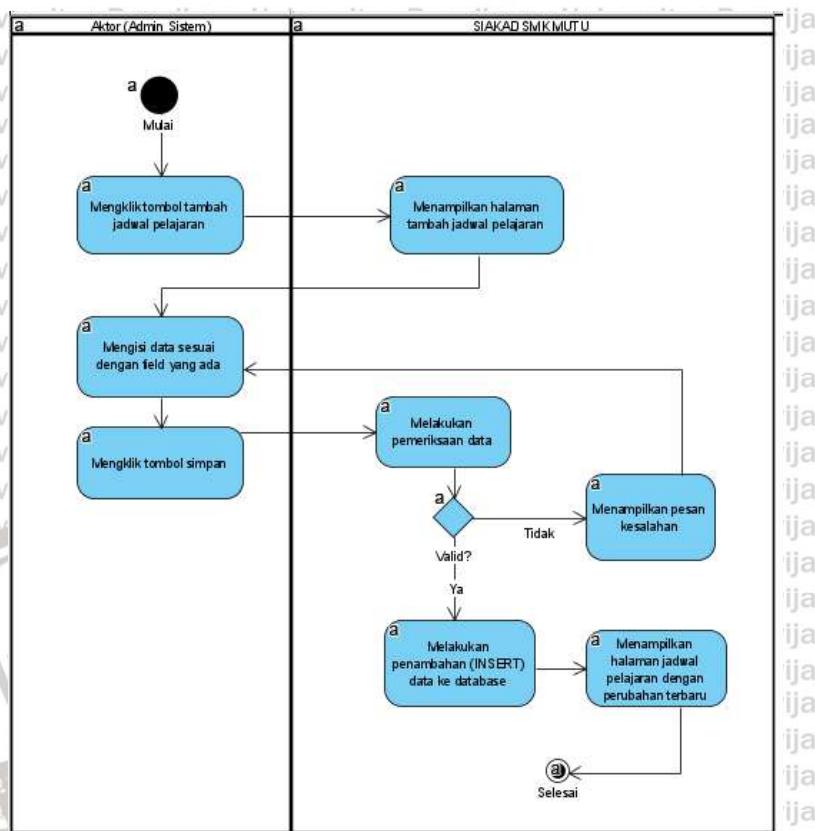
Activity melihat jadwal berdasarkan kelas memungkinkan aktor untuk dapat melihat data-data mengenai jadwal pelajaran pada masing-masing kelas yang saat ini ada di SMK Muhammadiyah 1 Kemlagi. *Activity diagram* untuk kegiatan melihat data jadwal pelajaran berdasarkan kelas dapat dilihat pada Gambar 4. 42.



Gambar 4. 42 Activity Diagram Melihat Data Jadwal Pelajaran Berdasarkan Kelas

B. Menambah Data Jadwal Pelajaran

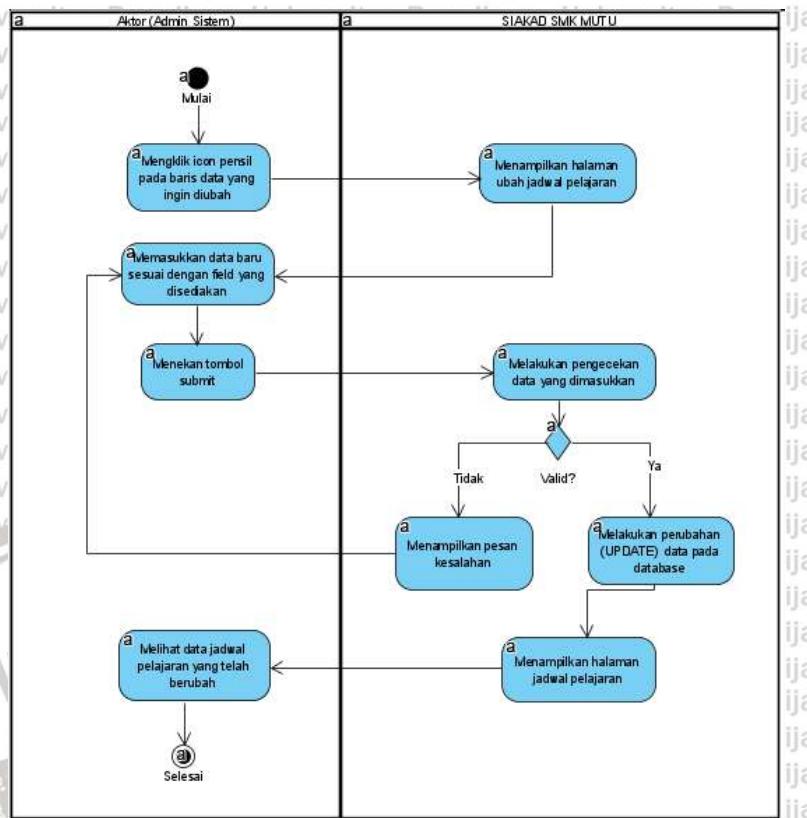
Activity diagram menambah data jadwal pelajaran merupakan model diagram untuk memasukkan data jadwal pelajaran baru pada database melalui sistem informasi. Aktor pada *activity* ini adalah admin sistem. *Activity diagram* untuk kegiatan menambah data jadwal pelajaran dapat dilihat pada Gambar 4. 43.



Gambar 4. 43 Activity Diagram Menambah Data Jadwal Pelajaran

C. Mengubah Data Jadwal Pelajaran

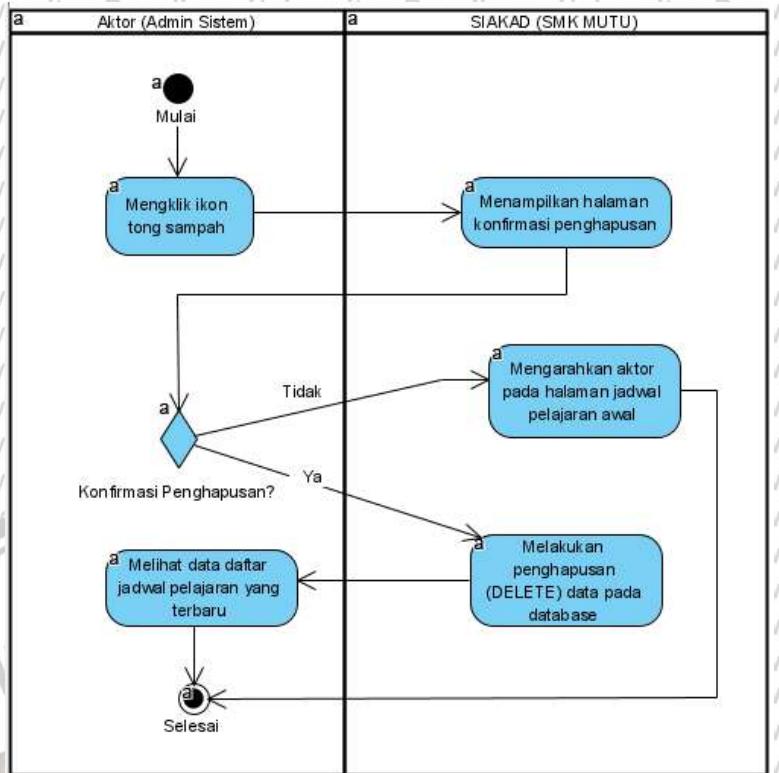
Activity diagram mengubah data jadwal pelajaran merupakan model diagram untuk melakukan perubahan informasi pada salah satu jadwal pelajaran yang sudah ada pada database melalui sistem informasi. Aktor pada *activity* ini adalah admin sistem. *Activity diagram* untuk kegiatan mengubah jadwal pelajaran dapat dilihat pada Gambar 4. 44.



Gambar 4. 44 Activity Diagram Mengubah Data Jadwal Pelajaran

D. Menghapus Data Jadwal Pelajaran

Activity diagram menghapus data jadwal pelajaran merupakan model diagram untuk melakukan penghapusan data jadwal pelajaran yang sudah ada pada *database* melalui sistem informasi. Aktor pada *activity* ini adalah admin sistem. *Activity diagram* untuk kegiatan menghapus data jadwal pelajaran dapat dilihat pada Gambar 4. 45.



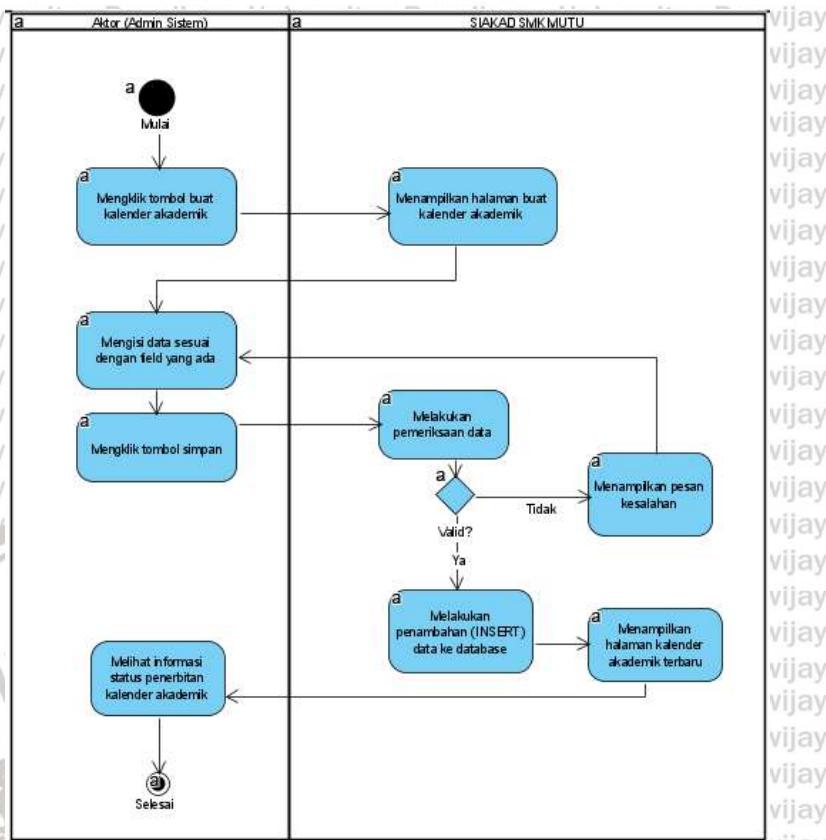
Gambar 4. 45 Activity Diagram menghapus Data Jadwal Pelajaran

10. Mengelola Kalender Akademik

Activity diagram mengelola kalender akademik untuk aktor admin sistem terdiri dari 3 (tiga) sub-activity yang meliputi pembuatan kalender akademik, pengubahan kalender akademik, dan penghapusan kalender akademik. Aktor pada activity ini adalah admin sistem. Masing-masing activity dalam pengelolaan jadwal pelajaran adalah sebagai berikut:

A. Membuat Kalender Akademik

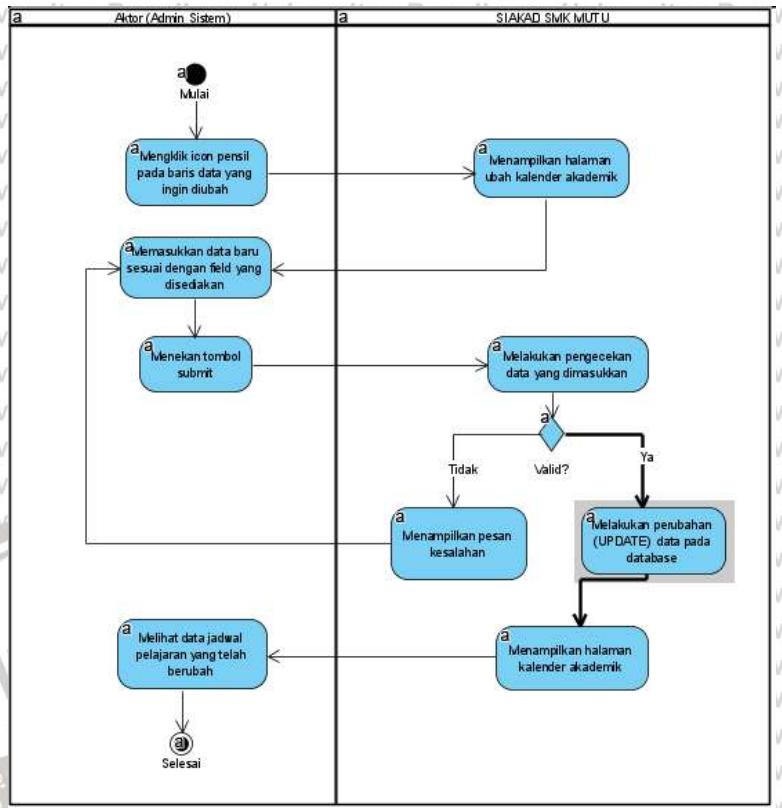
Activity diagram membuat kalender akademik merupakan model diagram untuk membuat kalender akademik baru pada database melalui sistem informasi. Aktor pada activity ini adalah admin sistem. Activity diagram untuk aktivitas membuat data kegiatan akademik dapat dilihat pada Gambar 4. 46.



Gambar 4. 46 Activity Diagram Membuat Kegiatan Akademik

B. Mengubah Kegiatan Akademik

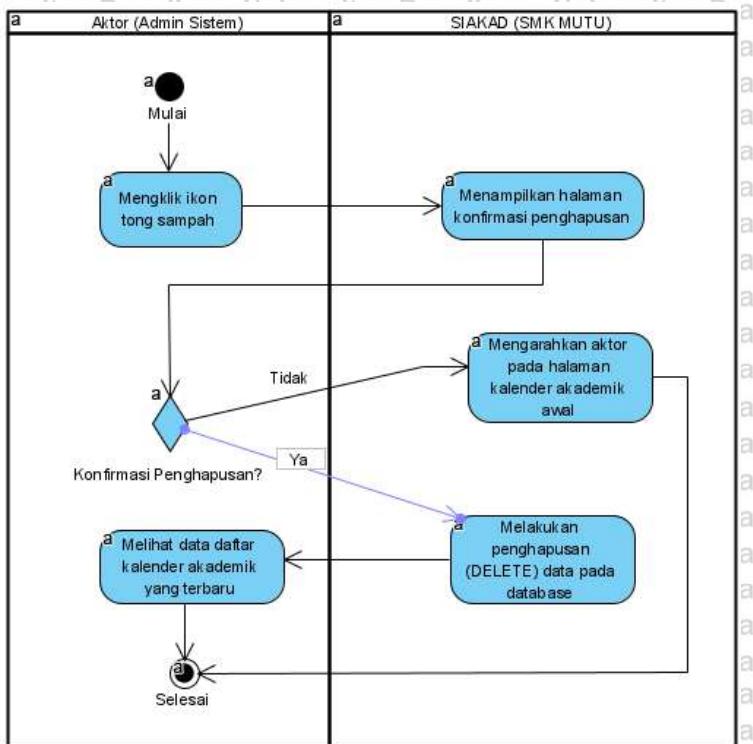
Activity diagram mengubah data kegiatan akademik merupakan model diagram untuk melakukan perubahan informasi pada salah satu kegiatan akademik yang sudah ada pada database melalui sistem informasi. Aktor pada activity ini adalah admin sistem. Activity diagram untuk kegiatan mengubah kegiatan akademik dapat dilihat pada Gambar 4. 47.



Gambar 4. 47 *Activity Diagram* Mengubah Kegiatan Akademik

C. Menghapus Kegiatan Akademik

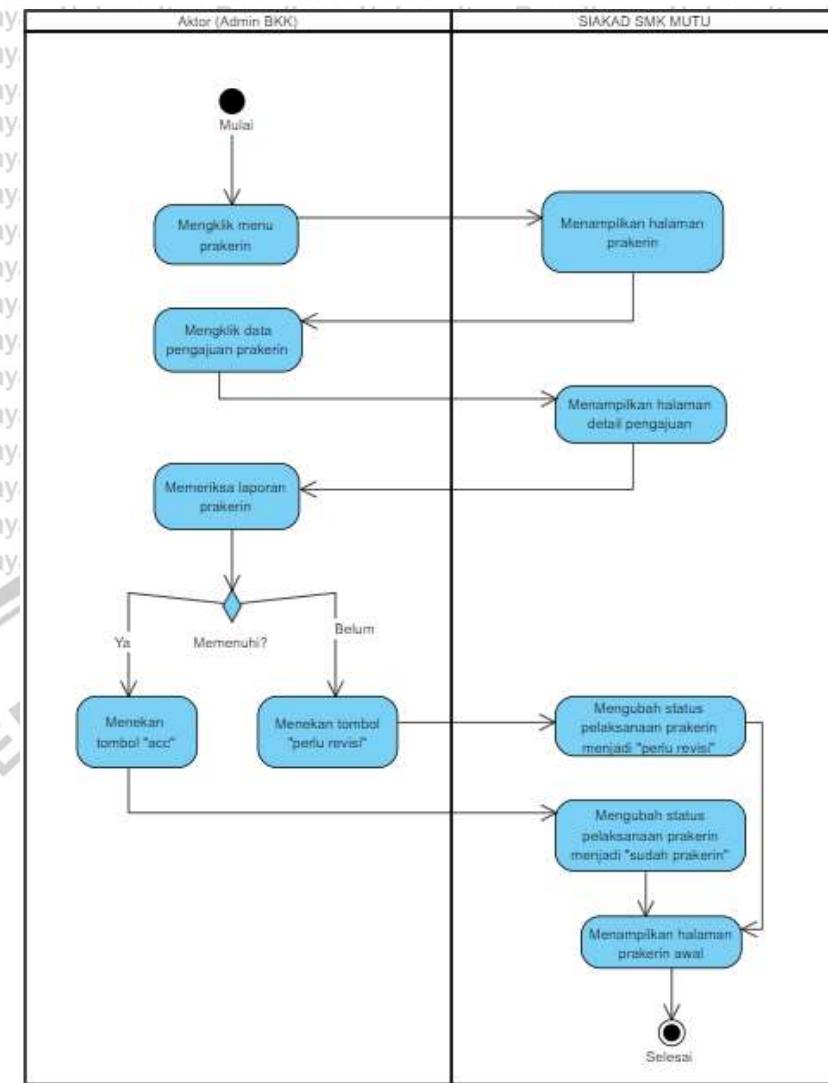
Activity diagram menghapus data kalender akademik merupakan model diagram untuk melakukan penghapusan kalender akademik yang sudah ada pada *database* melalui sistem informasi. Aktor pada *activity* ini adalah admin sistem. *Activity diagram* untuk kegiatan menghapus data kegiatan akademik dapat dilihat pada Gambar 4. 48.



Gambar 4. 48 Activity Diagram Menghapus Kegiatan Akademik

11. Memvalidasi Pelaksanaan Prakerin Siswa

Activity diagram memvalidasi pelaksanaan prakerin merupakan sebuah aktivitas untuk memvalidasi dan melakukan pendataan mengenai kegiatan praktik kerja industri siswa, aktor pada aktivitas ini adalah admin BKK, pada aktivitas ini admin BKK akan melakukan pengecekan pada laporan prakerin siswa dan memutuskan apakah laporan tersebut sudah layak atau belum dengan menerapkan sistem informasi. *Activity diagram* memvalidasi pelaksanaan prakerin siswa dapat dilihat pada Gambar 4.49.



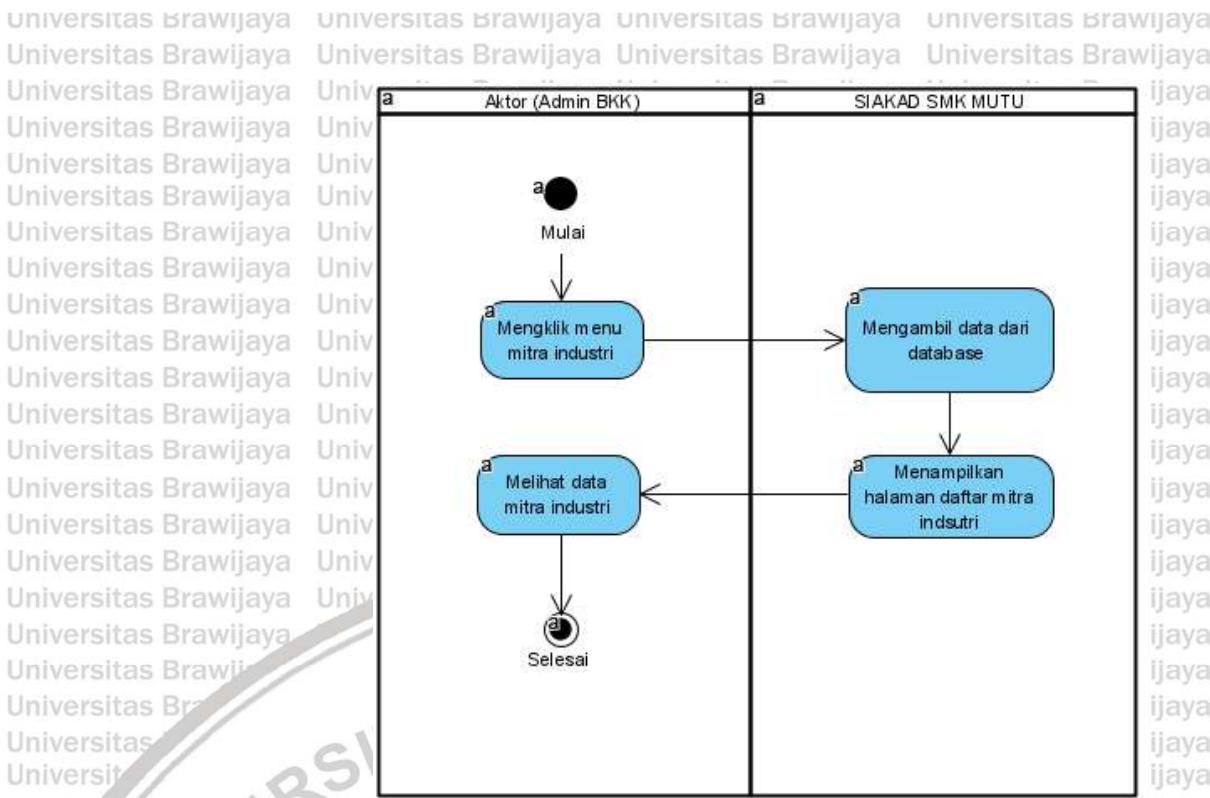
Gambar 4. 49 Activity Diagram Memvalidasi Prakerin Siswa

12. Mengelola Data Mitra Industri

Activity diagram mengelola data mitra industri terdiri dari 4 (empat) sub-activity yang meliputi penambahan data mitra industri, pengubahan informasi mitra industri, pembacaan data mitra industri, dan penghapusan data mitra industri. Aktor pada activity ini adalah admin BKK. Masing-masing activity dalam pengelolaan mitra industri adalah sebagai berikut:

A. Melihat Data Mitra Industri

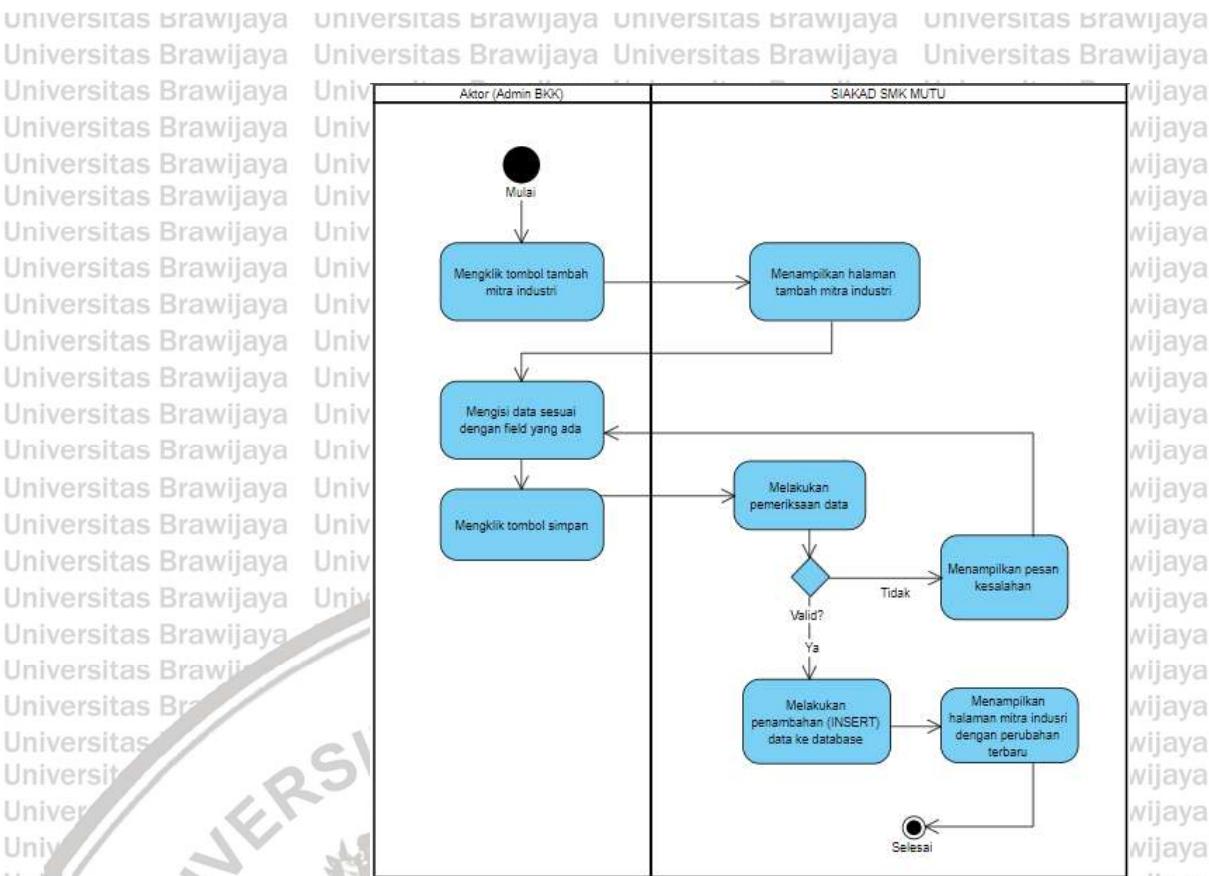
Activity melihat data mitra industri memungkinkan aktor untuk dapat melihat data-data mengenai daftar industri yang bermitra dengan SMK Muhammadiyah 1 Kemlagi. Activity diagram untuk kegiatan melihat data mitra industri dapat dilihat pada Gambar 4. 50.



Gambar 4. 50 Activity Diagram Melihat Mitra Industri

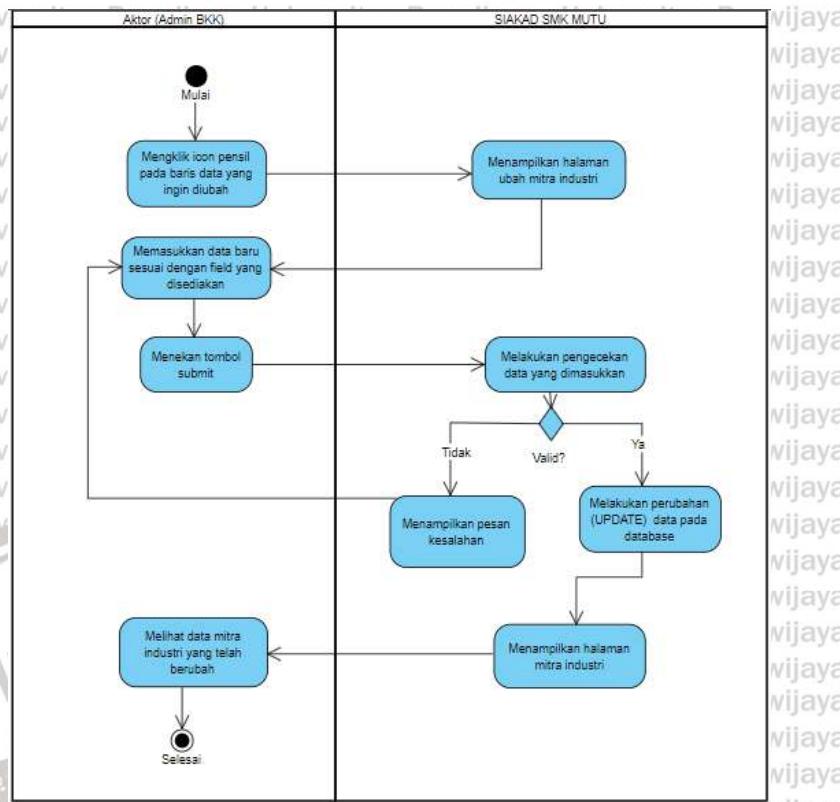
B. Menambah Data Mitra Industri

Activity diagram menambah data mitra industri merupakan model diagram untuk memasukkan data jadwal mitra industri baru pada database melalui sistem informasi. Aktor pada activity ini adalah admin BKK. Activity diagram untuk kegiatan menambah data mitra industri dapat dilihat pada Gambar 4. 51.

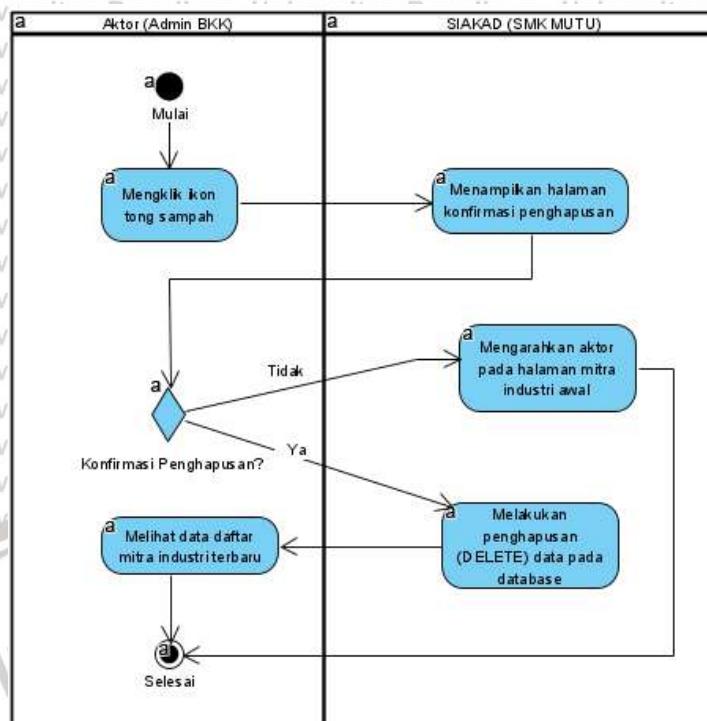
Gambar 4. 51 *Activity Diagram Menambah Mitra Industri*

C. Mengubah Data Mitra Industri

Activity diagram mengubah data mitra industri merupakan model diagram untuk melakukan perubahan informasi pada salah satu mitra industri yang sudah ada pada database melalui sistem informasi. Aktor pada *activity* ini adalah admin BKK. *Activity diagram* untuk kegiatan mengubah mitra industri dapat dilihat pada Gambar 4. 52.

**Gambar 4. 52 Activity Diagram Mengubah Mitra Industri****D. Menghapus Data Mitra Industri**

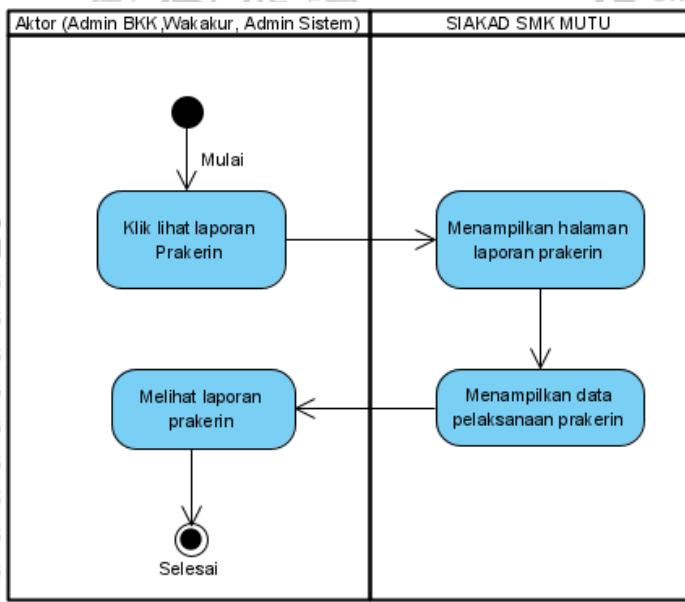
Activity diagram menghapus data mitra industri merupakan model diagram untuk melakukan penghapusan mitra industri yang sudah ada pada *database* melalui sistem informasi. Aktor pada *activity* ini adalah admin BKK. *Activity diagram* untuk kegiatan menghapus data mitra industri dapat dilihat pada Gambar 4. 53.



Gambar 4. 53 Activity Diagram Menghapus Mitra Industri

13. Melihat Data Pelaksanaan Prakerin

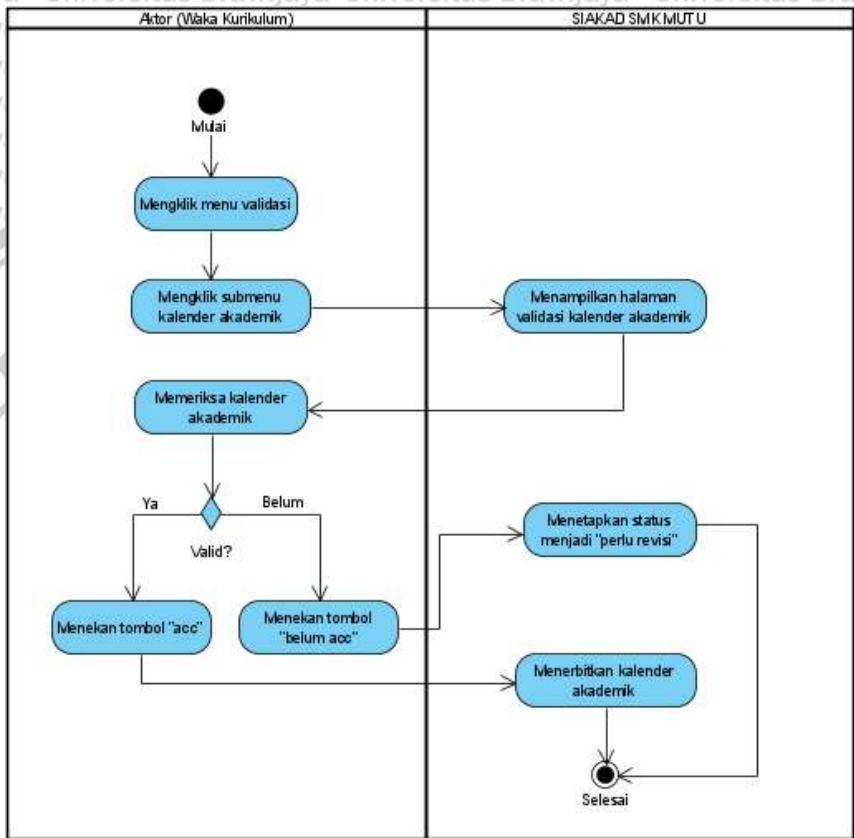
Activity diagram melihat data pelaksanaan prakerin memungkinkan aktor untuk melihat daftar pelaksanaan prakerin yang sudah divalidasi oleh admin BKK, aktor pada aktivitas ini adalah Waka Kurikulum, admin BKK dan Waka Kurikulm. Activity diagram melihat data pelaksanaan prakerin dapat dilihat pada Gambar 4. 54.



Gambar 4. 54 Activity Diagram Melihat Data Pelaksanaan Prakerin

14. Memvalidasi Kalender Akademik

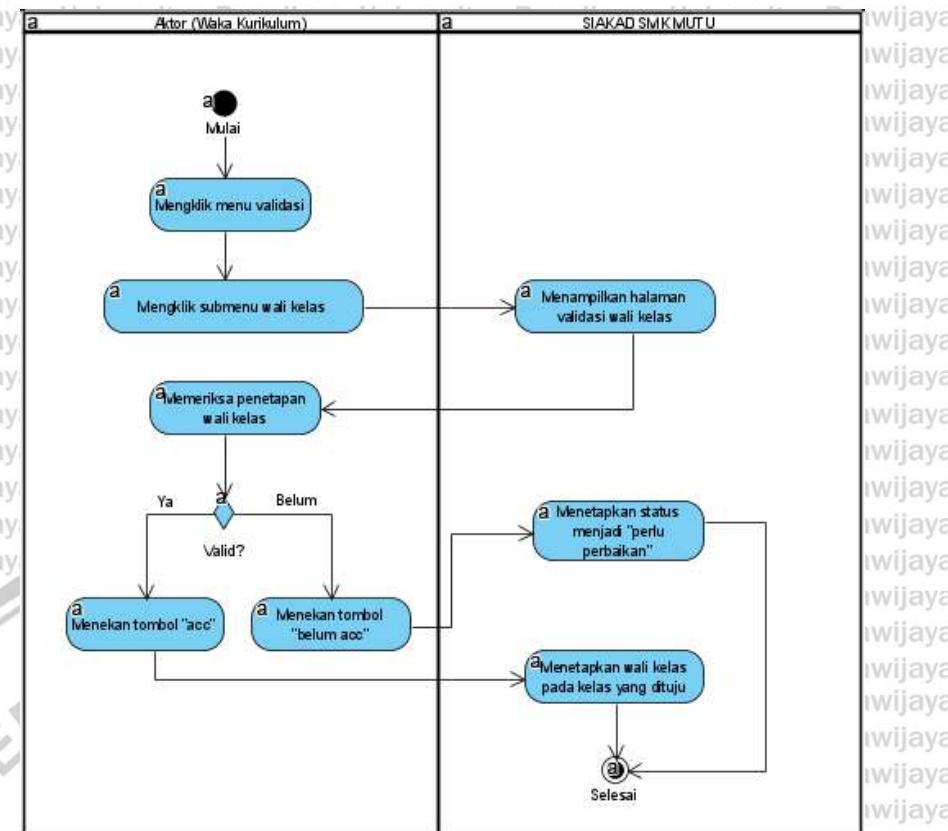
Activity diagram memvalidasi kalender akademik merupakan sebuah aktivitas untuk memvalidasi kalender akademik yang telah dibuat oleh admin sistem, aktor pada aktivitas ini adalah Waka Kurikulum, pada aktivitas ini Waka Kurikulum akan melakukan pengecekan pada kalender akademik yang telah dimasukkan oleh admin sistem, dan memutuskan apakah kalender tersebut sudah benar atau belum. *Activity diagram* memvalidasi kalender akademik dapat dilihat pada Gambar 4. 55.



Gambar 4. 55 *Activity Diagram* Memvalidasi Kalender Akademik

15. Memvalidasi Penetapan Wali Kelas

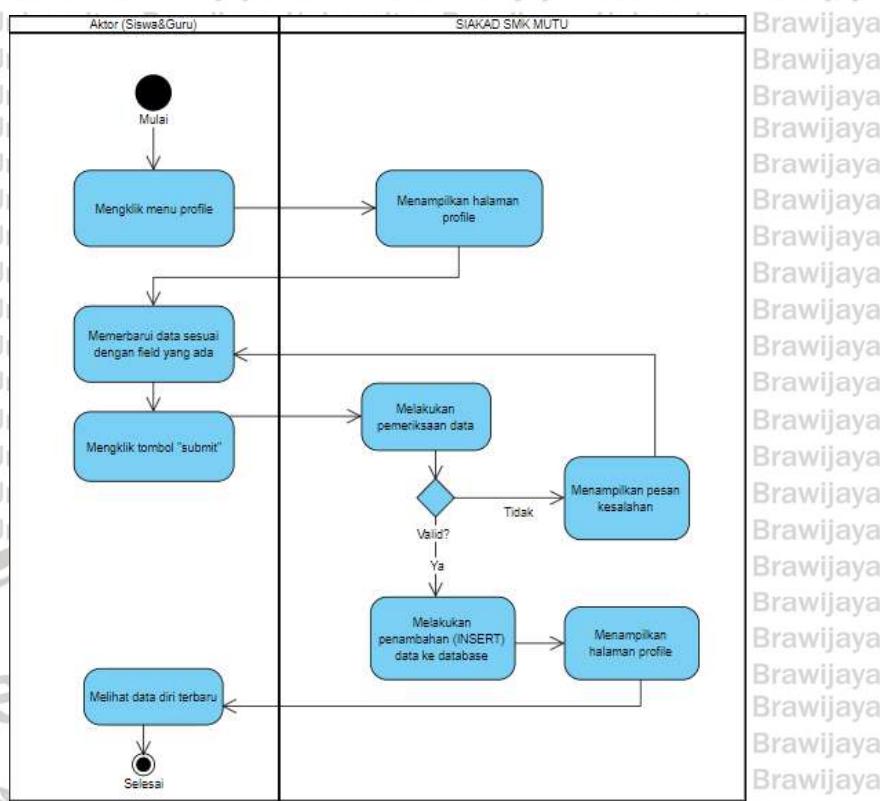
Activity diagram memvalidasi penetapan wali kelas merupakan sebuah aktivitas untuk memvalidasi aktivitas penetapan wali kelas yang telah dilakukan sebelumnya oleh admin sistem, aktor pada aktivitas ini adalah Waka Kurikulum, pada aktivitas ini Waka Kurikulum akan melakukan pengecekan pada daftar penempatan wali kelas yang telah dimasukkan oleh admin sistem, dan memutuskan apakah penempatan tersebut sudah benar atau belum. *Activity diagram* memvalidasi penetapan wali kelas dapat dilihat pada Gambar 4. 56.



Gambar 4. 56 *Activity Diagram Menetapkan Wali Kelas*

16. Memperbarui Data Diri

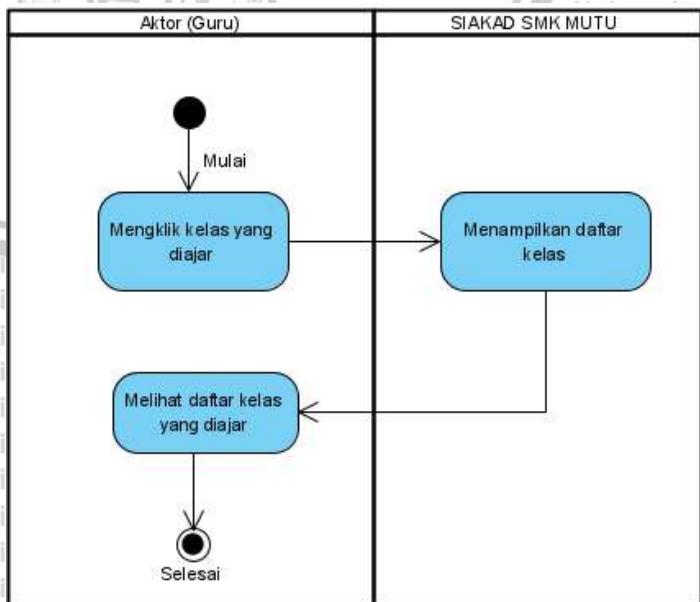
Activity diagram memperbarui data diri memungkinkan aktor untuk memperbarui data diri secara berkala. Aktor pada aktivitas ini meliputi guru dan siswa. *Activity diagram* memperbarui data diri dapat dilihat pada Gambar 4. 57.



Gambar 4. 57 Activity Diagram Memperbarui Data Diri

17. Melihat Daftar Kelas Yang Diwakili

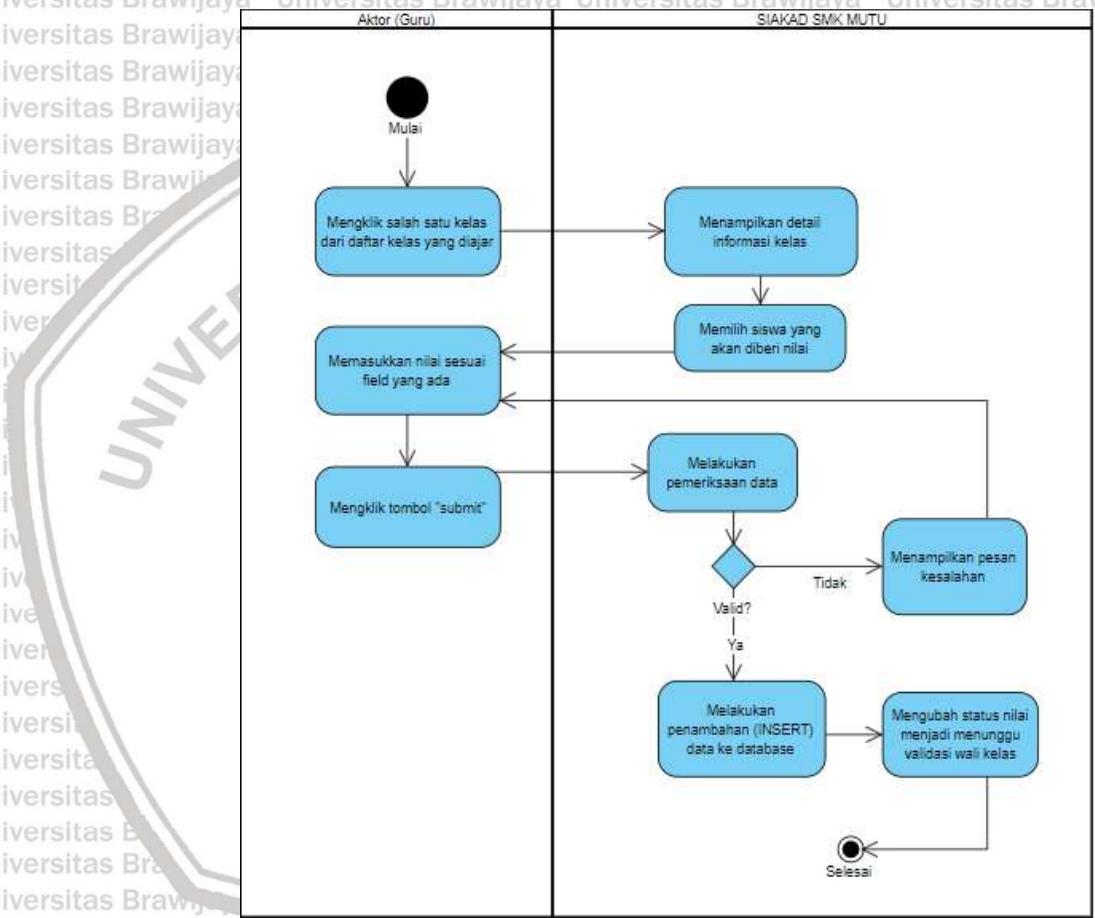
Activity diagram melihat daftar kelas yang diajar memungkinkan aktor dalam hal ini guru untuk mengetahui mengenai daftar kelas yang diajar pada tahun ajaran saat ini. Activity diagram melihat daftar kelas yang diajar dapat dilihat pada Gambar 4. 58.



Gambar 4. 58 Activity Diagram Melihat Kelas Yang Diajarkan

18. Memasukkan Nilai Hasil Evaluasi Siswa

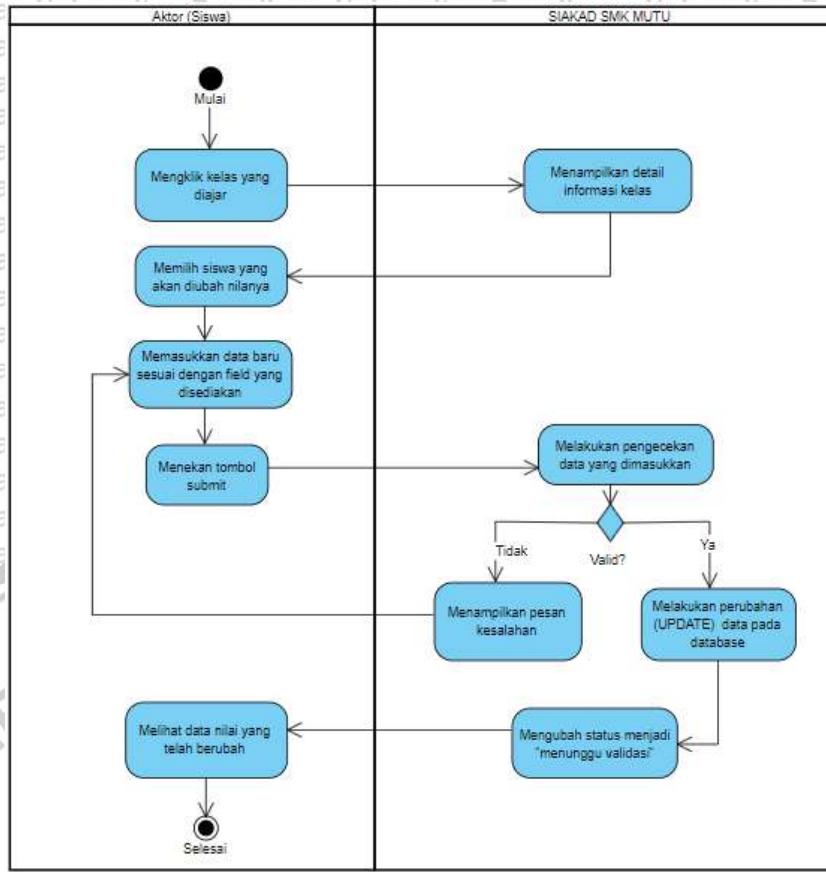
Brawijaya *Activity diagram* memasukkan nilai hasil evaluasi siswa memungkinkan aktor dalam hal ini guru untuk memasukkan nilai hasil evaluasi siswa melalui sistem informasi. Nilai yang dimasukkan oleh guru nantinya akan ditinjau oleh wali kelas untuk dilakukan proses lebih lanjut apabila diperlukan. *Activity diagram* memasukkan hasil evaluasi siswa dapat dilihat pada Gambar 4. 59.



Gambar 4. 59 Activity Diagram Memasukkan Nilai Hasil Evaluasi Siswa

19. Mengubah Nilai Hasil Evaluasi Siswa

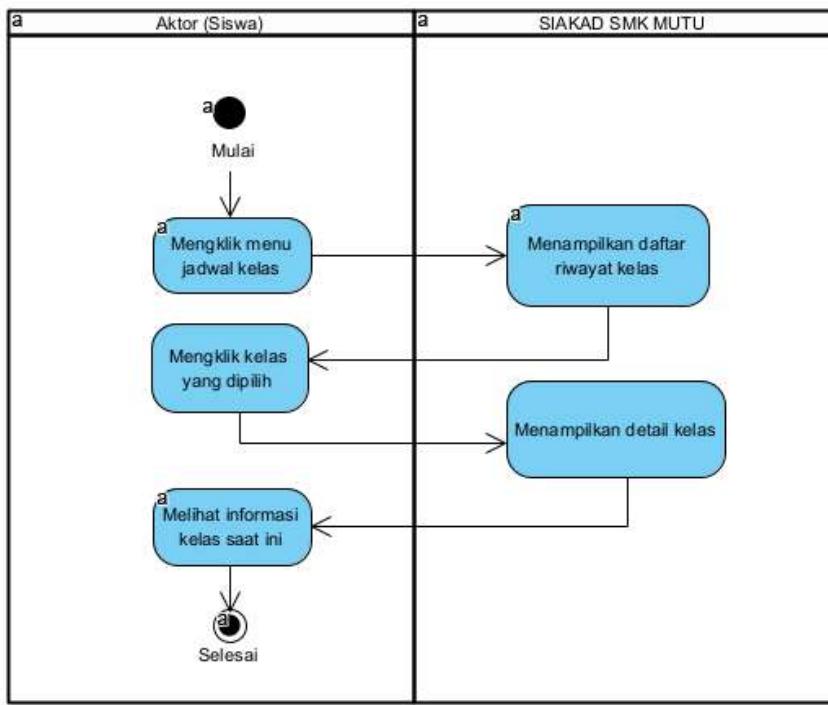
Activity diagram mengubah hasil nilai merupakan pemodelan aktivitas tindak lanjut apabila wali kelas meminta kompromi nilai pada satu atau lebih siswa, aktor pada aktivitas ini adalah guru mata pelajaran, guru mata pelajaran hanya diperkenankan mengubah nilai hasil evaluasi siswa apabila telah memberikan remedial atau penugasan pada siswa. *Activity diagram* mengubah hasil evaluasi siswa dapat dilihat pada Gambar 4.60.



Gambar 4. 60 Activity Diagram Mengubah Nilai Hasil Evaluasi

20. Melihat Detail Informasi Kelas

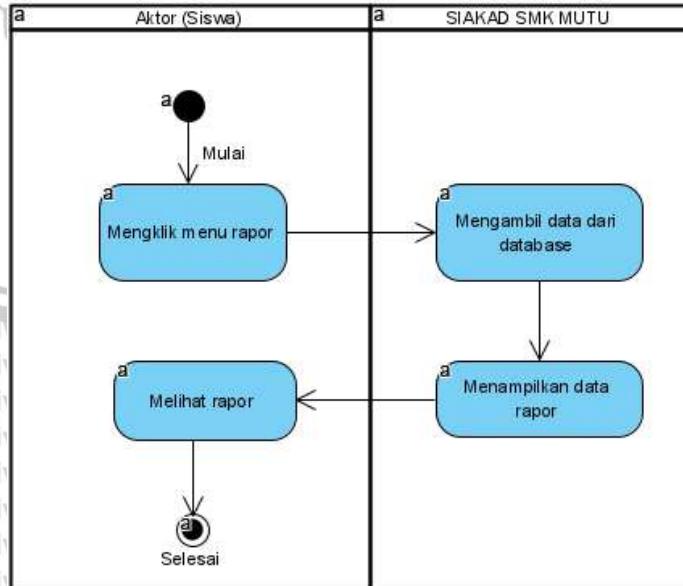
Activity diagram melihat informasi kelas saat ini merupakan pemodelan aktivitas yang dilakukan oleh siswa selaku aktor untuk mengetahui informasi-informasi dasar pada kelas yang didudukinya. *Activity diagram* melihat informasi kelas saat ini dapat dilihat pada Gambar 4. 61.



Gambar 4. 61 Activity Diagram Melihat Detail Kelas

21. Melihat Nilai Hasil Evaluasi

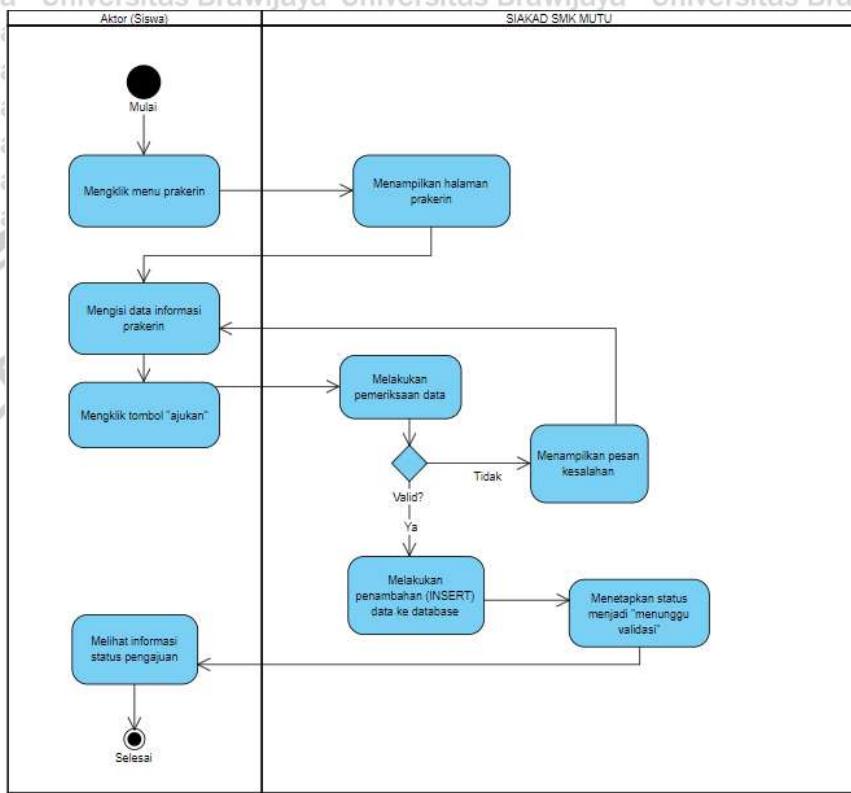
Activity diagram melihat nilai hasil evaluasi merupakan pemodelan aktivitas yang dilakukan oleh siswa selaku aktor untuk mengetahui nilai hasil evaluasi yang diperolehnya. Activity diagram melihat nilai hasil evaluasi dapat dilihat pada Gambar 4. 62.



Gambar 4. 62 Activity Diagram Melihat Hasil Evaluasi

22. Mengajukan Penuntasan Pelaksanaan Prakerin

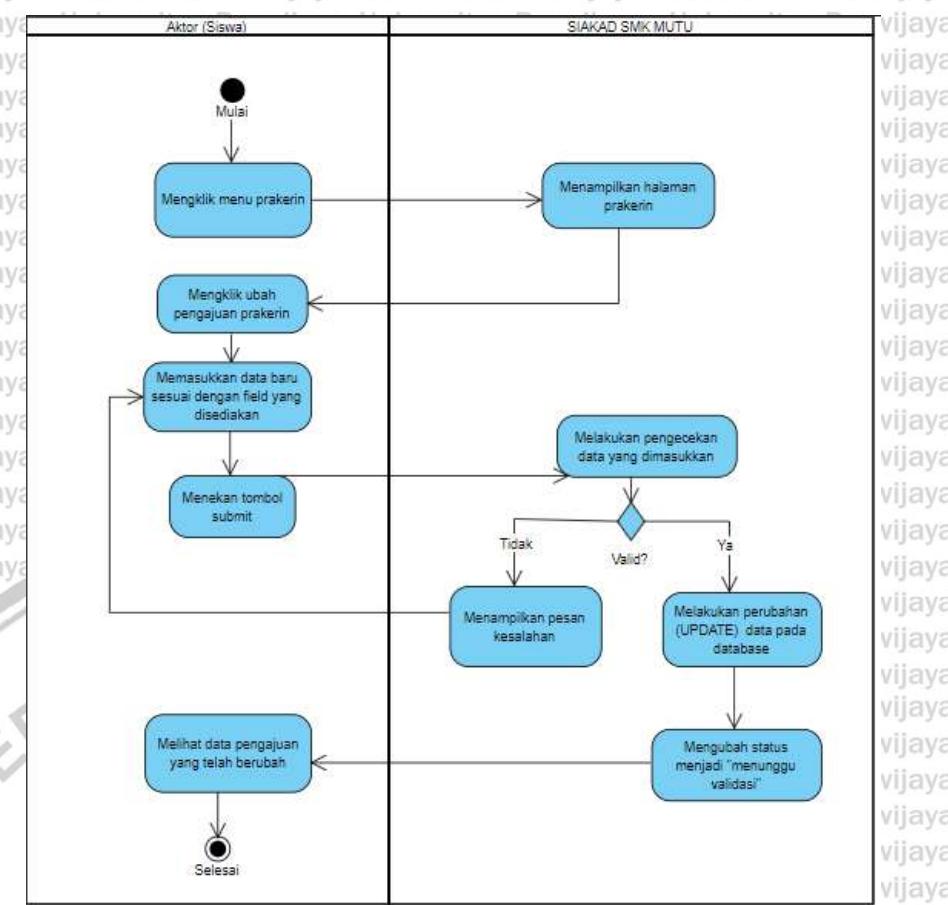
Activity diagram mengajukan penuntasan pelaksanaan prakerin merupakan pemodelan aktivitas yang dilakukan oleh siswa selaku aktor untuk memperoleh status telah melaksanakan kegiatan prakerin. Aktivitas pengajuan penuntasan pelaksanaan prakerin hanya boleh dilakukan oleh siswa, apabila siswa tersebut telah melaksanakan kegiatan prakerin dan menyelesaikan laporan pakerin. *Activity diagram* mengajukan penuntasan pelaksanaan prakerin dapat dilihat pada Gambar 4. 63.



Gambar 4. 63 *Activity Diagram* Pengajuan Penuntasan Prakerin

23. Mengubah Data Pelaksanaan Prakerin

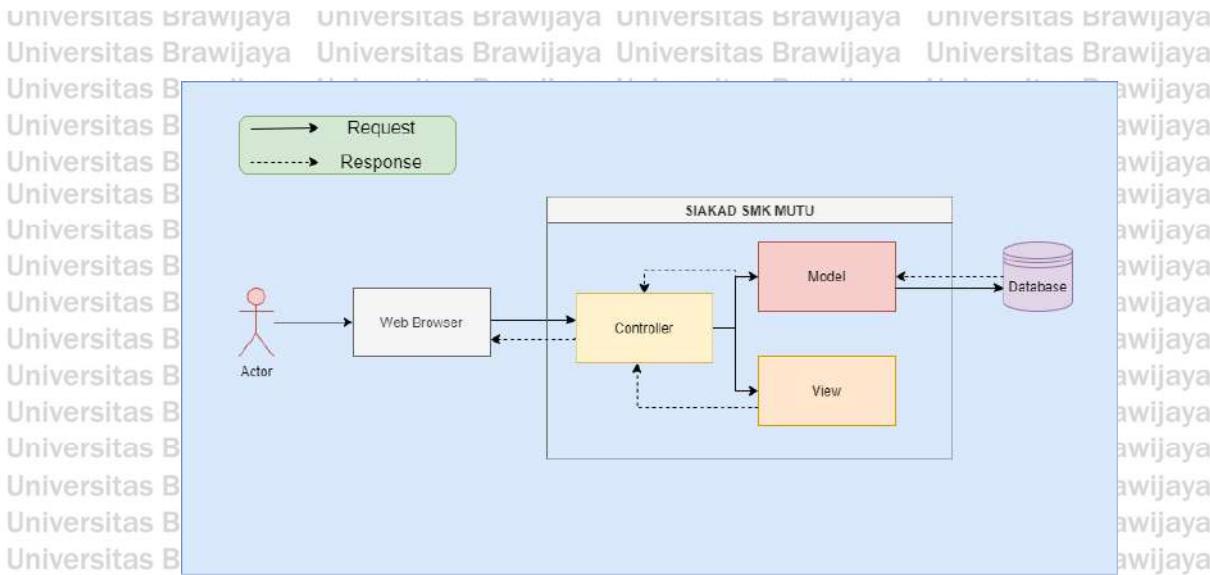
Activity diagram mengubah data pelaksanaan prakerin merupakan pemodelan aktivitas tindak lanjut apabila pengajuan penuntasan pelaksanaan prakerin dinyatakan belum memenuhi oleh admin BKK, aktor pada aktivitas ini adalah siswa, siswa perlu melakukan perubahan pada data yang dikirim sebelumnya. *Activity diagram* mengubah data pelaksanaan prakerin dapat dilihat pada Gambar 4. 64.



Gambar 4. 64 Activity Diagram Mengubah Data Pengajuan Prakerin

4.2.4 Arsitektur Sistem

Sistem informasi yang akan dibangun, menggunakan *framework* Codeigniter yang menerapkan arsitektur MVC (*Model, View, Controller*). Dalam pengaksesannya diperlukan peramban web versi terbaru (Mozilla Firefox atau Google Chrome). Karena merupakan sistem web dinamis maka sistem informasi akan melakukan manipulasi data dengan *database*. Koneksi dan pengelolaan data pada database akan dilakukan oleh *class-class model*. Penentuan tampilan akan dilakukan oleh *class-class controller* dan tampilan antarmuka dengan pengguna akhir akan diatur oleh *View*. Ilustrasi arsitektur sistem informasi dapat dilihat pada Gambar 4.65.



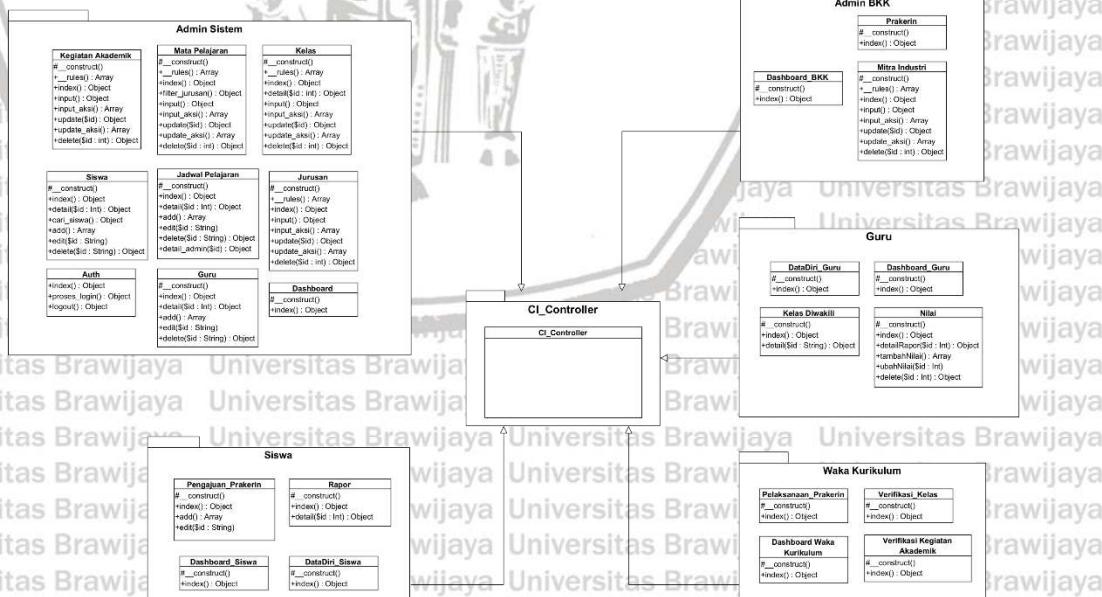
Gambar 4. 65 Arsitektur Sistem SIAKAD SMK MUTU

4.2.5 Class Diagram

Bagian *Class diagram* memodelkan struktur sistem informasi melalui sudut pandang relasi-relasi antar *class*. Pada penelitian ini sistem dibangun menggunakan *framework* Codeigniter. Relasi antar *class* dikelompokkan dalam dua bagian yaitu *Controller* dan *Model*.

4.2.5.1 Class Diagram Bagian Controller

Class yang ada pada bagian *controller* bertugas dalam mengatur halaman yang akan ditampilkan serta data yang mengisi halaman tersebut. *Class diagram* bagian *controller* pada Gambar 4. 66.



Gambar 4. 66 Class Diagram Bagian Controller

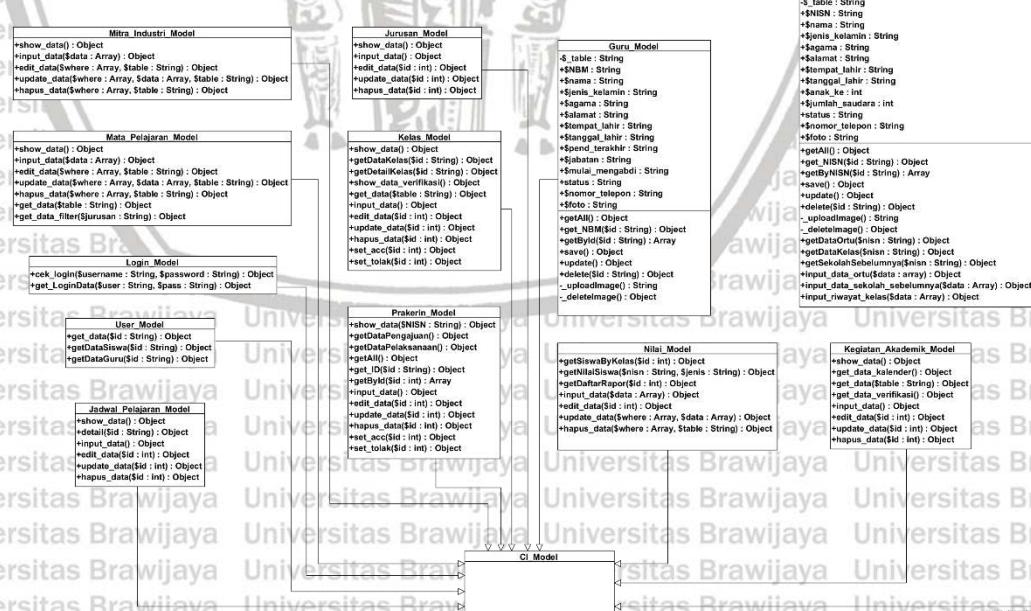
Bagian *controller* terdiri dari beberapa *package* sesuai dengan aktor-aktor yang telah didefinisikan masing-masing *package* terdiri dari beberapa *class*. *Class*

yang terdapat pada *package* admin sistem meliputi *class* kegiatan_akademik, *class* mata_pelajaran, *class* kelas, *class* siswa, *class* jadwal_pelajaran, *class* jurusan, *class* auth, *class* guru, dan *class* dashboard. Pada *package* admin BKK terdapat *class* dashboard, *class* prakerin, dan *class* mitra_industri. Pada *package* guru terdapat *class* dashboard, *class* data_diri, *class* nilai dan *class* kelas_diwakili. Pada *package* waka_kurikulum terdapat *class* dashboard, *class* pelaksanaan_prakerin, *class* verifikasi_kegiatan_akademik, *class* verifikasi_kelas. Pada *package* siswa terdapat *class* dashboard, *class* pengajuan_prakerin, *class* rapor, dan *class* data_diri.

Masing-masing *class* pada bagian *controller* menyediakan setidaknya salah satu dari beberapa operasi pada basis data yang meliputi pembacaan data dari basis data (*READ*), *input data* baru (*CREATE*), Pengubahan data yang ada (*UPDATE*), dan penghapusan data (*DELETE*). Setiap *class* juga memiliki *constructor* untuk melakukan pengecekan data *session* serta inisiasi untuk *model* dan *library* yang dibutuhkan dalam *controller*. Masing-masing *class* pada *package* yang ada merupakan generalisasi dari *class* *CI_Controller* yang merupakan *class* bawaan dari *framework* Codeigniter.

4.2.5.2 Class Diagram Bagian Model

Bagian *Model* memuat operasi data secara langsung dengan *database*, operasi tersebut termasuk namun tidak terbatas pada operasi CRUD (*Create*, *Read*, *Update*, *Delete*). Pada penelitian ini *database* yang digunakan merupakan *database* relasional sehingga terdapat operasi-operasi lebih lanjut terhadap *database* seperti *JOIN*, dan *GROUP*. *Class diagram* bagian *model* pada Gambar 4.



Gambar 4. 67 Class Diagram Bagian Model

Bagian model terdiri atas 12 (dua belas) *class* yang merupakan generalisasi dari *class* CI Model, *class-class* pada bagian *model* meliputi *class* User Model,

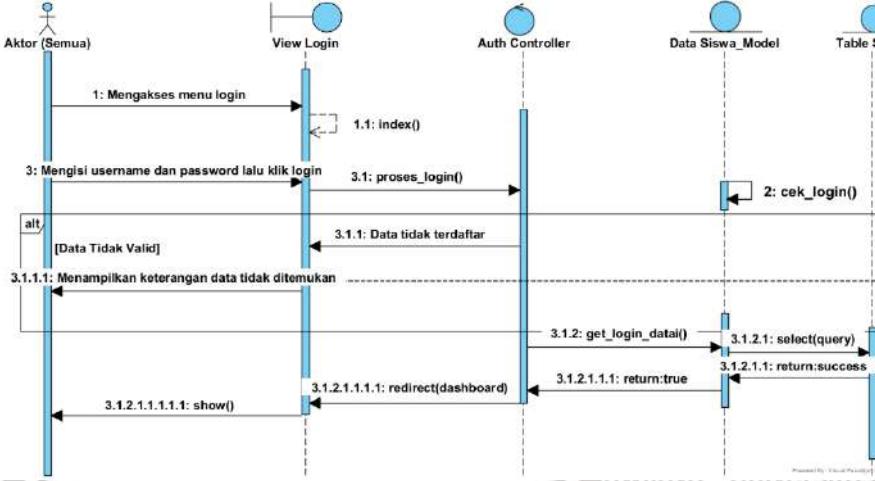
```
class Login_Model, class Jurusan_Model, class Kelas_Model, class Mata_Pelajaran_Model, class Guru_Model, class Siswa_Model, class Jadwal_Pelajaran, class Prakerin_Model, class Kegiatan_Akademik_Model, class Nilai_Model, dan class Mitra_Industri_Model.
```

4.2.6 Sequence Diagram

Sequence Diagram pada tahap ini memodelkan komunikasi yang terjadi antar objek pada siklus hidup sistem. *Sequence diagram* pada tahap ini terdiri dari beberapa objek sesuai dengan arsitektur sistem yang digunakan meliputi *view* yang merepresentasikan tampilan antarmuka, *controller* untuk merepresentasikan *flow* dari sistem, *model* yang merepresentasikan hubungan antara sistem dengan *database* dan *table* yang merupakan tempat penyimpanan data yang dikelola.

1. Sequence Diagram Login Sistem

Sequence diagram untuk melakukan *login* sistem merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas *login* sistem. Pemodelan *sequence diagram* fungsionalitas lihat *login* sistem dapat dilihat pada Gambar 4. 68.



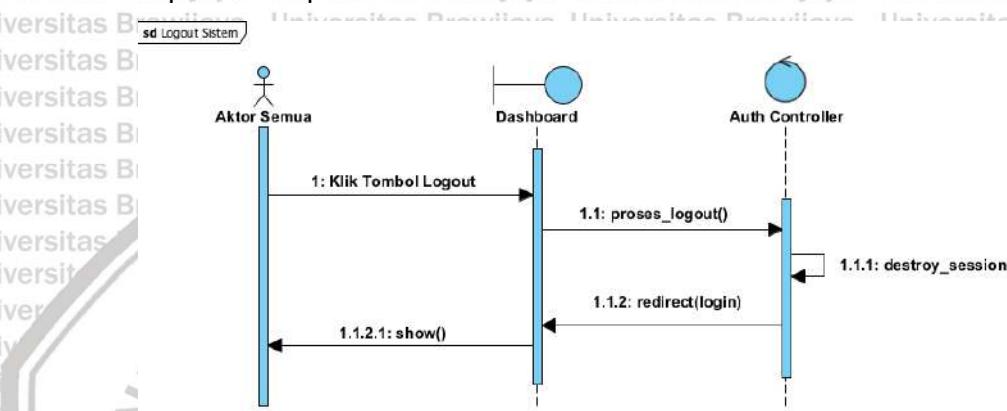
Gambar 4. 68 Sequence Diagram Login Sistem

Proses login dilakukan melalui beberapa bagian sistem yang memiliki *boundary* masing-masing. Bagian *View Login* berfungsi untuk menampilkan antarmuka halaman *login* pada sistem melalui *method* *index()*. Aktor perlu memasukan *username* dan *password* lalu mengklik pada tombol *login*, selanjutnya bagian *View Login* akan memanggil *method* *proses_login()* yang ada pada bagian *Auth Controller* yang selanjutnya akan memanggil *method* *cek_login()* yang melakukan pengecekan data *username* dan *password* yang dimasukkan. Apabila *input* yang dimasukkan oleh aktor tidak valid atau tidak terdaftar maka sistem akan menampilkan keterangan data tidak ditemukan. Apabila *input* yang dimasukkan terdaftar maka *method* *get_login_data()* akan dipanggil. Fungsi dari *method*

get_login_data() adalah mengambil data aktor pada *Table Siswa*. Setelah pemanggilan dengan *query Table Siswa* akan mengembalikan respon *success* pada Data Siswa Model yang mengirimkan respon true pada *Auth Controller* yang akan mengarahkan pada halaman *dashboard*.

2. Sequence Diagram Logout Sistem

Sequence diagram untuk melakukan *logout* sistem merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas *logout* sistem. Pemodelan sequence diagram fungsionalitas *logout* sistem dapat dilihat pada Gambar 4. 69.

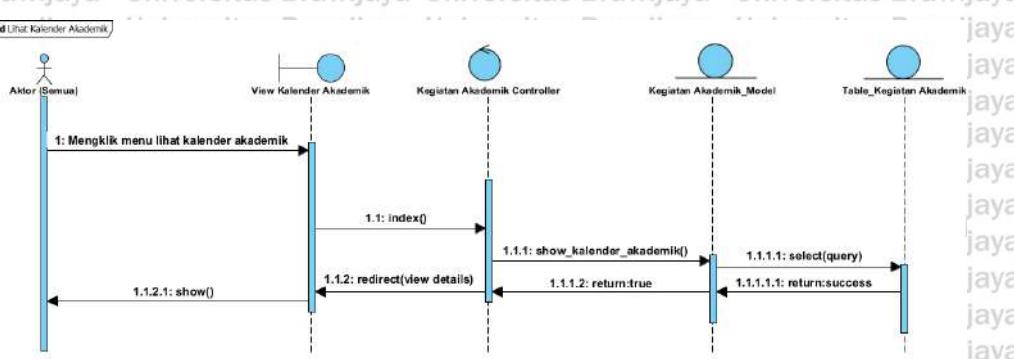


Gambar 4. 69 Sequence Diagram Logout Sistem

Proses *logout* dilakukan melalui beberapa bagian sistem yang memiliki *boundary* masing-masing. Saat aktor mengklik tombol *logout* pada halaman *dashboard* maka *method* *proses_logout()* yang ada pada bagian *Auth Controller*. Bagian *Auth Controller* selanjutnya akan memanggil *method* *destroy_session()* untuk menghapus data *login* aktor. Setelah itu *Auth Controller* akan mengirimkan respons untuk mengarahkan pada halaman *login* dan menampilkan halaman tersebut pada Aktor.

3. Sequence Diagram Lihat Kalender Akademik

Sequence diagram untuk melihat kalender akademik merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas lihat kalender akademik. Pemodelan sequence diagram fungsionalitas lihat kalender akademik dapat dilihat pada Gambar 4. 70.

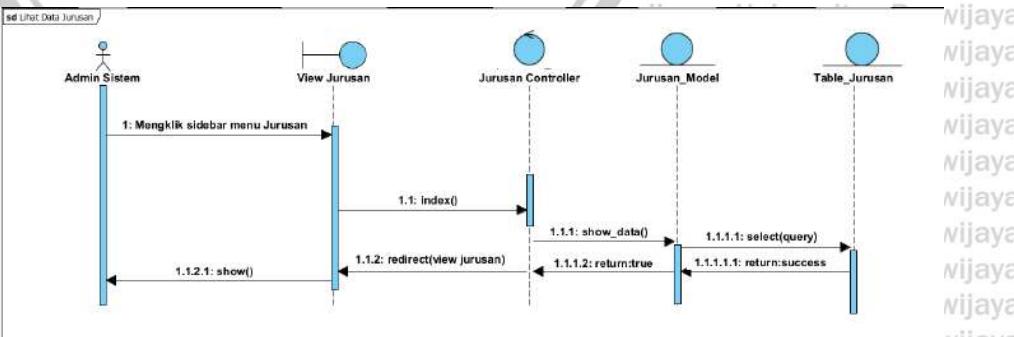


Gambar 4. 70 Sequence Diagram Lihat Kalender Akademik

Proses lihat kalender akademik dimulai dengan aktor melakukan klik terhadap menu kalender akademik, setelah itu bagian *View Kalender Akademik* akan memanggil *method index()* yang ada pada bagian *Kegiatan Akademik Controller*, selanjutnya *method show_kalender_akademik()* yang ada pada *Kegiatan Akademik Model* akan dipanggil oleh *controller*, *method* tersebut akan melakukan *query* pada *Table Kegiatan Akademik* di *database*. Bagian *Table Kegiatan Akademik* akan memberikan respon *success* pada *model* dan selanjutnya akan dikirim status *true* pada *controller* yang selanjutnya akan mengirimkan perintah untuk mengarahkan aktor pada *view detail kalender akademik*.

4. Sequence Diagram Lihat Daftar Jurusan

Sequence diagram untuk melihat daftar jurusan merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas lihat daftar jurusan. Pemodelan *sequence diagram* fungsionalitas lihat daftar jurusan dapat dilihat pada Gambar 4. 71.



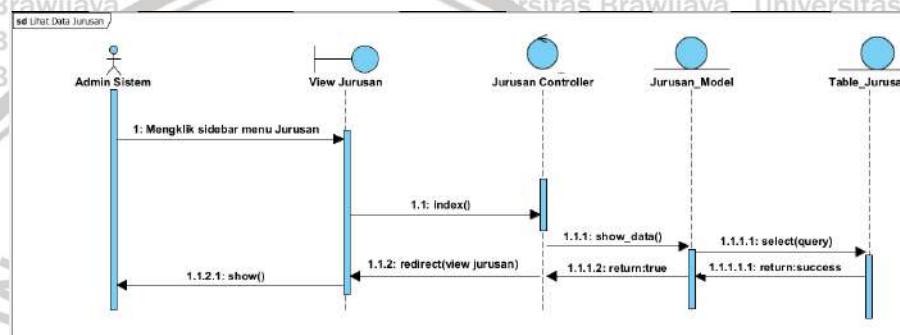
Gambar 4. 71 Sequence Diagram Lihat Daftar Jurusan

Proses lihat daftar jurusan dimulai dengan aktor melakukan klik terhadap menu jurusan, setelah itu bagian *View Jurusan* akan memanggil

method index() yang ada pada bagian *Jurusan Controller*, selanjutnya *method show_data()* yang ada pada *Jurusan Model* akan dipanggil oleh *controller*, *method* tersebut akan melakukan *query* pada *Table Jurusan* di *database*. Bagian *Table Jurusan* akan memberikan respon *success* pada *model* dan selanjutnya akan dikirim status *true* pada *controller* yang selanjutnya akan mengirimkan perintah untuk mengarahkan aktor pada *view jurusan* yang sudah menerima data dari *database*.

5. Sequence Diagram Tambah Data Jurusan

Sequence diagram tambah data jurusan merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas tambah data jurusan. Pemodelan *sequence diagram* fungsionalitas tambah data jurusan dapat dilihat pada Gambar 4. 72.



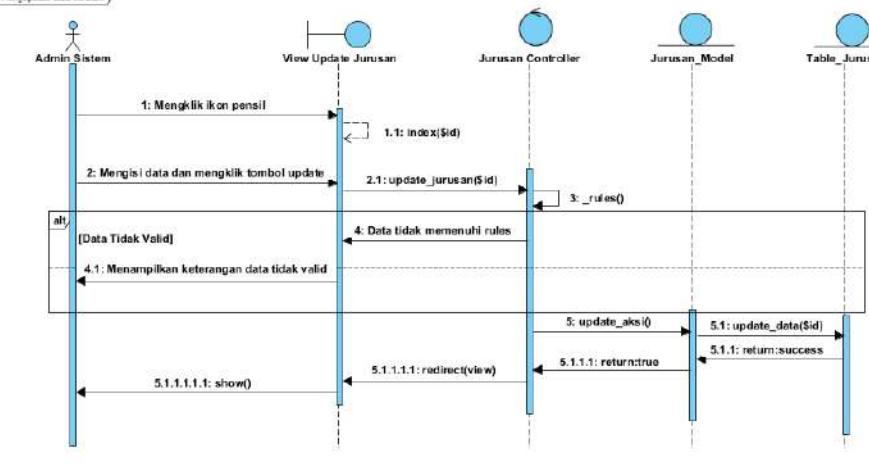
Gambar 4. 72 Sequence Diagram Tambah Data Jurusan

Proses tambah data jurusan dimulai dengan aktor melakukan klik pada tombol tambah jurusan yang ada diatas daftar jurusan. Selanjutnya akan ditampilkan halaman tambah jurusan oleh bagian *View Tambah Jurusan* dengan memanggil *method index()*. Setelah aktor mengisi data yang diminta oleh *field* yang ada dan menekan tombol simpan. Selanjutnya *method input_jurusan()* yang ada pada *controller* akan dipanggil, selanjutnya di bagian *controller* akan melakukan pengecekan apakah data yang dimasukkan oleh aktor valid (terisi semua, tidak ada duplikat), saat data dinyatakan tidak valid maka akan ditampilkan keterangan bahwa data yang dimasukkan tidak valid. Apabila data yang dimasukkan valid maka akan dieksekusi oleh *method input_aksi()* untuk ditambahkan pada tabel jurusan yang ada di *database*. Hasil *query* yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman daftar jurusan.

6. Sequence Diagram Ubah Data Jurusan

Sequence diagram ubah data jurusan merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas ubah data jurusan.

Pemodelan *sequence diagram* fungsionalitas ubah data jurusan dapat dilihat pada Gambar 4. 73.

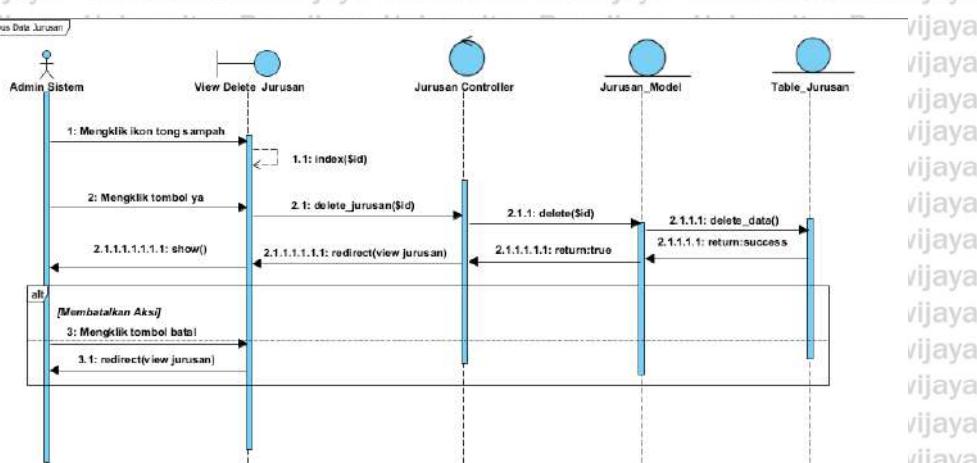


Gambar 4. 73 Sequence Diagram Ubah Data Jurusan

Proses ubah data jurusan dimulai dengan aktor melakukan klik pada ikon pensil yang di daftar jurusan, Selanjutnya akan ditampilkan halaman ubah jurusan oleh bagian *View Update Jurusan* dengan memanggil *method* *index(\$id)*. Setelah aktor mengisi data yang diminta oleh *field* yang ada dan menekan tombol simpan. Selanjutnya *method* *update_jurusan(\$id)* yang ada pada *controller* akan dipanggil, selanjutnya di bagian *controller* akan melakukan pengecekan apakah data yang dimasukkan oleh aktor valid (terisi semua, tidak ada duplikat), saat data dinyatakan tidak valid maka akan ditampilkan keterangan bahwa data yang dimasukkan tidak valid. Apabila data yang dimasukkan valid maka akan dieksekusi oleh *method* *update_aksi()* untuk dilakukan perubahan data sesuai dengan *id* pada tabel jurusan yang ada di *database*. Hasil *query* yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman daftar jurusan.

7. Sequence Diagram Hapus Data Jurusan

Sequence diagram hapus data jurusan merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas hapus data jurusan. Pemodelan *sequence diagram* fungsionalitas hapus data jurusan dapat dilihat pada Gambar 4. 74.

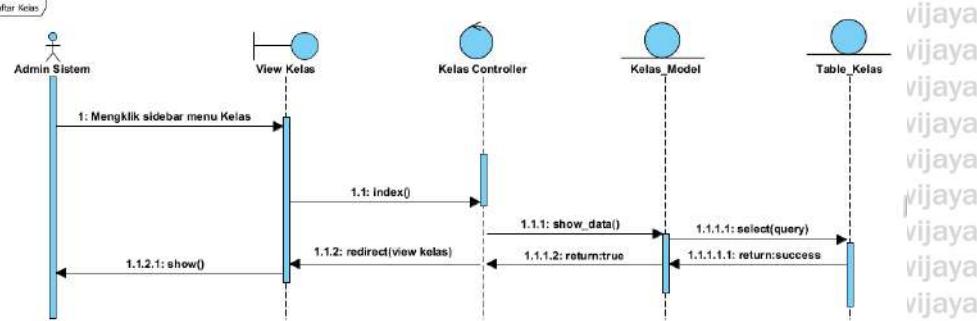


Gambar 4. 74 Sequence Diagram Hapus Data Jurusan

Proses hapus data jurusan dimulai dengan aktor melakukan klik pada ikon tong sampah yang ada pada daftar jurusan. Selanjutnya akan ditampilkan konfirmasi penghapusan data oleh bagian *View Delete Jurusan* dengan memanggil *method* *index(\$id)*. Apabila aktor mengklik tombol “ya” maka *method* *delete_jurusan(\$id)* yang ada pada *controller* akan dipanggil, selanjutnya akan diproses oleh *method* *delete(\$id)* yang terdapat di bagian *model* untuk selanjutnya melakukan *query* pada tabel *Jurusan* di *database*. Hasil *query* yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman daftar jurusan.

8. Sequence Diagram Lihat Daftar Kelas

Sequence diagram lihat daftar kelas merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas lihat daftar kelas. Pemodelan *sequence diagram* fungsionalitas lihat daftar kelas dapat dilihat pada Gambar 4. 75.

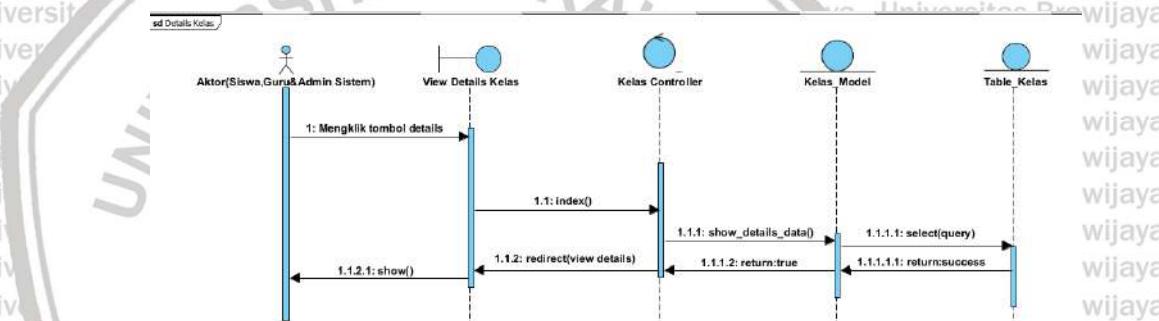


Gambar 4. 75 Sequence Diagram Lihat Daftar Kelas

Proses lihat daftar kelas dimulai dengan aktor melakukan klik terhadap menu kelas, setelah itu bagian *View Kelas* akan memanggil *method index()* yang ada pada bagian *Kelas Controller*, selanjutnya *method show_data()* yang ada pada *Kelas Model* akan dipanggil oleh *controller*, *method* tersebut akan melakukan *query* pada *Table Kelas* di *database*. Bagian *Table Kelas* akan memberikan respon *success* pada *model* dan selanjutnya akan dikirim status *true* pada *controller* yang selanjutnya akan mengirimkan perintah untuk mengarahkan aktor pada *view kelas* yang sudah menerima data dari *database*.

9. Sequence Diagram Lihat Detail Kelas

Sequence diagram lihat detail kelas merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas lihat detail kelas. Pemodelan *sequence diagram* fungsionalitas lihat detail kelas dapat dilihat pada Gambar 4. 76.



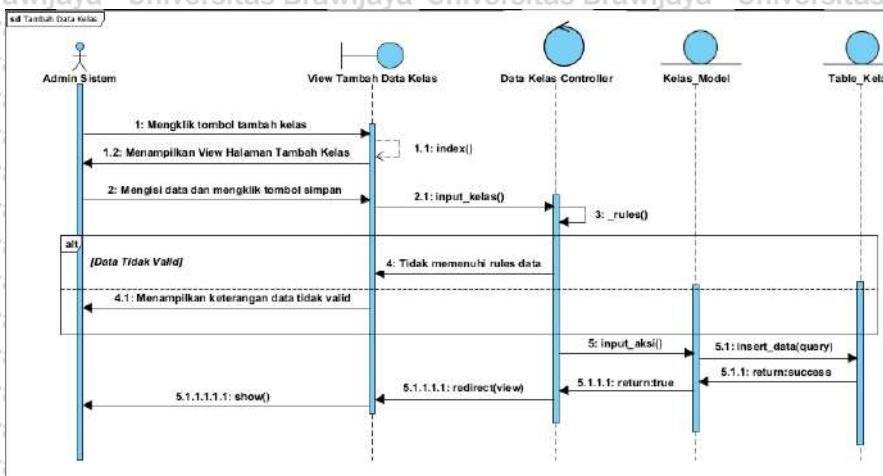
Gambar 4. 76 Sequence Diagram Lihat Detail Kelas

Proses lihat detail kelas dimulai dengan aktor melakukan klik terhadap tombol details pada daftar kelas, setelah itu bagian *View Details Kelas* akan memanggil *method index()* yang ada pada bagian *Kelas Controller*, selanjutnya *method show_details_data()* yang ada pada *Kelas Model* akan dipanggil oleh *controller*, *method* tersebut akan melakukan *query* pada *Table Kelas* di *database*. Bagian *Table Kelas* akan memberikan respon *success* pada *model* dan selanjutnya akan dikirim status *true* pada *controller* yang selanjutnya akan mengirimkan perintah untuk mengarahkan aktor pada *view details kelas* yang sudah menerima data dari *database*.

10. Sequence Diagram Tambah Data Kelas

Sequence diagram tambah data kelas merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas tambah data kelas.

Pemodelan *sequence diagram* fungsionalitas tambah data kelas dapat dilihat pada Gambar 4.77.

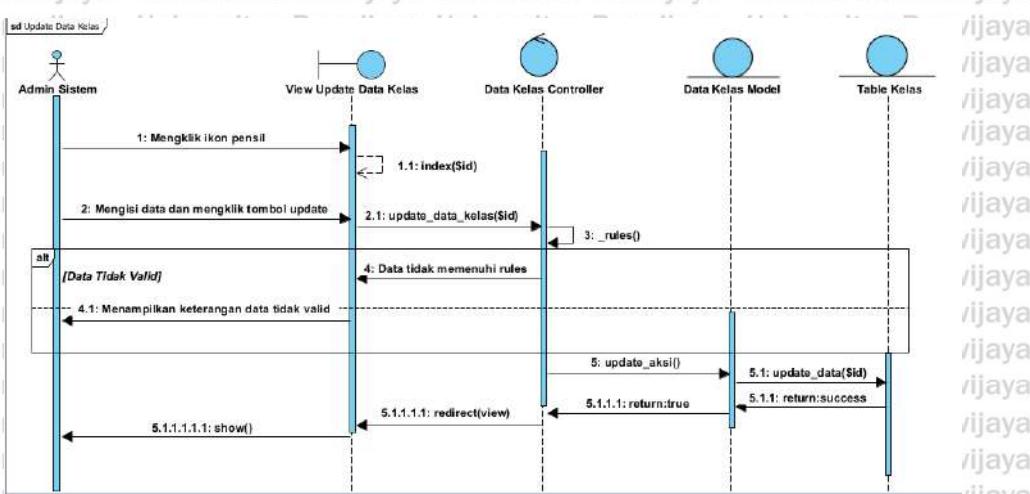


Gambar 4.77 Sequence Diagram Tambah Data Kelas

Proses tambah data kelas dimulai dengan aktor melakukan klik pada tombol tambah kelas yang ada diatas daftar kelas, Selanjutnya akan ditampilkan halaman tambah kelas oleh bagian *View Tambah Data Kelas* dengan memanggil *method* *index()*. Setelah aktor mengisi data yang diminta oleh *field* yang ada dan menekan tombol simpan. Selanjutnya *method* *input_kelas()* yang ada pada *controller* akan dipanggil, selanjutnya di bagian *controller* akan melakukan pengecekan apakah data yang dimasukkan oleh aktor valid (terisi semua, tidak ada duplikat), saat data dinyatakan tidak valid maka akan ditampilkan keterangan bahwa data yang dimasukkan tidak valid. Apabila data yang dimasukkan valid maka akan dieksekusi oleh *method* *input_aksi()* untuk ditambahkan pada tabel kelas yang ada di *database*. Hasil *query* yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman daftar kelas.

11. Sequence Diagram Ubah Data Kelas

Sequence diagram ubah data kelas merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas ubah data kelas. Pemodelan *sequence diagram* fungsionalitas ubah data kelas dapat dilihat pada Gambar 4.78.

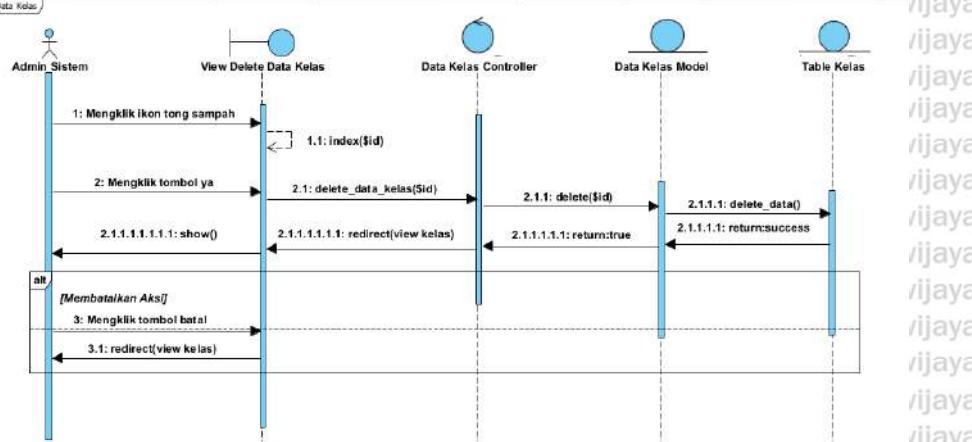


Gambar 4. 78 Sequence Diagram Ubah Data Kelas

Proses ubah data kelas dimulai dengan aktor melakukan klik pada ikon pensil yang di daftar kelas, Selanjutnya akan ditampilkan halaman ubah kelas oleh bagian *View Update Data Kelas* dengan memanggil *method index(\$id)*. Setelah aktor mengisi data yang diminta oleh *field* yang ada dan menekan tombol simpan. Selanjutnya *method update_data_kelas(\$id)* yang ada pada *controller* akan dipanggil, selanjutnya di bagian *controller* akan melakukan pengecekan apakah data yang dimasukkan oleh aktor valid (terisi semua, tidak ada duplikat), saat data dinyatakan tidak valid maka akan ditampilkan keterangan bahwa data yang dimasukkan tidak valid. Apabila data yang dimasukkan valid maka akan dieksekusi oleh *method update_aksi()* untuk dilakukan perubahan data sesuai dengan *id* pada tabel kelas yang ada di *database*. Hasil *query* yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman daftar kelas.

12. Sequence Diagram Hapus Data Kelas

Sequence diagram hapus data kelas merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkaitan dengan fungsionalitas hapus data kelas. Pemodelan sequence diagram fungsionalitas hapus data kelas dapat dilihat pada Gambar 4. 79.

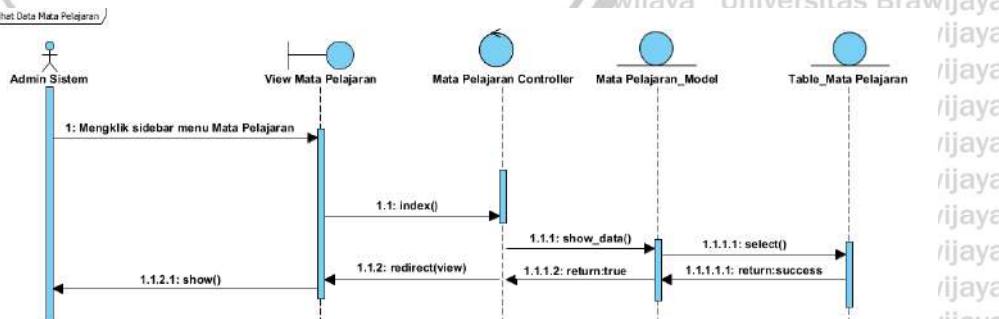


Gambar 4. 79 Sequence Diagram Hapus Data Kelas

Proses hapus data kelas dimulai dengan aktor melakukan klik pada ikon tong sampah yang ada pada daftar kelas. Selanjutnya akan ditampilkan konfirmasi penghapusan data oleh bagian *View Delete Data Kelas* dengan memanggil *method* *index(\$id)*. Apabila aktor mengklik tombol “ya” maka *method* *delete_data_kelas(\$id)* yang ada pada *controller* akan dipanggil, selanjutnya akan diproses oleh *method* *delete(\$id)* yang terdapat di bagian *model* untuk selanjutnya melakukan *query* pada tabel *Kelas* di *database*. Hasil *query* yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman daftar kelas.

13. Sequence Diagram Lihat Daftar Mata Pelajaran

Sequence diagram lihat daftar mata pelajaran merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian sistem informasi yang berkaitan dengan fungsionalitas lihat daftar mata pelajaran. Pemodelan *sequence diagram* fungsionalitas lihat daftar mata pelajaran dapat dilihat pada Gambar 4. 80.

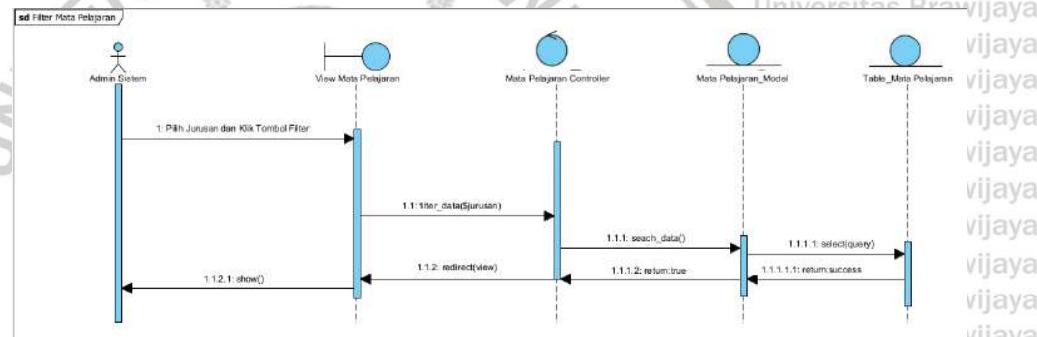


Gambar 4. 80 Sequence Diagram Lihat Daftar Mata Pelajaran

Proses lihat daftar mata pelajaran dimulai dengan aktor melakukan klik terhadap menu mata pelajaran, setelah itu bagian *View Mata Pelajaran* akan memanggil *method index()* yang ada pada bagian *Mata Pelajaran Controller*, selanjutnya *method show_data()* yang ada pada *Mata Pelajaran Model* akan dipanggil oleh *controller*, *method* tersebut akan melakukan *query* pada *Table Mata Pelajaran* di *database*. Bagian *Table Mata Pelajaran* akan memberikan respon *success* pada *model* dan selanjutnya akan dikirim status *true* pada *controller* yang selanjutnya akan mengirimkan perintah untuk mengarahkan aktor pada *view mata pelajaran* yang sudah menerima data dari *database*.

14. Sequence Diagram Filter Daftar Mata Pelajaran

Sequence diagram filter daftar mata pelajaran merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas *filter* daftar mata pelajaran. Pemodelan *sequence diagram* fungsionalitas *filter* daftar mata pelajaran dapat dilihat pada Gambar 4. 81.



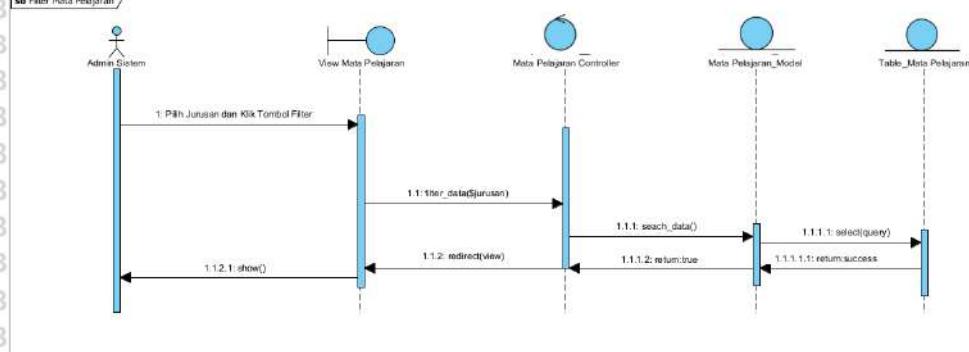
Gambar 4. 81 Sequence Diagram Filter Daftar Mata Pelajaran

Proses filter daftar mata pelajaran dimulai dengan aktor memilih jurusan dan melakukan klik tombol filter, setelah itu bagian *View Mata Pelajaran* akan memanggil *method filter_data(\$jurusan)* yang ada pada bagian *Mata Pelajaran Controller*, selanjutnya *method search_data()* yang ada pada *Mata Pelajaran Model* akan dipanggil oleh *controller*, *method* tersebut akan melakukan *query* pada *Table Mata Pelajaran* di *database*. Bagian *Table Mata Pelajaran* akan memberikan respon *success* pada *model* dan selanjutnya akan dikirim status *true* pada *controller* yang selanjutnya akan mengirimkan perintah untuk mengarahkan aktor pada *view mata pelajaran* yang sudah menerima data sesuai jurusan dari *database*.

15. Sequence Diagram Tambah Data Mata Pelajaran

Sequence diagram tambah data mata pelajaran merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas tambah

data mata pelajaran. Pemodelan *sequence diagram* fungsionalitas tambah data mata pelajaran dapat dilihat pada Gambar 4. 82.

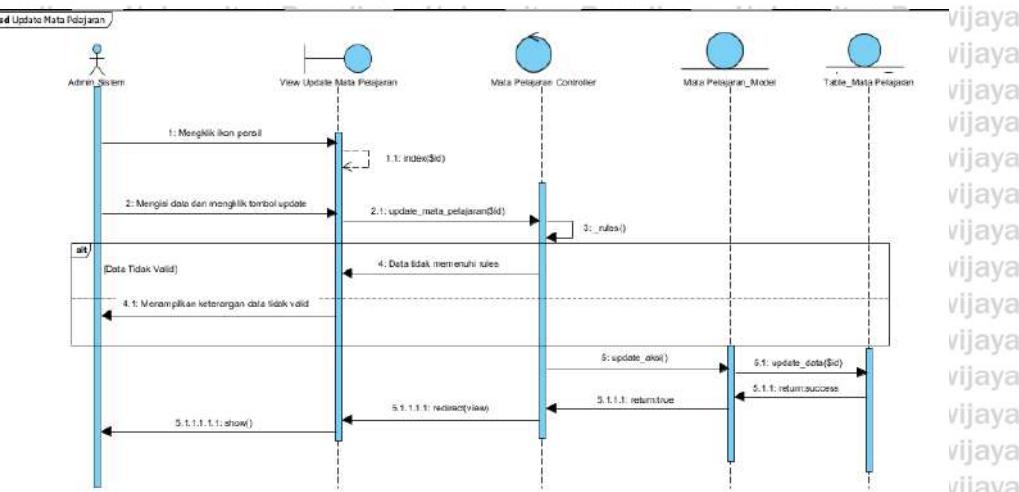


Gambar 4. 82 Sequence Diagram Tambah Data Mata Pelajaran

Proses tambah mata pelajaran dimulai dengan aktor melakukan klik pada tombol tambah mata pelajaran yang ada diatas daftar mata pelajaran, Selanjutnya akan ditampilkan halaman tambah kelas oleh bagian *View Tambah Mata Pelajaran* dengan memanggil *method index()*. Setelah aktor mengisi data yang diminta oleh *field* yang ada dan menekan tombol simpan. Selanjutnya *method input_mata_pelajaran()* yang ada pada *controller* akan dipanggil, selanjutnya di bagian *controller* akan melakukan pengecekan apakah data yang dimasukkan oleh aktor valid (terisi semua, tidak ada duplikat), saat data dinyatakan tidak valid maka akan ditampilkan keterangan bahwa data yang dimasukkan tidak valid. Apabila data yang dimasukkan valid maka akan dieksekusi oleh *method input_aksi()* untuk ditambahkan pada tabel mata pelajaran yang ada di *database*. Hasil *query* yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman daftar mata pelajaran.

16. Sequence Diagram Ubah Data Mata Pelajaran

Sequence diagram ubah data mata pelajaran merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkaitan dengan fungsionalitas ubah data mata pelajaran. Pemodelan *sequence diagram* fungsionalitas ubah data mata pelajaran dapat dilihat pada Gambar 4. 83.

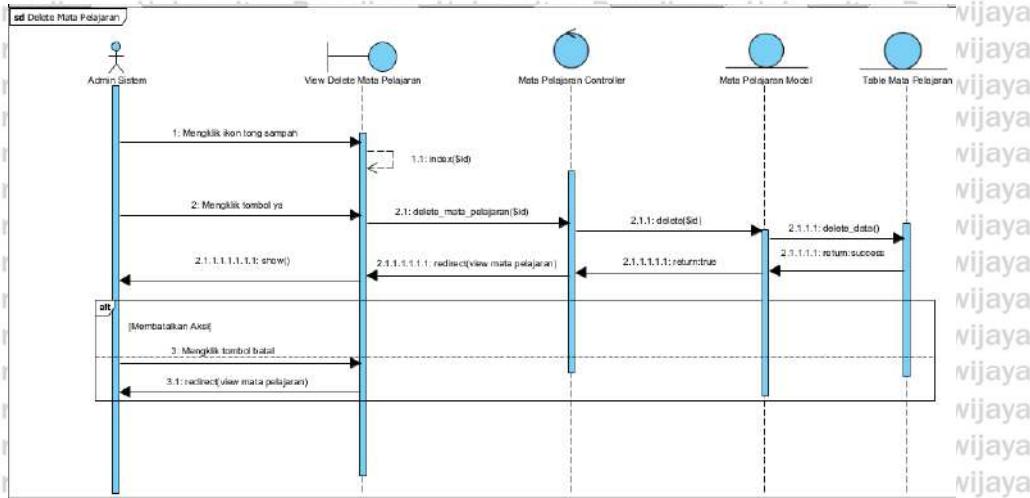


Gambar 4. 83 Sequence Diagram Ubah Data Mata Pelajaran

Proses ubah data mata pelajaran dimulai dengan aktor melakukan klik pada ikon pensil yang di daftar mata pelajaran, Selanjutnya akan ditampilkan halaman ubah mata pelajaran oleh bagian *View Update Mata Pelajaran* dengan memanggil *method index(\$id)*. Setelah aktor mengisi data yang diminta oleh *field* yang ada dan menekan tombol simpan. Selanjutnya *method update_mata_pelajaran(\$id)* yang ada pada *controller* akan dipanggil, selanjutnya di bagian *controller* akan melakukan pengecekan apakah data yang dimasukkan oleh aktor valid (terisi semua, tidak ada duplikat), saat data dinyatakan tidak valid maka akan ditampilkan keterangan bahwa data yang dimasukkan tidak valid. Apabila data yang dimasukkan valid maka akan dieksekusi oleh *method update_aksi()* untuk dilakukan perubahan data sesuai dengan *id* pada tabel mata pelajaran yang ada di *database*. Hasil *query* yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman daftar mata pelajaran.

17. Sequence Diagram Hapus Data Mata Pelajaran

Sequence diagram hapus data mata pelajaran merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas hapus data mata pelajaran. Pemodelan *sequence diagram* fungsionalitas hapus data mata pelajaran dapat dilihat pada Gambar 4. 84.

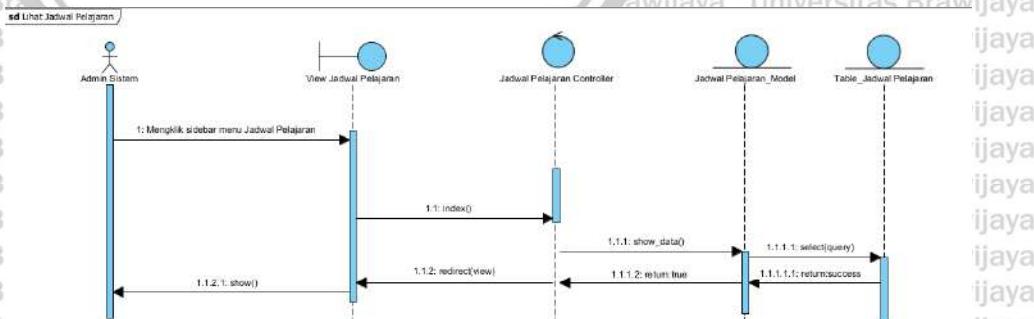


Gambar 4. 84 Sequence Diagram Hapus Data Mata Pelajaran

Proses hapus data mata pelajaran dimulai dengan aktor melakukan klik pada ikon tong sampah yang ada pada daftar mata pelajaran. Selanjutnya akan ditampilkan konfirmasi penghapusan data oleh bagian *View Delete Mata Pelajaran* dengan memanggil *method index(\$id)*. Apabila aktor mengklik tombol “ya” maka *method delete_mata_pelajaran(\$id)* yang ada pada *controller* akan dipanggil, selanjutnya akan diproses oleh *method delete(\$id)* yang terdapat di bagian *model* untuk selanjutnya melakukan *query* pada tabel Mata Pelajaran di *database*. Hasil *query* yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman mata pelajaran.

18. Sequence Diagram Lihat Daftar Jadwal Pelajaran

Sequence diagram lihat daftar jadwal pelajaran merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian sistem informasi yang berkenaan dengan fungsionalitas lihat daftar jadwal pelajaran. Pemodelan *sequence diagram* fungsionalitas lihat daftar jadwal pelajaran dapat dilihat pada Gambar 4. 85



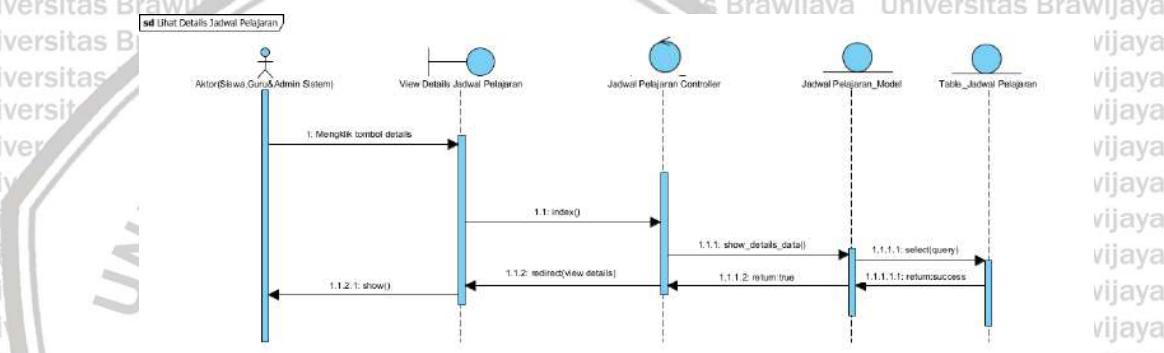
Gambar 4. 85 Sequence Diagram Lihat Daftar Jadwal Pelajaran

Proses lihat daftar jadwal pelajaran dimulai dengan aktor melakukan klik terhadap menu jadwal pelajaran, setelah itu bagian *View* melakukannya.

Kelas akan memanggil *method index()* yang ada pada bagian *Jadwal Pelajaran Controller*, selanjutnya *method show_data()* yang ada pada *Jadwal Pelajaran Model* akan dipanggil oleh *controller*, *method tersebut* akan melakukan *query* pada *Table Jadwal Pelajaran* di *database*. Bagian *Table Jadwal Pelajaran* akan memberikan respon *success* pada *model* dan selanjutnya akan dikirim status *true* pada *controller* yang selanjutnya akan mengirimkan perintah untuk mengarahkan aktor pada *view jadwal pelajaran* yang sudah menerima data dari *database*.

19. Sequence Diagram Detail Jadwal Pelajaran

Sequence diagram lihat detail jadwal pelajaran merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian sistem informasi yang berkenaan dengan fungsionalitas lihat detail jadwal pelajaran. Pemodelan *sequence diagram* fungsionalitas lihat detail jadwal pelajaran dapat dilihat pada Gambar 4. 86.

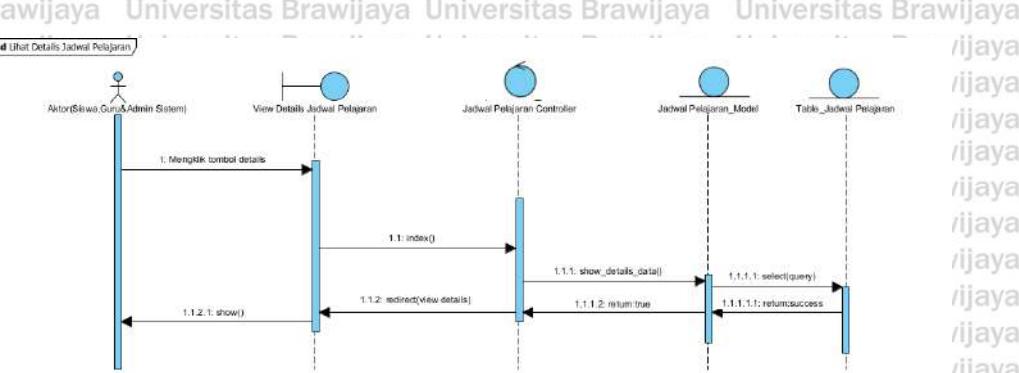


Gambar 4. 86 Sequence Diagram Lihat Detail Jadwal Pelajaran

Proses detail jadwal pelajaran dimulai dengan aktor melakukan klik terhadap menu detail pada daftar jadwal pelajaran, setelah itu bagian *View Details Jadwal Pelajaran* akan memanggil *method index()* yang ada pada bagian *Jadwal Pelajaran Controller*, selanjutnya *method show_details_data()* yang ada pada *Jadwal Pelajaran Model* akan dipanggil oleh *controller*, *method tersebut* akan melakukan *query* pada *Table Jadwal Pelajaran* di *database*. Bagian *Table Jadwal Pelajaran* akan memberikan respon *success* pada *model* dan selanjutnya akan dikirim status *true* pada *controller* yang selanjutnya akan mengirimkan perintah untuk mengarahkan aktor pada *view details jadwal pelajaran* yang sudah menerima data dari *database*.

20. Sequence Diagram Tambah Jadwal Pelajaran

Sequence diagram tambah jadwal pelajaran merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian sistem informasi yang berkenaan dengan fungsionalitas tambah jadwal pelajaran. Pemodelan *sequence diagram* fungsionalitas tambah jadwal pelajaran dapat dilihat pada Gambar 4.87.

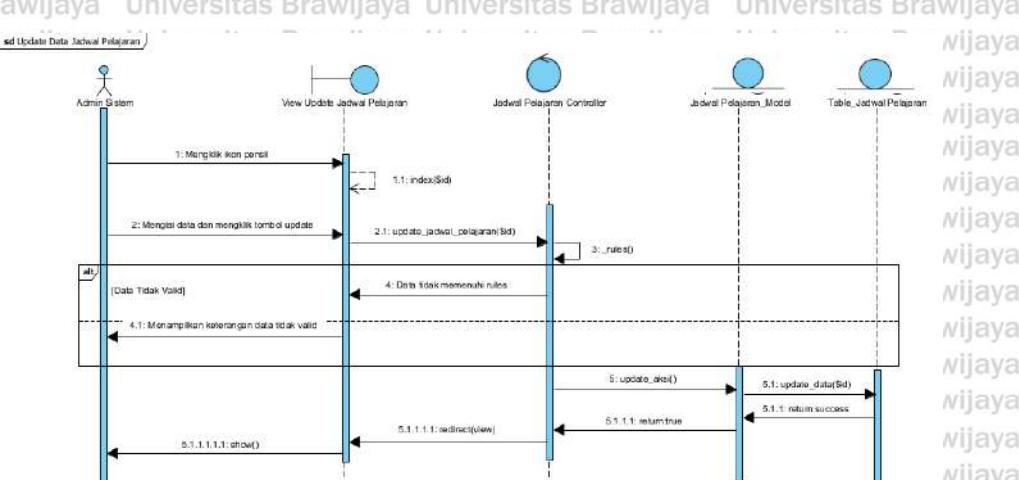


Gambar 4. 87 Sequence Diagram Tambah Jadwal Pelajaran

Proses tambah jadwal pelajaran dimulai dengan aktor melakukan klik pada tombol tambah mata pelajaran yang ada diatas daftar mata pelajaran, Selanjutnya akan ditampilkan halaman tambah kelas oleh bagian *View Tambah Jadwal Pelajaran* dengan memanggil *method index()*. Setelah aktor mengisi data yang diminta oleh *field* yang ada dan menekan tombol simpan. Selanjutnya *method input_jadwal_pelajaran()* yang ada pada *controller* akan dipanggil, selanjutnya di bagian *controller* akan melakukan pengecekan apakah data yang dimasukkan oleh aktor valid (terisi semua, tidak ada duplikat), saat data dinyatakan tidak valid maka akan ditampilkan keterangan bahwa data yang dimasukkan tidak valid. Apabila data yang dimasukkan valid maka akan dieksekusi oleh *method input_aksi()* untuk ditambahkan pada tabel jadwal pelajaran yang ada di *database*. Hasil *query* yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman daftar jadwal pelajaran.

21. Sequence Diagram Ubah Jadwal Pelajaran

Sequence diagram ubah jadwal pelajaran merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas ubah jadwal pelajaran. Pemodelan *sequence diagram* fungsionalitas ubah jadwal pelajaran dapat dilihat pada Gambar 4. 88.

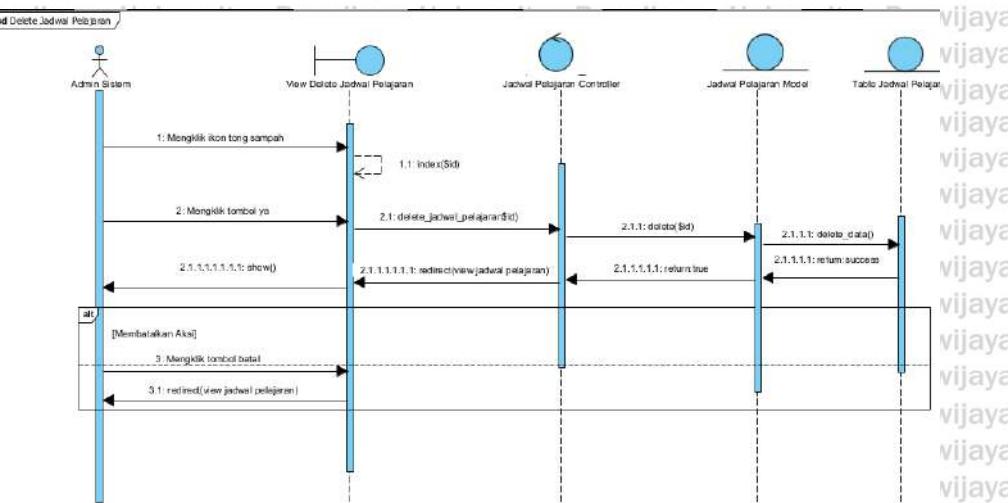


Gambar 4. 88 Sequence Diagram Ubah Jadwal Pelajaran

Proses ubah data jadwal pelajaran dimulai dengan aktor melakukan klik pada ikon pensil yang di daftar jadwal pelajaran. Selanjutnya akan ditampilkan halaman ubah mata pelajaran oleh bagian *View Update Jadwal Pelajaran* dengan memanggil *method index(\$id)*. Setelah aktor mengisi data yang diminta oleh *field* yang ada dan menekan tombol simpan. Selanjutnya *method update_jadwal_pelajaran(\$id)* yang ada pada *controller* akan dipanggil, selanjutnya di bagian *controller* akan melakukan pengecekan apakah data yang dimasukkan oleh aktor valid (terisi semua, tidak ada duplikat), saat data dinyatakan tidak valid maka akan ditampilkan keterangan bahwa data yang dimasukkan tidak valid. Apabila data yang dimasukkan valid maka akan dieksekusi oleh *method update_aksi()* untuk dilakukan perubahan data sesuai dengan *id* pada tabel jadwal pelajaran yang ada di *database*. Hasil *query* yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman daftar jadwal pelajaran.

22. Sequence Diagram Hapus Jadwal Pelajaran

Sequence diagram hapus jadwal pelajaran merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas hapus jadwal pelajaran. Pemodelan sequence diagram fungsionalitas hapus jadwal pelajaran dapat dilihat pada Gambar 4. 89.

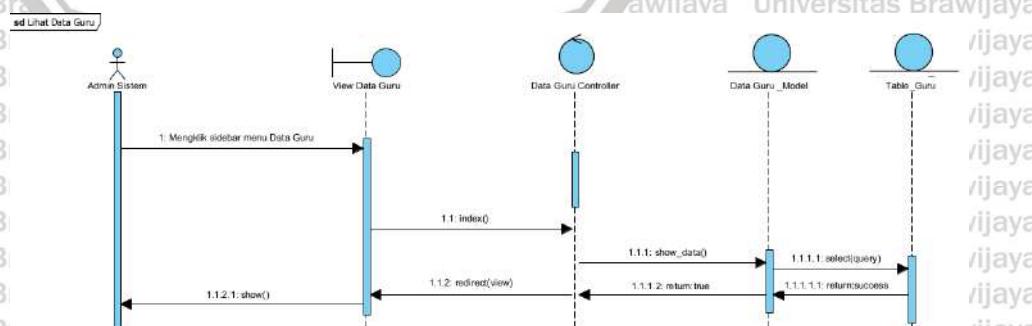


Gambar 4. 89 Sequence Diagram Hapus Jadwal Pelajaran

Proses hapus data jadwal pelajaran dimulai dengan aktor melakukan klik pada ikon tong sampah yang ada pada daftar jadwal pelajaran, Selanjutnya akan ditampilkan konfirmasi penghapusan data oleh bagian *View Delete Jadwal Pelajaran* dengan memanggil *method index(\$id)*. Apabila aktor mengklik tombol “ya” maka *method delete_jadwal_pelajaran(\$id)* yang ada pada *controller* akan dipanggil, selanjutnya akan diproses oleh *method delete(\$id)* yang terdapat di bagian *model* untuk selanjutnya melakukan *query* pada tabel Jadwal Pelajaran di *database*. Hasil *query* yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman jadwal pelajaran.

23. Sequence Diagram Lihat Daftar Guru

Sequence diagram lihat daftar guru merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas lihat daftar guru. Pemodelan sequence diagram fungsionalitas lihat daftar guru dapat dilihat pada Gambar 4. 90.



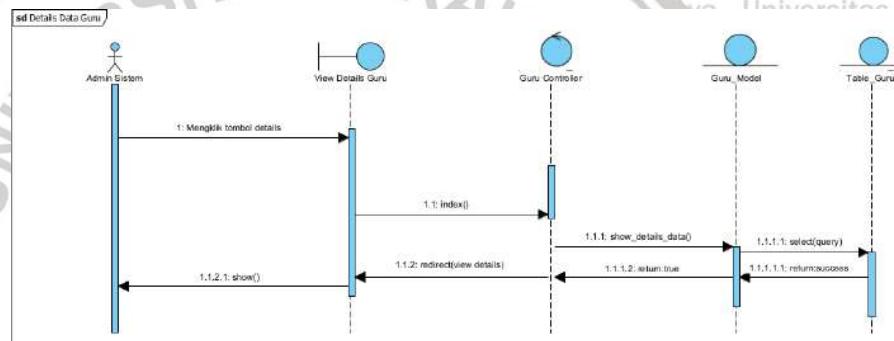


Gambar 4. 90 Sequence Diagram Lihat Daftar Guru

Proses lihat daftar guru dimulai dengan aktor melakukan klik terhadap menu guru, setelah itu bagian *View Guru* akan memanggil *method index()* yang ada pada bagian *Guru Controller*, selanjutnya *method show_data()* yang ada pada *Guru Model* akan dipanggil oleh *controller*, *method* tersebut akan melakukan *query* pada *Table Guru* di *database*. Bagian *Table Guru* akan memberikan respon *success* pada *model* dan selanjutnya akan dikirim status *true* pada *controller* yang selanjutnya akan mengirimkan perintah untuk mengarahkan aktor pada *view guru* yang sudah menerima data dari *database*.

24. Sequence Diagram Detail Data Guru

Sequence diagram detail data guru merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas detail data guru. Pemodelan *sequence diagram* fungsionalitas detail data guru dapat dilihat pada Gambar 4. 91.

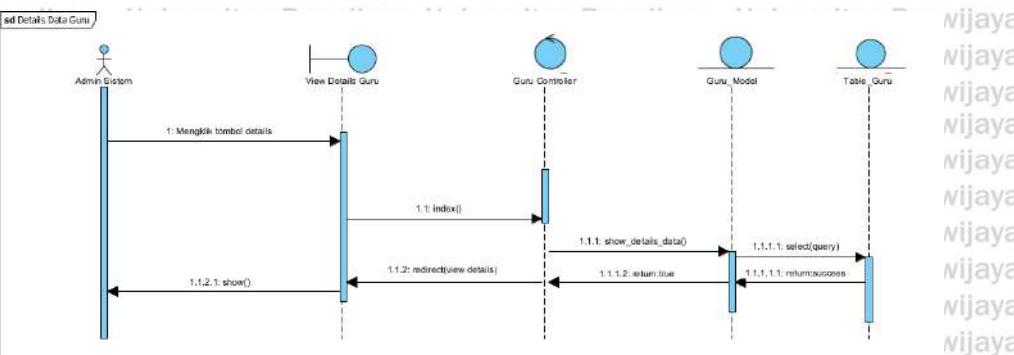


Gambar 4. 91 Sequence Diagram Detail Data Guru

Proses lihat details guru dimulai dengan aktor melakukan klik terhadap tombol details yang ada pada daftar guru, setelah itu bagian *View Detail Guru* akan memanggil *method index()* yang ada pada bagian *Guru Controller*, selanjutnya *method show_data()* yang ada pada *Guru Model* akan dipanggil oleh *controller*, *method* tersebut akan melakukan *query* pada *Table Guru* di *database*. Bagian *Table Guru* akan memberikan respon *success* pada *model* dan selanjutnya akan dikirim status *true* pada *controller* yang selanjutnya akan mengirimkan perintah untuk mengarahkan aktor pada *view details guru* yang sudah menerima data dari *database*.

25. Sequence Diagram Tambah Data Guru

Sequence diagram tambah data guru merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas tambah data guru. Pemodelan *sequence diagram* fungsionalitas tambah data guru dapat dilihat pada Gambar 4. 92.

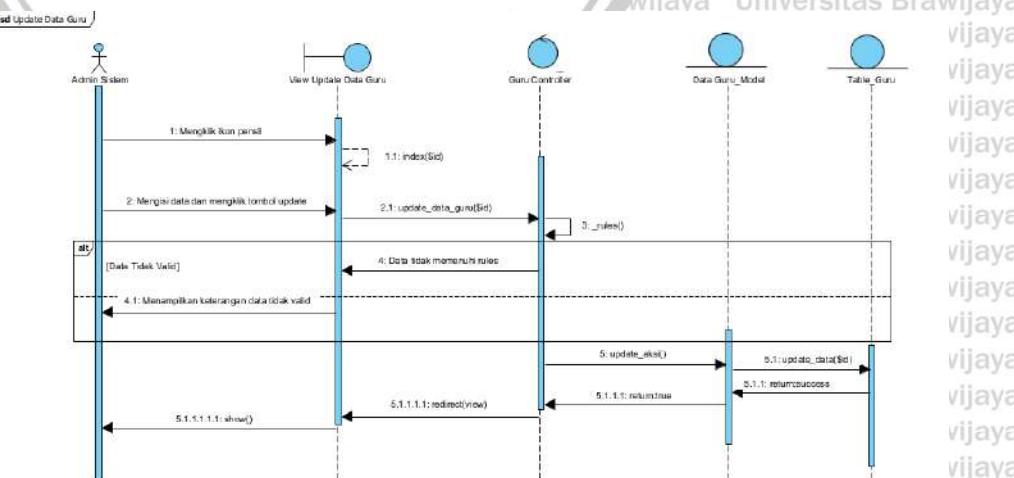


Gambar 4. 92 Sequence Diagram Tambah Data Guru

Proses tambah data guru dimulai dengan aktor melakukan klik pada tombol tambah data guru yang ada diatas daftar guru, Selanjutnya akan ditampilkan halaman tambah guru oleh bagian *View Tambah Guru* dengan memanggil *method index()*. Setelah aktor mengisi data yang diminta oleh *field* yang ada dan menekan tombol simpan. Selanjutnya *method input_guru()* yang ada pada *controller* akan dipanggil, selanjutnya di bagian *controller* akan melakukan pengecekan apakah data yang dimasukkan oleh aktor valid (terisi semua, tidak ada duplikat), saat data dinyatakan tidak valid maka akan ditampilkan keterangan bahwa data yang dimasukkan tidak valid. Apabila data yang dimasukkan valid maka akan dieksekusi oleh *method input_aksi()* untuk ditambahkan pada tabel guru yang ada di *database*. Hasil *query* yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman daftar guru.

26. Sequence Diagram Ubah Data Guru

Sequence diagram ubah data guru merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas ubah data guru. Pemodelan *sequence diagram* fungsionalitas ubah data guru dapat dilihat pada Gambar 4. 93.

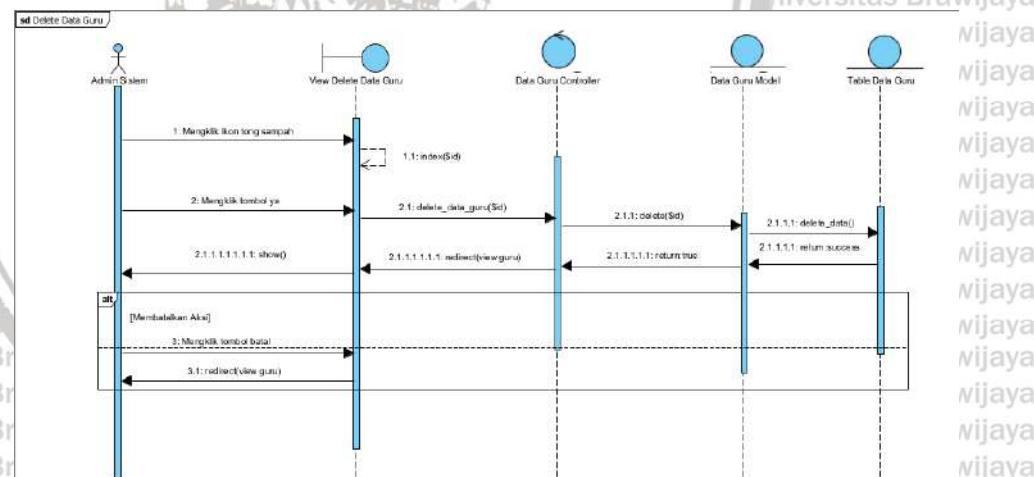


Gambar 4. 93 Sequence Diagram Ubah Data Guru

Proses ubah data guru dimulai dengan aktor melakukan klik pada ikon pensil yang di daftar guru, Selanjutnya akan ditampilkan halaman ubah data guru oleh bagian *View Update Data Guru* dengan memanggil *method index(\$id)*. Setelah aktor mengisi data yang diminta oleh *field* yang ada dan menekan tombol simpan. Selanjutnya *method update_data_guru(\$id)* yang ada pada *controller* akan dipanggil, selanjutnya di bagian *controller* akan melakukan pengecekan apakah data yang dimasukkan oleh aktor valid (terisi semua, tidak ada duplikat), saat data dinyatakan tidak valid maka akan ditampilkan keterangan bahwa data yang dimasukkan tidak valid. Apabila data yang dimasukkan valid maka akan dieksekusi oleh *method update_aksi()* untuk dilakukan perubahan data sesuai dengan *id* pada tabel guru yang ada di *database*. Hasil *query* yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman daftar guru.

27. Sequence Diagram Hapus Data Guru

Sequence diagram hapus data guru merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkaitan dengan fungsionalitas hapus data guru. Pemodelan *sequence diagram* fungsionalitas hapus data guru dapat dilihat pada Gambar 4. 94.

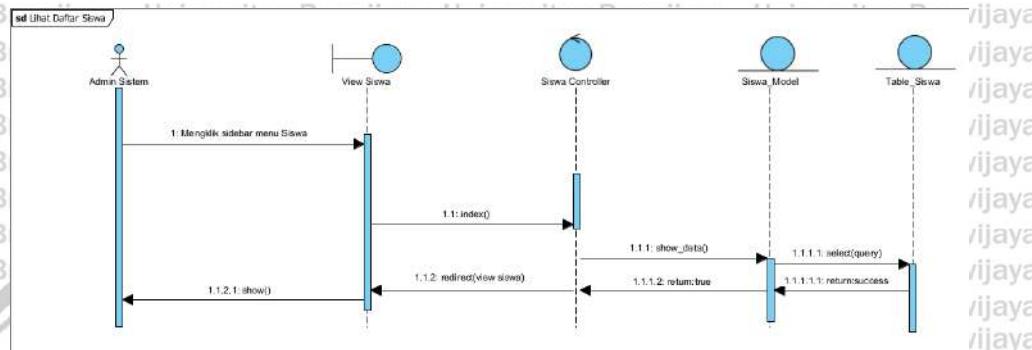


Gambar 4. 94 Sequence Diagram Hapus Data Guru

Proses hapus data guru dimulai dengan aktor melakukan klik pada ikon tong sampah yang ada pada daftar guru, Selanjutnya akan ditampilkan konfirmasi penghapusan data oleh bagian *View Delete Data Guru* dengan memanggil *method index(\$id)*. Apabila aktor mengklik tombol "ya" maka *method delete_data_guru(\$id)* yang ada pada *controller* akan dipanggil, selanjutnya akan diproses oleh *method delete(\$id)* yang terdapat di bagian *model* untuk selanjutnya melakukan *query* pada tabel Guru di *database*. Hasil *query* yang sukses akan mengirimkan respon *success* dan selanjutnya

28. Sequence Diagram Lihat Daftar Siswa

Sequence diagram lihat daftar siswa merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkaitan dengan fungsionalitas lihat daftar siswa. Pemodelan *sequence diagram* fungsionalitas lihat daftar siswa dapat dilihat pada Gambar 4. 95.

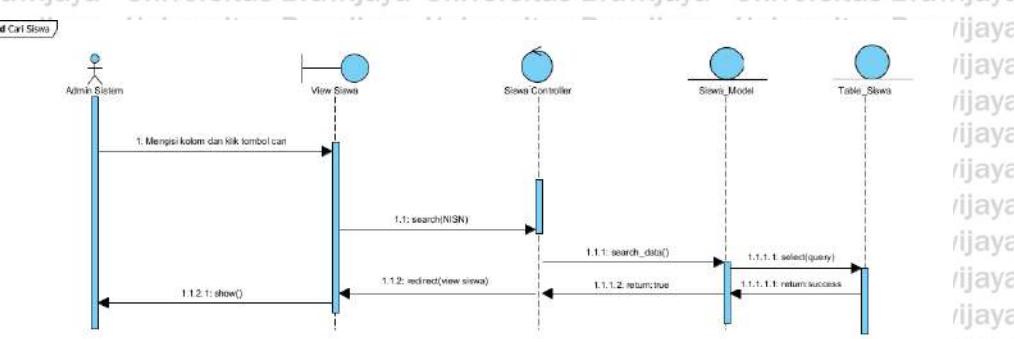


Gambar 4. 95 *Sequence Diagram* Lihat Daftar Siswa

Proses lihat daftar siswa dimulai dengan aktor melakukan klik terhadap menu siswa, setelah itu bagian *View Siswa* akan memanggil *method index()* yang ada pada bagian *Siswa Controller*, selanjutnya *method show_data()* yang ada pada *Siswa Model* akan dipanggil oleh *controller*, *method* tersebut akan melakukan *query* pada *Table Siswa* di *database*. Bagian *Table Siswa* akan memberikan respon *success* pada *model* dan selanjutnya akan dikirim status *true* pada *controller* yang selanjutnya akan mengirimkan perintah untuk mengarahkan aktor pada *view siswa* yang sudah menerima data dari *database*.

29. Sequence Diagram Cari Data Siswa

Sequence diagram cari data siswa merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkaitan dengan fungsionalitas cari data siswa. Pemodelan *sequence diagram* fungsionalitas cari data siswa dapat dilihat pada Gambar 4. 96.

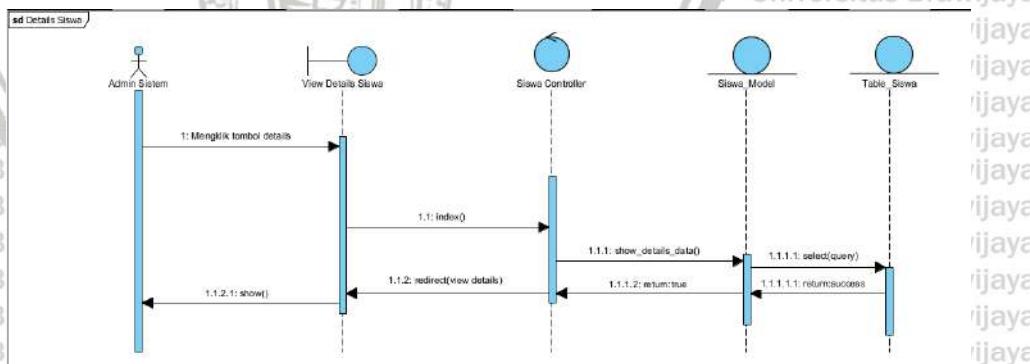


Gambar 4. 96 Sequence Diagram Cari Data Siswa

Proses cari data siswa dimulai dengan aktor mengisi kolom pencarian dan klik tombol cari, setelah itu bagian *View Siswa* akan memanggil *method search(NISN)* yang ada pada bagian *Siswa Controller*, selanjutnya *method search_data()* yang ada pada *Siswa Model* akan dipanggil oleh *controller*, *method* tersebut akan melakukan *query* pada *Table Siswa* di *database*. Bagian *Table Siswa* akan memberikan respon *success* pada *model* dan selanjutnya akan dikirim status *true* pada *controller* yang selanjutnya akan mengirimkan perintah untuk mengarahkan aktor pada *view siswa* yang sudah menerima data dari *database*.

30. Sequence Diagram Detail Data Siswa

Sequence diagram detail data siswa merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas detail data siswa. Pemodelan *sequence diagram* fungsionalitas detail data siswa dapat dilihat pada Gambar 4. 97.



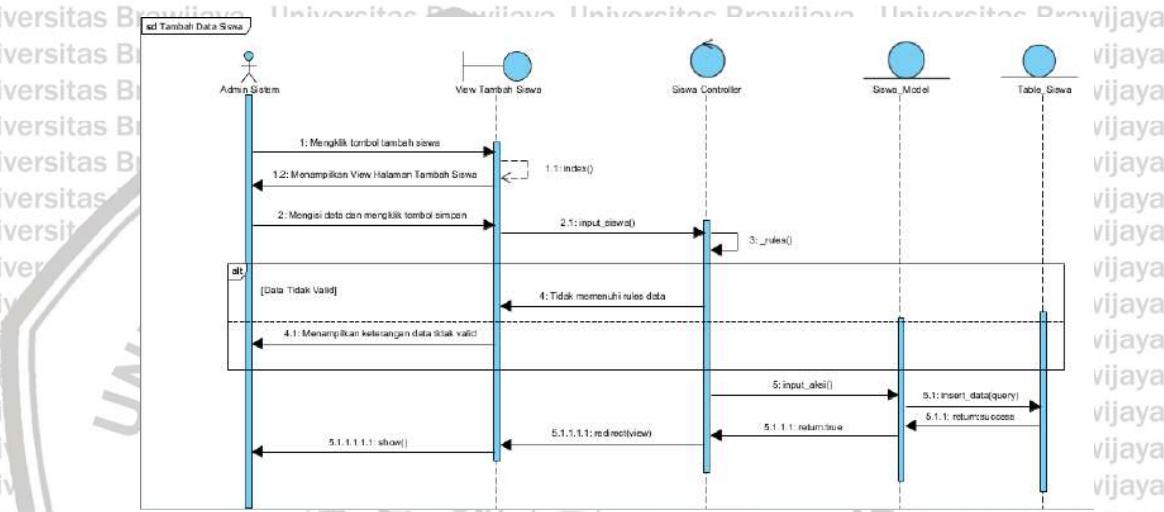
Gambar 4. 97 Sequence Diagram Detail Data Siswa

Proses lihat detail siswa dimulai dengan aktor melakukan klik terhadap tombol *details* yang ada di daftar siswa, setelah itu bagian *View Details Siswa* akan memanggil *method details()* yang ada pada bagian *Siswa Controller*, selanjutnya *method show_details_data()* yang ada pada *Siswa Model* akan dipanggil oleh *controller*, *method* tersebut akan

mengakses *Table Siswa* di *database*. Bagian *Table Siswa* akan memberikan respon *success* pada *model* dan selanjutnya akan dikirim status *true* pada *controller* yang selanjutnya akan mengirimkan perintah untuk mengarahkan aktor pada *view siswa* yang sudah menerima data dari *database*.

31. Sequence Diagram Tambah Data Siswa

Sequence diagram tambah data siswa merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas tambah data siswa. Pemodelan *sequence diagram* fungsionalitas tambah data siswa dapat dilihat pada Gambar 4.98.



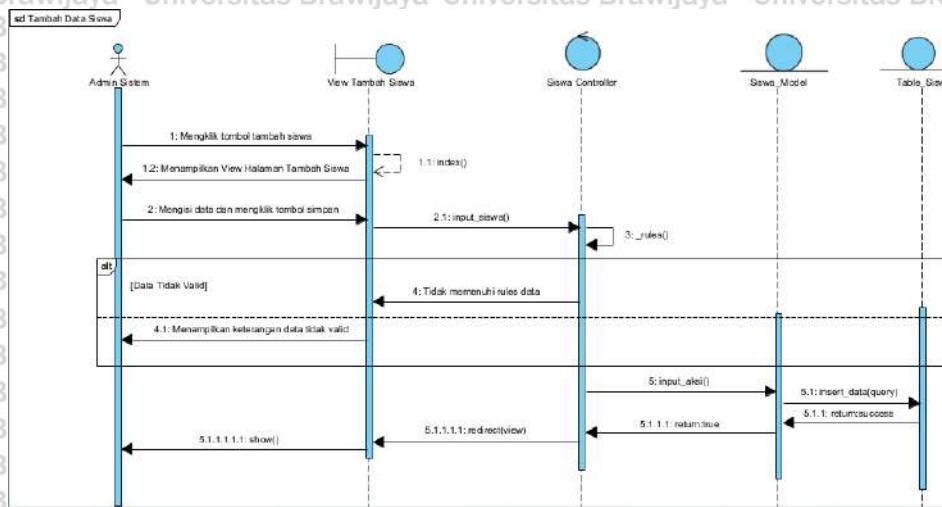
Gambar 4. 98 Sequence Diagram Tambah Data Siswa

Proses tambah data siswa dimulai dengan aktor melakukan klik pada tombol tambah data siswa yang ada diatas daftar siswa, Selanjutnya akan ditampilkan halaman tambah guru oleh bagian *View Tambah Guru* dengan memanggil *method* *index()*. Setelah aktor mengisi data yang diminta oleh *field* yang ada dan menekan tombol simpan. Selanjutnya *method* *input_siswa()* yang ada pada *controller* akan dipanggil, selanjutnya di bagian *controller* akan melakukan pengecekan apakah data yang dimasukkan oleh aktor valid (terisi semua, tidak ada duplikat), saat data dinyatakan tidak valid maka akan ditampilkan keterangan bahwa data yang dimasukkan tidak valid. Apabila data yang dimasukkan valid maka akan dieksekusi oleh *method* *input_aksi()* untuk ditambahkan pada tabel siswa yang ada di *database*. Hasil *query* yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman daftar siswa.

32. Sequence Diagram Ubah Data Siswa

Sequence diagram ubah data siswa merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas ubah data siswa.

Pemodelan *sequence diagram* fungsionalitas ubah data siswa dapat dilihat pada Gambar 4. 99.

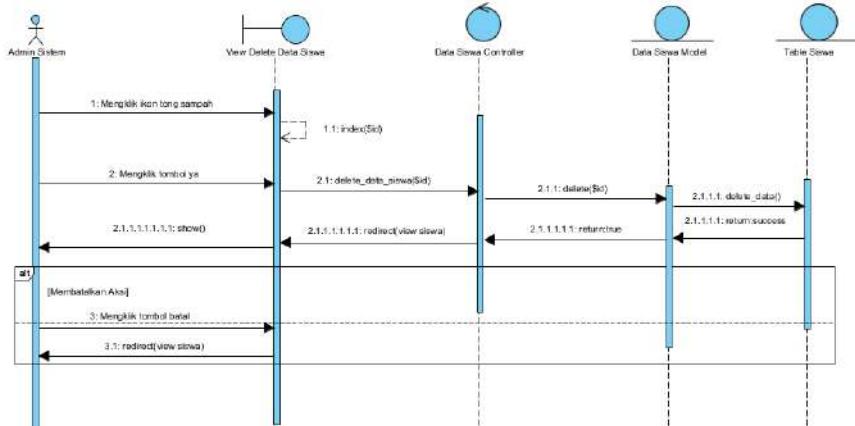


Gambar 4. 99 Sequence Diagram Ubah Data Siswa

Proses ubah data siswa dimulai dengan aktor melakukan klik pada ikon pensil yang di daftar siswa, Selanjutnya akan ditampilkan halaman ubah data siswa oleh bagian *Update Data Siswa* dengan memanggil *method index(\$id)*. Setelah aktor mengisi data yang diminta oleh *field* yang ada dan menekan tombol simpan. Selanjutnya *method update_data_siswa(\$id)* yang ada pada *controller* akan dipanggil, selanjutnya di bagian *controller* akan melakukan pengecekan apakah data yang dimasukkan oleh aktor valid (terisi semua, tidak ada duplikat), saat data dinyatakan tidak valid maka akan ditampilkan keterangan bahwa data yang dimasukkan tidak valid. Apabila data yang dimasukkan valid maka akan dieksekusi oleh *method update_aksi()* untuk dilakukan perubahan data sesuai dengan *id* pada tabel siswa yang ada di *database*. Hasil *query* yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman daftar siswa.

33. Sequence Diagram Hapus Data Siswa

Sequence diagram hapus data siswa merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkaitan dengan fungsionalitas hapus data siswa. Pemodelan *sequence diagram* fungsionalitas hapus data siswa dapat dilihat pada Gambar 4. 100.

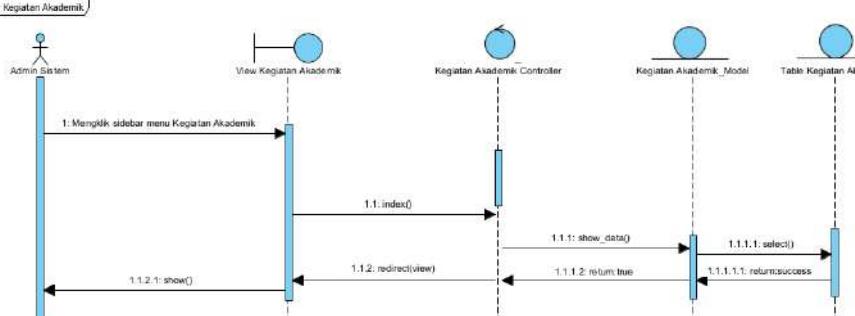


Gambar 4. 100 Sequence Diagram Hapus Data Siswa

Proses hapus data siswa dimulai dengan aktor melakukan klik pada ikon tong sampah yang ada pada daftar siswa, Selanjutnya akan ditampilkan konfirmasi penghapusan data oleh bagian *View Delete Data Siswa* dengan memanggil *method* *index(\$id)*. Apabila aktor mengklik tombol “ya” maka *method* *delete_data_siswa(\$id)* yang ada pada *controller* akan dipanggil, selanjutnya akan diproses oleh *method* *delete(\$id)* yang terdapat di bagian *model* untuk selanjutnya melakukan *query* pada tabel *Siswa* di *database*. Hasil *query* yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman siswa.

34. Sequence Diagram Lihat Daftar Kegiatan Akademik

Sequence diagram lihat daftar kegiatan akademik merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas lihat daftar kegiatan akademik. Pemodelan *sequence diagram* fungsionalitas lihat daftar kegiatan akademik dapat dilihat pada Gambar 4. 101.



Gambar 4. 101 Sequence Diagram Lihat Daftar Kegiatan Akademik

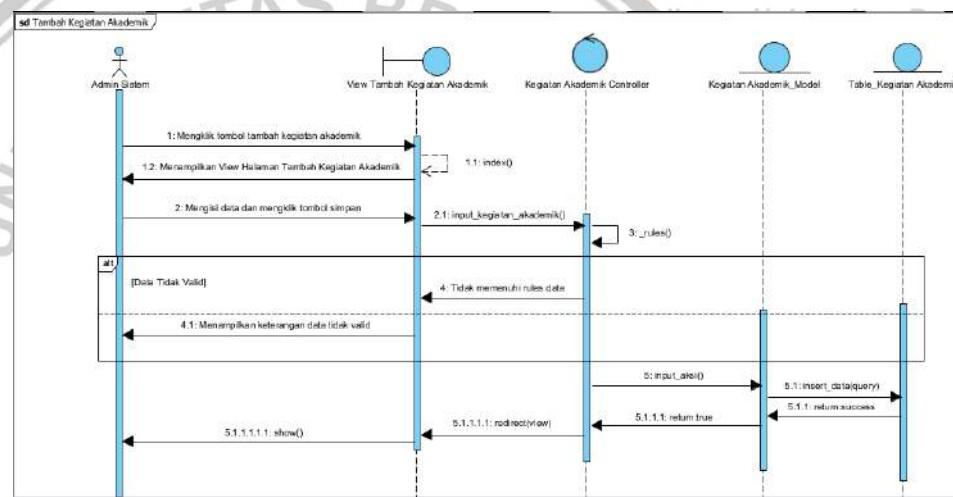
Proses lihat daftar kegiatan akademik dimulai dengan aktor melakukan klik terhadap menu kegiatan akademik, setelah itu bagian *View Kegiatan Akademik* akan memanggil *method* *index()* yang ada pada bagian

Kegiatan Akademik Controller, selanjutnya *method show_data()* yang ada pada *Kegiatan Akademik Model* akan dipanggil oleh *controller*, *method* tersebut akan melakukan *query* pada *Table Kegiatan Akademik* di *database*. Bagian *Table Kegiatan Akademik* akan memberikan respon *success* pada *model* dan selanjutnya akan dikirim status *true* pada *controller* yang selanjutnya akan mengirimkan perintah untuk mengarahkan aktor pada *view* kegiatan akademik yang sudah menerima data dari *database*.

35. Sequence Diagram Tambah Data Kegiatan Akademik

Sequence diagram tambah data kegiatan akademik merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas tambah data kegiatan akademik. Pemodelan *sequence diagram* fungsionalitas tambah data kegiatan akademik dapat dilihat pada

Gambar 4. 102.



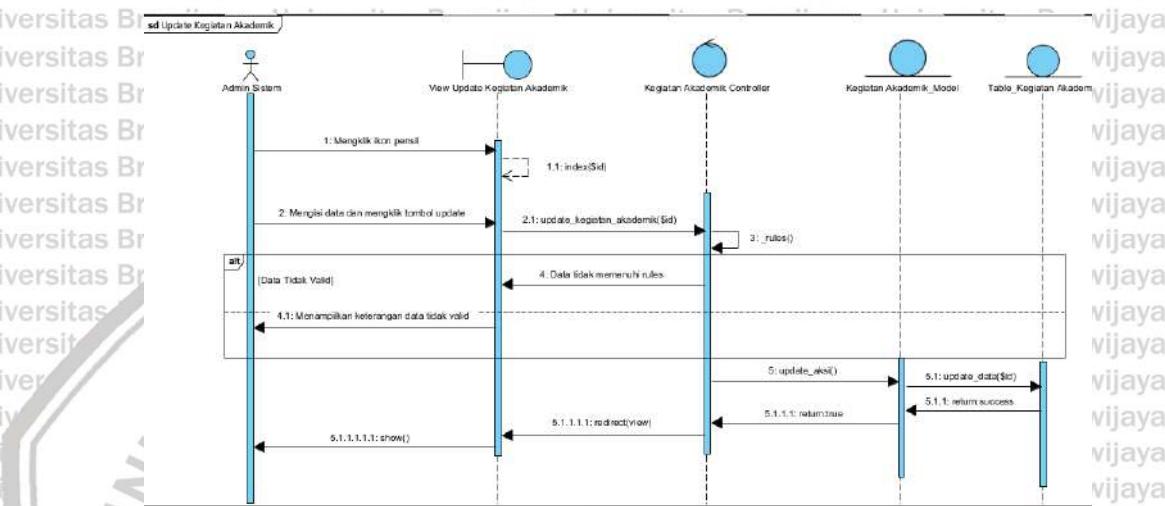
Gambar 4. 102 Sequence Diagram Tambah Data Kegiatan Akademik

Proses tambah data kegiatan akademik dimulai dengan aktor melakukan klik pada tombol tambah kegiatan akademik yang ada diatas daftar kegiatan akademik, Selanjutnya akan ditampilkan halaman tambah kegiatan akademik oleh bagian *View Tambah Kegiatan Akademik* dengan memanggil *method index()*. Setelah aktor mengisi data yang diminta oleh *field* yang ada dan menekan tombol simpan. Selanjutnya *method input_kegiatan_akademik()* yang ada pada *controller* akan dipanggil, selanjutnya di bagian *controller* akan melakukan pengecekan apakah data yang dimasukkan oleh aktor valid (terisi semua, tidak ada duplikat), saat data dinyatakan tidak valid maka akan ditampilkan keterangan bahwa data yang dimasukkan tidak valid. Apabila data yang dimasukkan valid maka akan dieksekusi oleh *method input_aksi()* untuk ditambahkan pada tabel Kegiatan Akademik yang ada di *database*. Hasil *query* yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller*.

dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman daftar kegiatan akademik.

36. Sequence Diagram Ubah Data Kegiatan Akademik

Sequence diagram ubah data kegiatan akademik merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas ubah data kegiatan akademik. Pemodelan *sequence diagram* fungsionalitas ubah data kegiatan akademik dapat dilihat pada Gambar 4. 103.

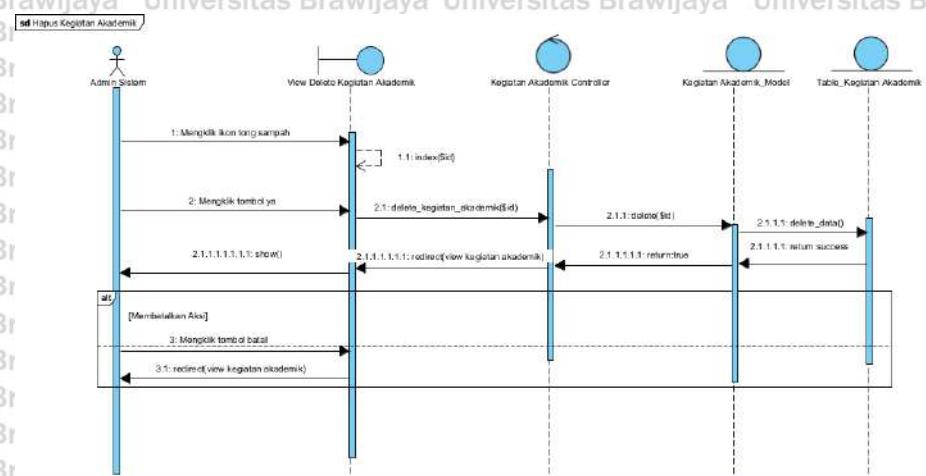


Gambar 4. 103 Sequence Diagram Ubah Data Kegiatan Akademik

Proses ubah data kegiatan akademik dimulai dengan aktor melakukan klik pada ikon pensil yang di daftar kegiatan akademik. Selanjutnya akan ditampilkan halaman ubah data kegiatan akademik oleh bagian *View Update Kegiatan Akademik* dengan memanggil *method* *index(\$id)*. Setelah aktor mengisi data yang diminta oleh *field* yang ada dan menekan tombol simpan. Selanjutnya *method* *update_kegiatan_akademik(\$id)* yang ada pada *controller* akan dipanggil, selanjutnya di bagian *controller* akan melakukan pengecekan apakah data yang dimasukkan oleh aktor valid (terisi semua, tidak ada duplikat), saat data dinyatakan tidak valid maka akan ditampilkan keterangan bahwa data yang dimasukkan tidak valid. Apabila data yang dimasukkan valid maka akan dieksekusi oleh *method* *update_aksi()* untuk dilakukan perubahan data sesuai dengan *id* pada tabel Kegiatan Akademik yang ada di *database*. Hasil query yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman daftar kegiatan akademik.

37. Sequence Diagram Hapus Data Kegiatan Akademik

Sequence diagram hapus data kegiatan akademik merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas hapus data

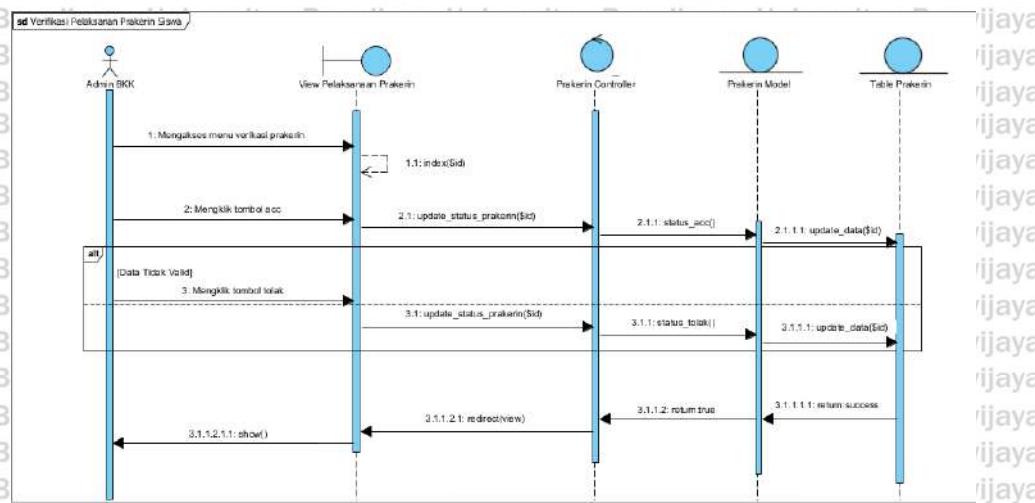


Gambar 4. 104 Sequence Diagram Hapus Data Kegiatan Akademik

Proses hapus data kegiatan akademik dimulai dengan aktor melakukan klik pada ikon tong sampah yang ada pada daftar kegiatan akademik. Selanjutnya akan ditampilkan konfirmasi penghapusan data oleh bagian *View Delete Kegiatan Akademik* dengan memanggil *method index(\$id)*. Apabila aktor mengklik tombol “ya” maka *method delete_kegiatan_akademik(\$id)* yang ada pada *controller* akan dipanggil, selanjutnya akan diproses oleh *method delete(\$id)* yang terdapat di bagian *model* untuk selanjutnya melakukan *query* pada tabel *Kegiatan Akademik* di *database*. Hasil *query* yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman kegiatan akademik.

38. Sequence Diagram Validasi Pelaksanaan Prakerin

Sequence diagram validasi pelaksanaan prakerin merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsi validasi pelaksanaan prakerin. Pemodelan *sequence diagram* fungsi validasi pelaksanaan prakerin dapat dilihat pada Gambar 4. 105.

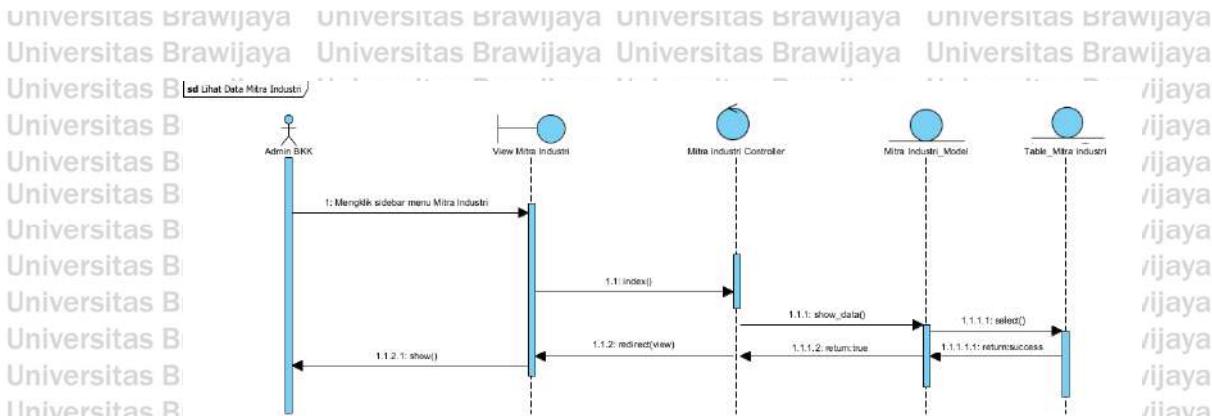


Gambar 4. 105 Sequence Diagram Verifikasi Pelaksanaan Prakerin

Proses validasi penuntasan prakerin dimulai dengan aktor melakukan klik pada menu verifikasi prakerin, selanjutnya *View Pelaksanaan Prakerin* akan menampilkan halaman dengan memanggil *method index()* yang ada pada *class* tersebut. Setelah itu aktor akan mengklik tombol “acc” yang selanjutnya akan diproses oleh *method update_status_prakerin(\$id)* yang ada pada bagian *Prakerin Controller*, selanjutnya *controller* akan melakukan pemanggilan pada *method status_acc()* yang ada pada bagian *Prakerin Model*. Apabila aktor menemukan kesalahan data maka aktor perlu menekan tombol “tolak” yang akan melakukan pemanggilan *method update_status_prakerin(\$id)* di bagian *Prakerin Controller* dan meneruskan proses pada *method status_tolak()* yang ada di *Prakerin Model*. Masing-masing *flow* proses tersebut akan melakukan perubahan (*update*) pada data di *database* tepatnya pada *Table Prakerin*. Respon *success* akan dikirimkan pada *model* saat data berhasil dilakukan *update*, selanjutnya *model* akan mengirimkan respon *true* yang akan melakukan pengarahan aktor pada *view prakerin* yang sudah berubah status.

39. Sequence Diagram Lihat Daftar Mitra Industri

Sequence diagram lihat daftar mitra industri merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkaitan dengan fungsionalitas lihat daftar mitra industri. Pemodelan sequence diagram fungsionalitas lihat daftar mitra industri dapat dilihat pada Gambar 4. 106.

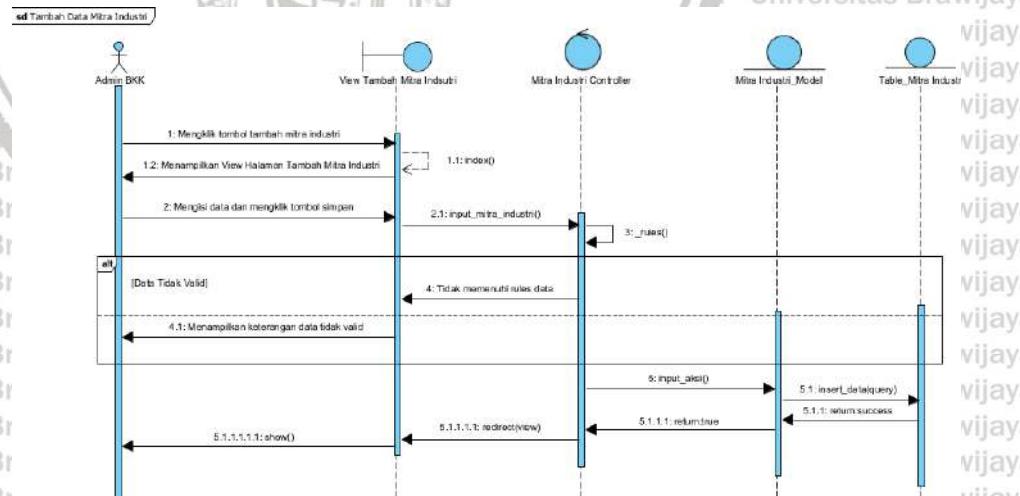


Gambar 4. 106 Sequence Diagram Lihat Daftar Mitra Industri

Proses lihat daftar mitra industri dimulai dengan aktor melakukan klik terhadap menu kegiatan akademik, setelah itu bagian *View Mitra Industri* akan memanggil *method index()* yang ada pada bagian *Mitra Industri Controller*, selanjutnya *method show_data()* yang ada pada *Mitra Industri Model* akan dipanggil oleh *controller*, *method* tersebut akan melakukan *query* pada *Table Mitra Industri* di *database*. Bagian *Table Mitra Industri* akan memberikan respon *success* pada *model* dan selanjutnya akan dikirim status *true* pada *controller* yang selanjutnya akan mengirimkan perintah untuk mengarahkan aktor pada *view mitra industri* yang sudah menerima data dari *database*.

40. Sequence Diagram Tambah Data Mitra Industri

Sequence diagram tambah data mitra industri merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas tambah data mitra industri. Pemodelan *sequence diagram* fungsionalitas tambah data mitra industri dapat dilihat pada Gambar 4. 107.



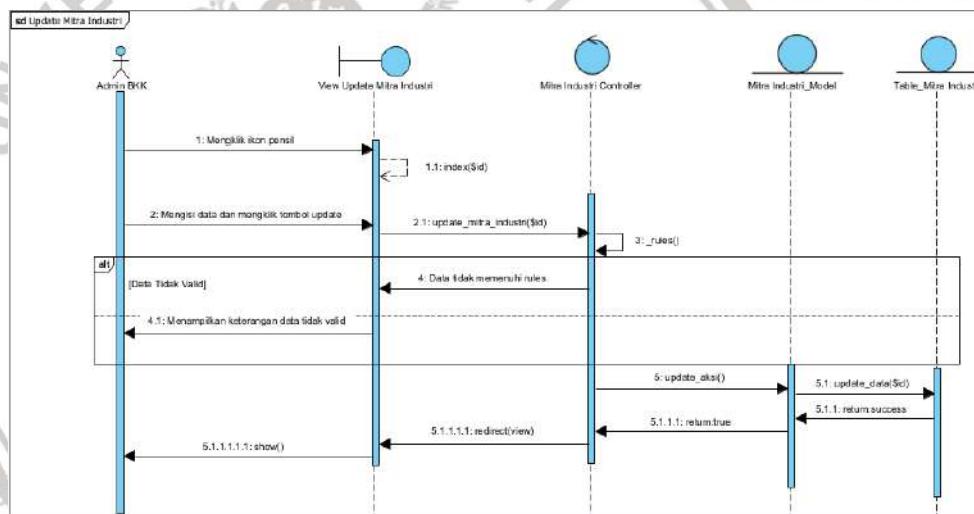
Gambar 4. 107 Sequence Diagram Tambah Data Mitra Industri

Proses tambah data mitra industri dimulai dengan aktor melakukan klik pada tombol tambah mitra industri yang ada diatas daftar kegiatan akademik, Selanjutnya akan ditampilkan halaman tambah mitra industri

oleh bagian *View Tambah Mitra Industri* dengan memanggil *method index()*. Setelah aktor mengisi data yang diminta oleh *field* yang ada dan menekan tombol simpan. Selanjutnya *method input_mitra_industri()* yang ada pada *controller* akan dipanggil, selanjutnya di bagian *controller* akan melakukan pengecekan apakah data yang dimasukkan oleh aktor valid (terisi semua, tidak ada duplikat), saat data dinyatakan tidak valid maka akan ditampilkan keterangan bahwa data yang dimasukkan tidak valid. Apabila data yang dimasukkan valid maka akan dieksekusi oleh *method input_aksi()* untuk ditambahkan pada tabel *Mitra Industri* yang ada di *database*. Hasil query yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman daftar mitra industri.

41. Sequence Diagram Ubah Data Mitra Industri

Sequence diagram ubah data mitra industri merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas ubah data mitra industri. Pemodelan *sequence diagram* fungsionalitas ubah data mitra industri dapat dilihat pada Gambar 4. 108.



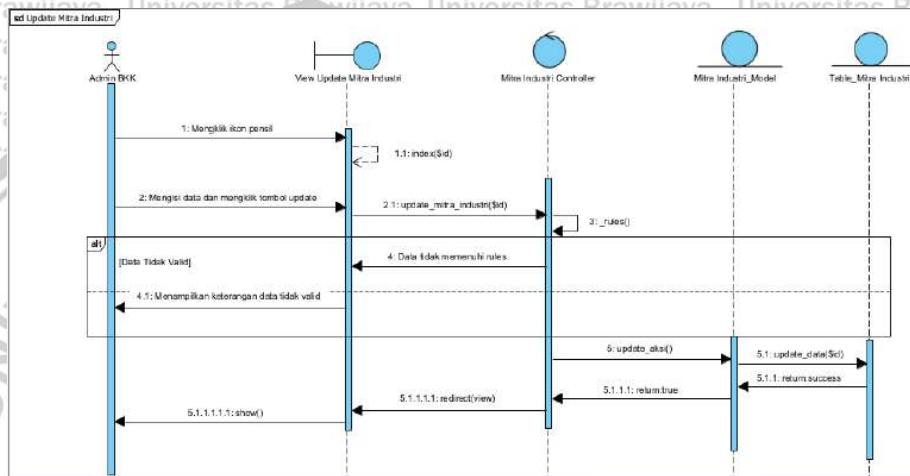
Gambar 4. 108 Sequence Diagram Ubah Data Mitra Industri

Proses ubah data mitra industri dimulai dengan aktor melakukan klik pada ikon pensil yang di daftar mitra industri, Selanjutnya akan ditampilkan halaman ubah data mitra industri oleh bagian *View Update Mitra Industri* dengan memanggil *method index(\$id)*. Setelah aktor mengisi data yang diminta oleh *field* yang ada dan menekan tombol simpan. Selanjutnya *method update_mitra_industri(\$id)* yang ada pada *controller* akan dipanggil, selanjutnya di bagian *controller* akan melakukan pengecekan apakah data yang dimasukkan oleh aktor valid (terisi semua, tidak ada duplikat), saat data dinyatakan tidak valid maka akan ditampilkan keterangan bahwa data yang dimasukkan tidak valid. Apabila data yang dimasukkan valid maka akan dieksekusi oleh *method update_aksi()* untuk

dilakukan perubahan data sesuai dengan *id* pada tabel mitra industri yang ada di *database*. Hasil *query* yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman daftar mitra industri.

42. Sequence Diagram Hapus Data Mitra Industri

Sequence diagram hapus data mitra industri merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian sistem informasi yang berkenaan dengan fungsionalitas hapus data mitra industri. Pemodelan *sequence diagram* fungsionalitas hapus data mitra industri dapat dilihat pada Gambar 4. 109.

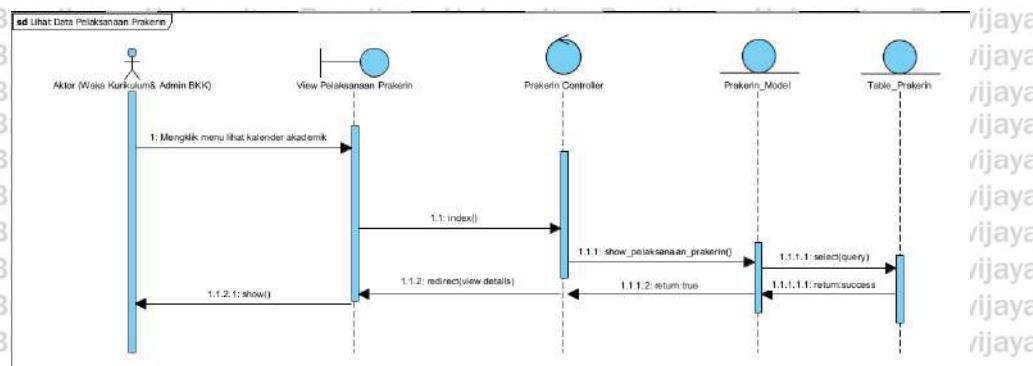


Gambar 4. 109 Sequence Diagram Hapus Data Mitra Industri

Proses hapus data mitra industri dimulai dengan aktor melakukan klik pada ikon tong sampah yang ada pada daftar mitra industri. Selanjutnya akan ditampilkan konfirmasi penghapusan data oleh bagian *View Delete Mitra Industri* dengan memanggil method *index(\$id)*. Apabila aktor mengklik tombol “ya” maka method *delete_mitra_industri(\$id)* yang ada pada *controller* akan dipanggil, selanjutnya akan diproses oleh method *delete(\$id)* yang terdapat di bagian *model* untuk selanjutnya melakukan *query* pada tabel *Mitra Industri* di *database*. Hasil *query* yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman mitra industri.

43. Sequence Diagram Lihat Pelaksanaan Prakerin

Sequence diagram lihat data pelaksanaan prakerin merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian sistem informasi yang berkenaan dengan fungsionalitas lihat data pelaksanaan prakerin. Pemodelan *sequence diagram* fungsionalitas lihat data pelaksanaan prakerin dapat dilihat pada Gambar 4. 110.

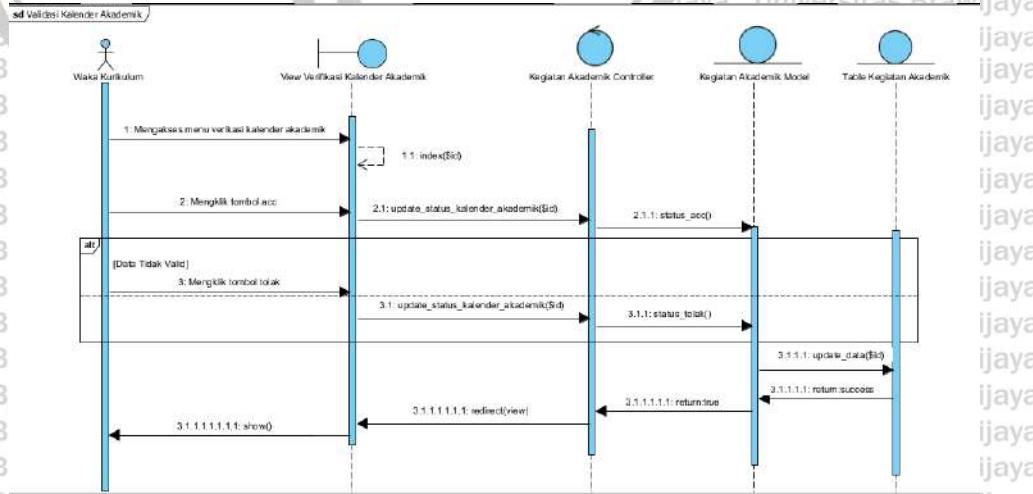


Gambar 4. 110 Sequence Diagram Lihat Pelaksanaan Prakerin

Proses lihat pelaksanaan prakerin dimulai dengan aktor melakukan klik terhadap menu lihat pelaksanaan prakerin, setelah itu bagian *Pelaksanaan Prakerin* akan memanggil *method* *pelaksanaan()* yang ada pada bagian *Prakerin Controller*, selanjutnya *method* *show_pelaksanaan_prakerin()* yang ada pada *Prakerin Model* akan dipanggil oleh *controller*, *method* tersebut akan melakukan *query* pada *Table Prakerin* di *database*. Bagian *Table Prakerin* akan memberikan respon *success* pada *model* dan selanjutnya akan dikirim status *true* pada *controller* yang selanjutnya akan mengirimkan perintah untuk mengarahkan aktor pada *view* pelaksanaan prakerin yang sudah menerima data dari *database*.

44. Sequence Diagram Validasi Kalender Akademik

Sequence diagram validasi kalender akademik merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian sistem informasi yang berkenaan dengan fungsionalitas validasi kalender akademik. Pemodelan *sequence diagram* fungsionalitas validasi kalender akademik dapat dilihat pada Gambar 4. 111.

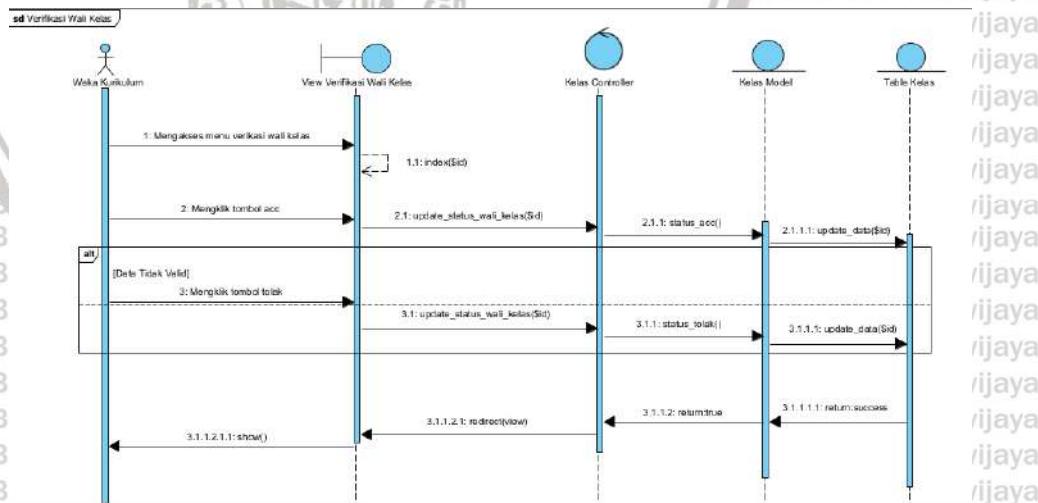


Gambar 4. 111 Sequence Diagram Validasi Kalender Akademik

Proses validasi kalender akademik dimulai dengan aktor melakukan klik pada menu verifikasi kalender akademik, selanjutnya *View Verifikasi Kalender Akademik* akan menampilkan halaman dengan memanggil *method index()* yang ada pada *classs* tersebut. Setelah itu aktor akan mengklik tombol “acc” yang selanjutnya akan diproses oleh *method update_status_akademik(\$id)* yang ada pada bagian *Kegiatan Akademik Controller*, selanjutnya *controller* akan melakukan pemanggilan pada *method status_acc()* yang ada pada bagian Kegiatan Akademik Model. Apabila aktor menemukan kesalahan data maka aktor perlu menekan tombol “tolak” yang akan melakukan pemanggilan *method update_status_kalender_akademik(\$id)* di bagian *Kegiatan Akademik Controller* dan meneruskan proses pada *method status_tolak()* yang ada di *Model Kegiatan Akademik*. Masing-masing *flow* proses tersebut akan melakukan perubahan (*update*) pada data di *database* tepatnya pada *Table Kegiatan Akademik*. Respon *success* akan dikirimkan pada *model* saat data berhasil dilakukan *update*, selanjutnya *model* akan mengirimkan respon *true* yang akan melakukan pengarahan aktor pada *view* halaman kalender akademik yang sudah berubah status.

45. Sequence Diagram Verifikasi Wali Kelas

Sequence diagram verifikasi wali kelas merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas validasi penetapan wali kelas. Pemodelan *sequence diagram* fungsionalitas validasi penetapan wali kelas dapat dilihat pada Gambar 4. 112.



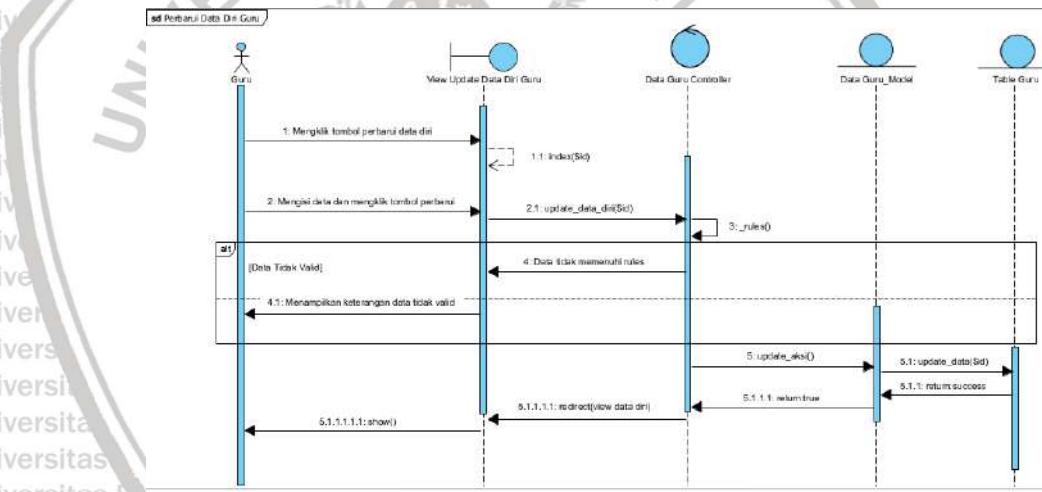
Gambar 4. 112 Sequence Diagram Verifikasi Wali Kelas

Proses validasi kalender akademik dimulai dengan aktor melakukan klik pada menu verifikasi wali kelas, selanjutnya *View Verifikasi Wali Kelas* akan menampilkan halaman dengan memanggil *method index()* yang ada pada *classs* tersebut. Setelah itu aktor akan mengklik tombol “acc” yang akan melakukan pemanggilan *method status_acc()* yang ada pada bagian *Kegiatan Akademik Controller*, selanjutnya *controller* akan melakukan pemanggilan pada *method status_tolak()* yang ada di *Model Kegiatan Akademik*. Masing-masing *flow* proses tersebut akan melakukan perubahan (*update*) pada data di *database* tepatnya pada *Table Kegiatan Akademik*. Respon *true* akan dikirimkan pada *model* saat data berhasil dilakukan *update*, selanjutnya *model* akan mengirimkan respon *true* yang akan melakukan pengarahan aktor pada *view* halaman kalender akademik yang sudah berubah status.

selanjutnya akan diproses oleh *method update_status_wali_kelas(\$id)* yang ada pada bagian *Kelas Controller*, selanjutnya *controller* akan melakukan pemanggilan pada *method status_acc()* yang ada pada bagian *Kelas Model*. Apabila aktor menemukan kesalahan data maka aktor perlu menekan tombol “tolak” yang akan melakukan pemanggilan *method update_status_wali_kelas(\$id)* di bagian *Kelas Controller* dan meneruskan proses pada *method status_tolak()* yang ada di *Kelas Model*. Masing-masing *flow* proses tersebut akan melakukan perubahan (*update*) pada data di *database* tepatnya pada *Table Kelas*. Respon *success* akan dikirimkan pada *model* saat data berhasil dilakukan *update*, selanjutnya *model* akan mengirimkan respon *true* yang akan melakukan pengarahan aktor pada *view* verifikasi wali kelas yang sudah berubah status.

46. Sequence Diagram Perbarui Data Diri Guru

Sequence diagram perbarui data diri guru merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkaitan dengan fungsionalitas perbarui data diri guru. Pemodelan *sequence diagram* fungsionalitas perbarui data diri guru dapat dilihat pada Gambar 4. 113.



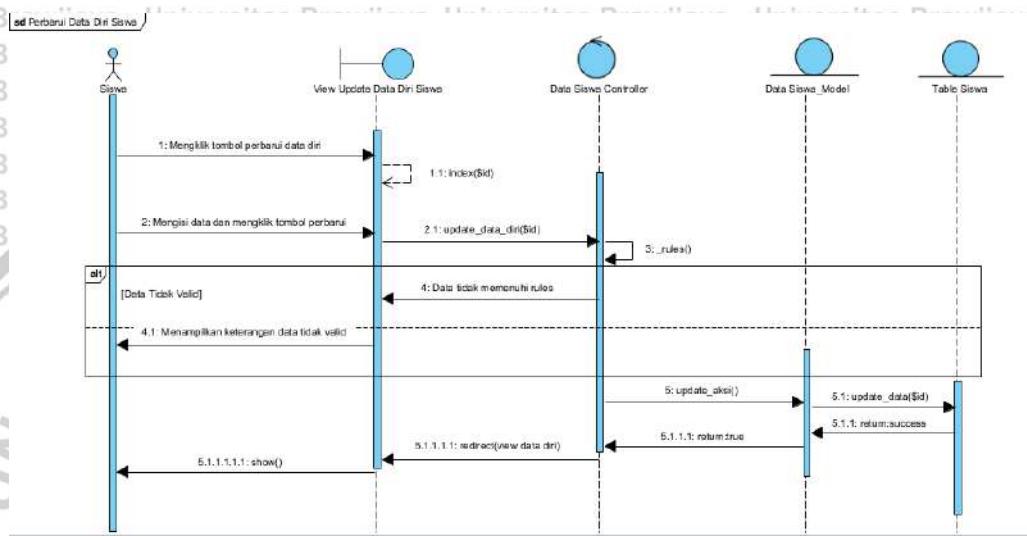
Gambar 4. 113 *Sequence Diagram* Perbarui Data Diri Guru

Proses perbarui data guru dimulai dengan aktor melakukan klik pada tombol perbarui data diri, Selanjutnya akan ditampilkan halaman ubah data guru oleh bagian *View Update Data Diri Guru* dengan memanggil *method index(\$id)*. Setelah aktor mengisi data yang diminta oleh *field* yang ada dan menekan tombol simpan. Selanjutnya *method update_data_diri(\$id)* yang ada pada *controller* akan dipanggil, selanjutnya di bagian *controller* akan melakukan pengecekan apakah data yang dimasukkan oleh aktor valid (terisi semua, tidak ada duplikat), saat data dinyatakan tidak valid maka akan ditampilkan keterangan bahwa data yang dimasukkan tidak valid. Apabila data yang dimasukkan valid maka akan dieksekusi oleh *method update_aksi()* untuk dilakukan perubahan data sesuai dengan *id* pada tabel guru yang ada di *database*. Hasil *query* yang

sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman daftar guru.

47. Sequence Diagram Perbarui Data Diri Siswa

Sequence diagram perbarui data diri siswa merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas perbarui data diri siswa. Pemodelan *sequence diagram* fungsionalitas perbarui data diri siswa dapat dilihat pada Gambar 4.114.



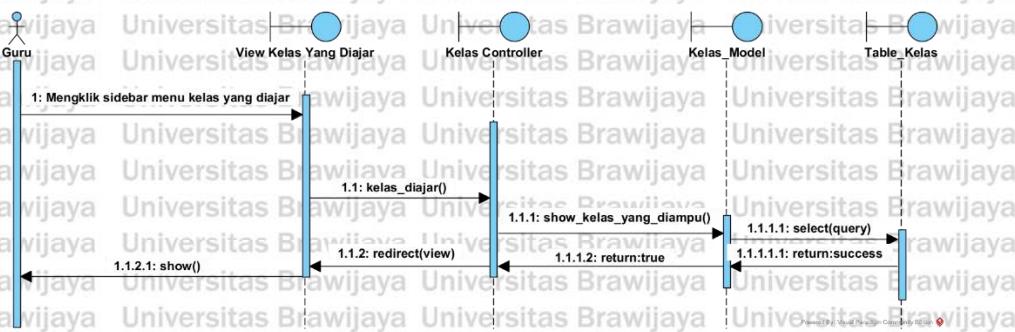
Gambar 4. 114 Sequence Diagram Perbarui Data Diri Siswa

Proses perbarui data diri siswa dimulai dengan aktor melakukan klik pada tombol perbarui data diri, Selanjutnya akan ditampilkan halaman ubah data diri siswa oleh bagian *View Update Data Diri Siswa* dengan memanggil method *index(\$id)*. Setelah aktor mengisi data yang diminta oleh *field* yang ada dan menekan tombol simpan. Selanjutnya method *update_data_diri(\$id)* yang ada pada *controller* akan dipanggil, selanjutnya di bagian *controller* akan melakukan pengecekan apakah data yang dimasukkan oleh aktor valid (terisi semua, tidak ada duplikat), saat data dinyatakan tidak valid maka akan ditampilkan keterangan bahwa data yang dimasukkan tidak valid. Apabila data yang dimasukkan valid maka akan dieksekusi oleh method *update_aksi()* untuk dilakukan perubahan data sesuai dengan *id* pada tabel siswa yang ada di *database*. Hasil query yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman daftar siswa.

48. Sequence Diagram Lihat Daftar Kelas Yang Diwakili

Sequence diagram lihat daftar kelas yang diwakili merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-

bagian sistem informasi yang berkenaan dengan fungsionalitas lihat daftar kelas yang diwakili. Pemodelan *sequence diagram* fungsionalitas lihat daftar kelas yang diwakili dapat dilihat pada Gambar 4. 115.

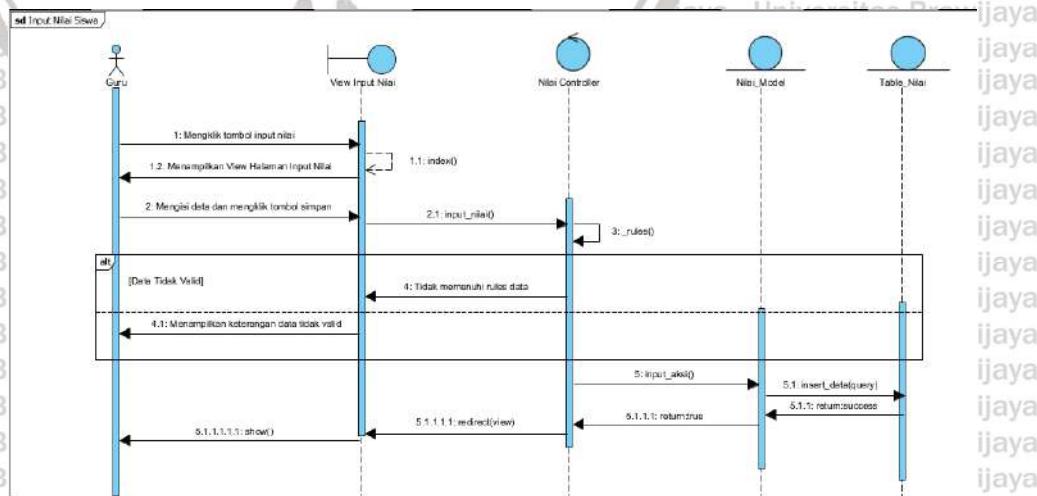


Gambar 4. 115 Sequence Diagram Lihat Kelas Yang Diwakili

Proses lihat kelas yang diajar dimulai dengan aktor melakukan klik terhadap menu kelas yang diajar, setelah itu bagian *View Kelas Yang Diajar* akan memanggil *method* *kelas_diajar()* yang ada pada bagian *Kelas Controller*, selanjutnya *method* *show_kelas.yang_diampu()* yang ada pada *Kelas Model* akan dipanggil oleh *controller*, *method* tersebut akan melakukan *query* pada *Table Kelas* di *database*. Bagian *Table Kelas* akan memberikan respon *success* pada *model* dan selanjutnya akan dikirim status *true* pada *controller* yang selanjutnya akan mengirimkan perintah untuk mengarahkan aktor pada *view* kelas yang diajar yang sudah menerima data dari *database*.

49. Sequence Diagram Input Nilai Hasil Evaluasi

*Sequence diagram input nilai hasil evaluasi merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas *input* nilai hasil evaluasi. Pemodelan *sequence diagram* fungsionalitas *input* nilai hasil evaluasi dapat dilihat pada Gambar 4. 116.*

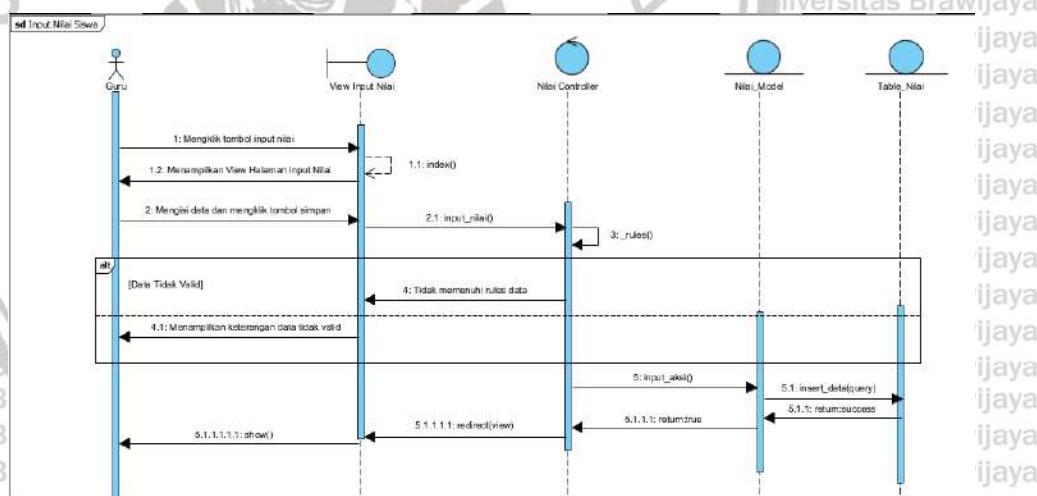


Gambar 4. 116 Sequence Diagram Input Nilai Hasil Evaluasi

Proses *input* nilai hasil evaluasi dimulai dengan aktor melakukan klik pada tombol *input* nilai di daftar kelas yang diajar. Selanjutnya akan ditampilkan halaman *input* nilai oleh bagian *View Input Nilai* dengan memanggil *method index()*. Setelah aktor mengisi data yang diminta oleh *field* yang ada dan menekan tombol simpan. Selanjutnya *method input_nilai()* yang ada pada *controller* akan dipanggil, selanjutnya di bagian *controller* akan melakukan pengecekan apakah data yang dimasukkan oleh aktor valid (terisi semua, tidak ada duplikat), saat data dinyatakan tidak valid maka akan ditampilkan keterangan bahwa data yang dimasukkan tidak valid. Apabila data yang dimasukkan valid maka akan dieksekusi oleh *method input_aksi()* untuk ditambahkan pada tabel *Nilai* yang ada di *database*. Hasil *query* yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman daftar nilai.

50. Sequence Diagram Lihat Daftar Nilai Hasil Evaluasi

Sequence diagram lihat daftar nilai hasil evaluasi merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas lihat daftar nilai hasil evaluasi. Pemodelan *sequence diagram* fungsionalitas lihat daftar nilai hasil evaluasi dapat dilihat pada Gambar 4. 117.

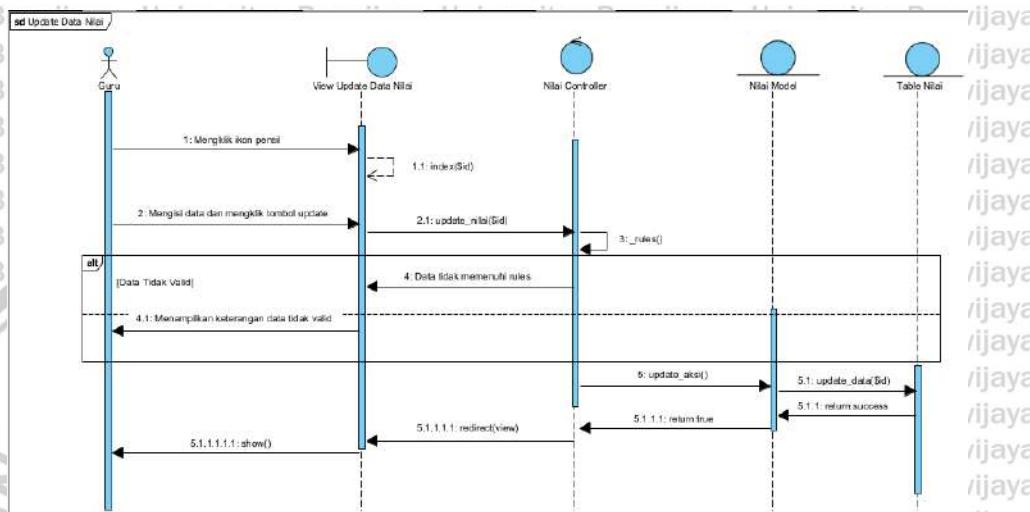


Gambar 4. 117 Sequence Diagram Lihat Daftar Nilai Hasil Evaluasi

Proses lihat daftar nilai dimulai dengan aktor melakukan klik terhadap menu lihat daftar nilai, setelah itu bagian *View Daftar Nilai* akan memanggil *method index()* yang ada pada bagian *Nilai Controller*, selanjutnya *method show_daftar_nilai()* yang ada pada *Nilai Model* akan dipanggil oleh *controller*, *method* tersebut akan melakukan *query* pada *Table Nilai* di *database*. Bagian *Table Kelas* akan memberikan respon *success* pada *model* dan selanjutnya akan dikirim status *true* pada *controller* yang selanjutnya akan mengirimkan perintah untuk

51. Sequence Diagram Ubah Nilai Hasil Evaluasi

Sequence diagram ubah nilai hasil evaluasi merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas ubah nilai hasil evaluasi. Pemodelan sequence diagram fungsionalitas ubah nilai hasil evaluasi dapat dilihat pada Gambar 4.118.

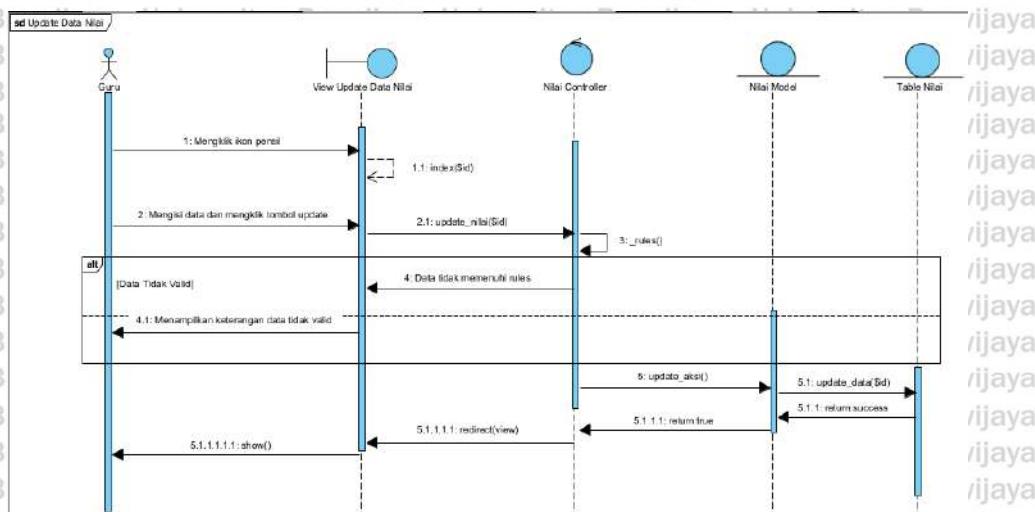


Gambar 4. 118 Sequence Diagram Ubah Nilai Hasil Evaluasi

Proses ubah nilai hasil evaluasi dimulai dengan aktor melakukan klik pada ikon pensil yang di daftar nilai, Selanjutnya akan ditampilkan halaman ubah data nilai oleh bagian *View Update Data Nilai* dengan memanggil *method* *index(\$id)*. Setelah aktor mengisi data yang diminta oleh *field* yang ada dan menekan tombol simpan. Selanjutnya *method* *update_nilai\$id* yang ada pada *controller* akan dipanggil, selanjutnya di bagian *controller* akan melakukan pengecekan apakah data yang dimasukkan oleh aktor valid (terisi semua, tidak ada duplikat), saat data dinyatakan tidak valid maka akan ditampilkan keterangan bahwa data yang dimasukkan tidak valid. Apabila data yang dimasukkan valid maka akan dieksekusi oleh *method* *update_aksi()* untuk dilakukan perubahan data sesuai dengan *id* pada tabel *Nilai* yang ada di *database*. Hasil *query* yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman daftar nilai.

52. Sequence Diagram Lihat Daftar Rapor

Sequence diagram lihat daftar rapor merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas validasi lihat daftar rapor. Pemodelan sequence diagram fungsionalitas validasi lihat daftar rapor dapat dilihat pada Gambar 4.119.

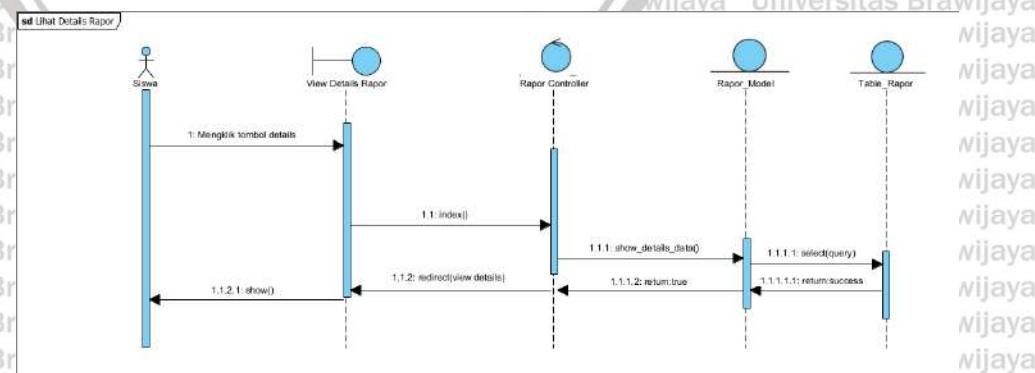


Gambar 4. 119 Sequence Diagram Lihat Daftar Rapor

Proses lihat daftar rapor dimulai dengan aktor melakukan klik terhadap menu lihat rapor, setelah itu bagian *View Daftar Nilai* akan memanggil *method index()* yang ada pada bagian *Rapor Controller*, selanjutnya *method show_daftar()* yang ada pada *Rapor Model* akan dipanggil oleh *controller*, *method* tersebut akan melakukan *query* pada *Table Rapor* di *database*. Bagian *Table Rapor* akan memberikan respon *success* pada *model* dan selanjutnya akan dikirim status *true* pada *controller* yang selanjutnya akan mengirimkan perintah *show()* untuk mengarahkan aktor pada *view* lihat rapor yang sudah menerima data dari *database*.

53. Sequence Diagram Detail Rapor

Sequence diagram detail rapor merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas detail rapor. Pemodelan *sequence diagram* fungsionalitas validasi detail rapor dapat dilihat pada Gambar 4. 120.



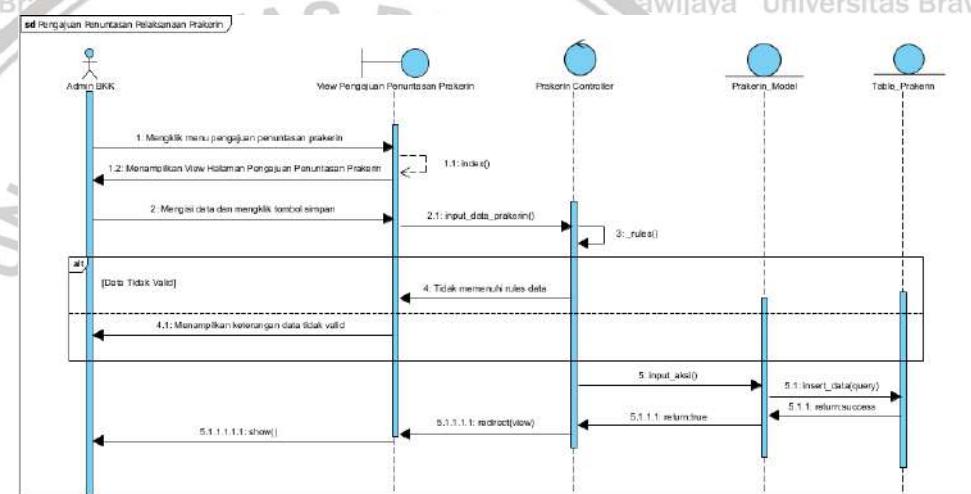
Gambar 4. 120 Sequence Diagram Detail Rapor

Proses lihat detail rapor dimulai dengan aktor melakukan klik terhadap tombol *details* yang ada pada daftar rapor, setelah itu bagian

View Details Rapor akan memanggil *method details()* yang ada pada bagian *Rapor Controller*, selanjutnya *method show_details_data()* yang ada pada *Rapor Model* akan dipanggil oleh *controller*, *method* tersebut akan melakukan *query* pada *Table Rapor* di *database*. Bagian *Table Rapor* akan memberikan respon *success* pada *model* dan selanjutnya akan dikirim status *true* pada *controller* yang selanjutnya akan mengirimkan perintah untuk mengarahkan aktor pada *view detail rapor* yang sudah menerima data dari *database*.

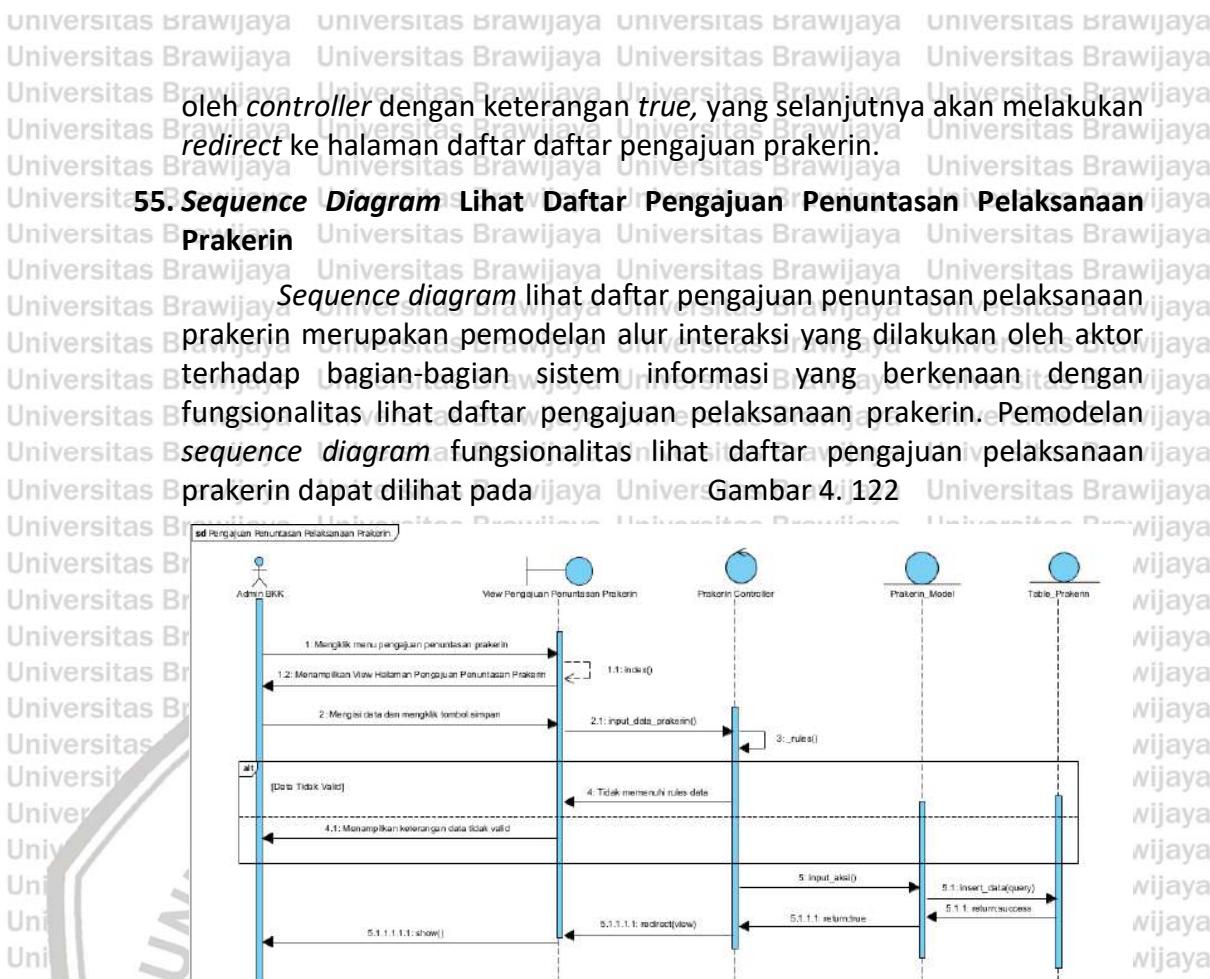
54. Sequence Diagram Pengajuan Penuntasan Pelaksanaan Prakerin

Sequence diagram pengajuan penuntasan pelaksanaan prakerin merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas pengajuan pelaksanaan prakerin. Pemodelan *sequence diagram* fungsionalitas pengajuan pelaksanaan prakerin dapat dilihat pada Gambar 4. 121.



Gambar 4. 121 *Sequence Diagram Pengajuan Penuntasan Prakerin*

Proses pengajuan penuntasan pelaksanaan prakerin dimulai dengan aktor melakukan klik pada tombol pengajuan yang ada diatas daftar pengajuan penuntasan prakerin, Selanjutnya akan ditampilkan halaman pengajuan penuntasan pelaksanaan prakerin oleh bagian *View Pengajuan Penuntasan Prakerin* dengan memanggil *method index()*. Setelah aktor mengisi data yang diminta oleh *field* yang ada dan menekan tombol simpan. Selanjutnya *method input_data_prakerin()* yang ada pada *controller* akan dipanggil, selanjutnya di bagian *controller* akan melakukan pengecekan apakah data yang dimasukkan oleh aktor valid (terisi semua, tidak ada duplikat), saat data dinyatakan tidak valid maka akan ditampilkan keterangan bahwa data yang dimasukkan tidak valid. Apabila data yang dimasukkan valid maka akan dieksekusi oleh *method input_aksi()* untuk ditambahkan pada tabel *Prakerin* yang ada di *database*. Hasil *query* yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return*

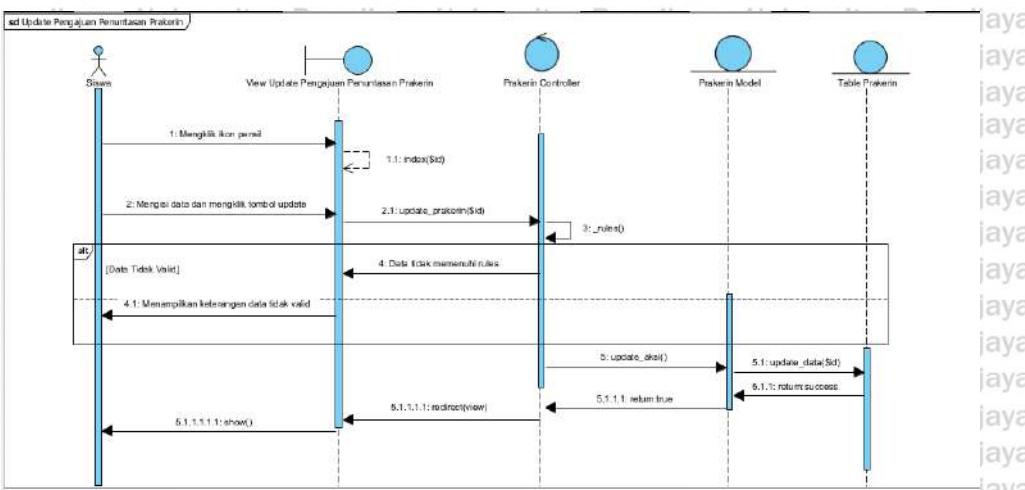


Gambar 4. 122 Sequence Diagram Lihat Daftar Pengajuan Prakerin

Proses lihat daftar pengajuan penuntasan pelaksaan prakerin dimulai dengan aktor melakukan klik terhadap menu prakerin, setelah itu bagian *View Prakerin* akan memanggil *method index_pengajuan()* yang ada pada bagian *Prakerin Controller*, selanjutnya *method show_data()* yang ada pada *Prakerin Model* akan dipanggil oleh *controller*, *method tersebut* akan melakukan *query* pada *Table Prakerin* di *database*. Bagian *Table Prakerin* akan memberikan respon *success* pada *model* dan selanjutnya akan dikirim *status true* pada *controller* yang selanjutnya akan mengirimkan perintah untuk mengarahkan aktor pada *view* lihat daftar pengajuan penuntasan prakerin yang sudah menerima data dari *database*.

56. Sequence Diagram Ubah Pengajuan Penuntasan Pelaksanaan Prakerin

Sequence diagram ubah pengajuan penuntasan pelaksaan prakerin merupakan pemodelan alur interaksi yang dilakukan oleh aktor terhadap bagian-bagian sistem informasi yang berkenaan dengan fungsionalitas ubah pengajuan pelaksanaan prakerin. Pemodelan *sequence diagram* fungsionalitas ubah pengajuan pelaksanaan prakerin dapat dilihat pada Gambar 4. 123.



Gambar 4. 123 Sequence Diagram Ubah Pengajuan Penuntasan Prakerin

Proses ubah pengajuan penuntasan prakerin dimulai dengan aktor melakukan klik pada ikon pensil yang di daftar pengajuan. Selanjutnya akan ditampilkan halaman ubah data pengajuan oleh bagian *View Update Pengajuan Penuntasan Pelaksanaan Prakerin* dengan memanggil *method index(\$id)*. Setelah aktor mengisi data yang diminta oleh *field* yang ada dan menekan tombol simpan. Selanjutnya *method update_prakerin(\$id)* yang ada pada *controller* akan dipanggil, selanjutnya di bagian *controller* akan melakukan pengecekan apakah data yang dimasukkan oleh aktor valid (terisi semua, tidak ada duplikat), saat data dinyatakan tidak valid maka akan ditampilkan keterangan bahwa data yang dimasukkan tidak valid. Apabila data yang dimasukkan valid maka akan dieksekusi oleh *method update_aksi()* untuk dilakukan perubahan data sesuai dengan *id* pada tabel Prakerin yang ada di *database*. Hasil *query* yang sukses akan mengirimkan respon *success* dan selanjutnya akan di *return* oleh *controller* dengan keterangan *true*, yang selanjutnya akan melakukan *redirect* ke halaman daftar nilai.

4.2.7 Perancangan Database

Database memegang peran yang penting dalam fungsiionalitas sistem informasi, sehingga diperlukan perancangan yang baik agar fungsionalitas yang diharapkan dalam sistem informasi dapat berjalan dengan baik. Implementasi desain *database* merujuk pada spesifikasi kebutuhan terhadap pengolahan data yang terlampir pada Lampiran A. Pada penelitian akan dilakukan adopsi teknologi baru yang mempengaruhi proses bisnis yang ada saat ini, maka data-data yang perlu diproses oleh *database* akan lebih banyak. Rancangan *database* yang diperlukan dalam pengembangan Sistem Informasik Akademik SMK Muhammadiyah 1 Kemlagi adalah sebagai berikut:

1. Tabel Pengguna (*Users*)

Tabel pengguna (*users*) berfungsi untuk menyimpan data yang digunakan oleh aktor untuk melakukan *login* pada sistem informasi. Detail dari struktur tabel pengguna administrator sekolah pada Tabel 4. 58.

Tabel 4. 58 Tabel Data Pengguna (Users)

| No | Nama Kolom | Tipe Data | Keterangan |
|----|------------|--------------|--|
| 1 | Id | Integer | Sebagai kunci utama (<i>primary key</i>) yang membuat data unik. |
| 2 | Username | Varchar(255) | Sebagai keterangan berupa nama yang digunakan untuk mengidentifikasi pengguna. |
| 3 | Password | Varchar(255) | Sebagai kata kunci rahasia untuk melakukan verifikasi <i>login</i> pengguna. |
| 4 | Level | Integer | Sebagai data untuk identifikasi hak akses oleh sistem informasi. |

2. Tabel Data Siswa

Tabel data siswa berfungsi untuk menyimpan data-data yang terkait dengan data diri siswa dan riwayat akademik siswa. Detail dari struktur tabel data siswa pada Tabel 4. 59.

Tabel 4. 59 Tabel Data Siswa

| No | Nama Kolom | Tipe Data | Keterangan |
|----|------------|--------------|---|
| 1 | Id | Integer | Sebagai tanda unik dari data siswa. |
| 2 | NISN | Integer | Sebagai kunci utama (<i>primary key</i>) yang membuat data pada tabel unik. |
| 3 | Nama | Varchar(255) | Sebagai field untuk menyimpan nama lengkap siswa. |

Tabel 4. 59 Tabel Data Siswa (Lanjutan)

| No | Nama Kolom | Tipe Data | Keterangan |
|----|---------------|---------------|---|
| 4 | Jenis Kelamin | Varchar (25) | Sebagai <i>field</i> untuk menyimpan jenis kelamin siswa. |
| 5 | Tempat lahir | Varchar(255) | Sebagai <i>field</i> untuk menyimpan tempat lahir siswa. |
| 6 | Tanggal lahir | Date | Sebagai <i>field</i> untuk menyimpan tanggal lahir siswa. |
| 7 | Alamat | Varchar(255) | Sebagai <i>field</i> untuk menyimpan alamat siswa. |
| 8 | Id_agama | Small integer | Sebagai <i>field</i> untuk menghubungkan dengan data pada tabel agama |
| 9 | anak_ke | Small integer | Sebagai <i>field</i> untuk menyimpan urutan kelahiran siswa dalam keluarga. |
| 10 | Id_status | Small integer | Sebagai <i>field</i> untuk menghubungkan dengan data pada tabel status. |
| 11 | Nomor telepon | Varchar(15) | Sebagai <i>field</i> untuk menyimpan nomor telepon siswa. |
| 12 | Foto | Varchar (255) | Sebagai <i>field</i> untuk menyimpan alamat penyimpanan foto siswa. |

3. Tabel Data Guru

Tabel data guru berfungsi untuk menyimpan data-data yang terkait dengan data diri guru dan data administratif lainnya. Detail dari struktur tabel data guru pada Tabel 4. 60.

Tabel 4. 60 Tabel Data Guru

| No | Nama Kolom | Tipe Data | Keterangan |
|----|---------------|---------------|---|
| 1 | NBM | Integer | Sebagai kunci utama (<i>primary key</i>) yang membuat data pada tabel unik. |
| 2 | Nama | Varchar(255) | Sebagai <i>field</i> untuk menyimpan nama lengkap guru beserta gelarnya. |
| 3 | Jenis_kelamin | Varchar(255) | Sebagai <i>field</i> untuk menyimpan jenis kelamin guru. |
| 4 | Tempat_lahir | Varchar(255) | Sebagai <i>field</i> untuk menyimpan tempat lahir guru. |
| 5 | Tanggal_lahir | Date | Sebagai <i>field</i> untuk menyimpan tanggal lahir guru. |
| 6 | Alamat | Varchar(255) | Sebagai <i>field</i> untuk menyimpan alamat guru. |
| 7 | Id_agama | Small integer | Sebagai <i>field</i> untuk menghubungkan dengan data pada tabel agama |
| 8 | Pend_terakhir | Varchar(10) | Sebagai <i>field</i> untuk menyimpan pendidikan terakhir yang guru. |
| 9 | Jabatan | Varchar(100) | Sebagai <i>field</i> untuk menyimpan jabatan dari guru. |
| 10 | Nomor_telepon | Varchar(15) | Sebagai <i>field</i> untuk menyimpan nomor telepon guru. |
| 11 | Status | Varchar (25) | Sebagai <i>field</i> untuk menyimpan status guru. |
| 12 | Foto | Varchar (255) | Sebagai <i>field</i> untuk menyimpan alamat penyimpanan (<i>url</i>) foto guru. |

4. Tabel Jurusan

Tabel data guru berfungsi untuk menyimpan data-data yang terkait dengan data informasi jurusan yang ada di SMK Muhammadiyah 1 Kemlagi. Detail dari struktur tabel jurusan pada Tabel 4. 61.

Tabel 4. 61 Tabel Data Jurusan

| No | Nama Kolom | Tipe Data | Keterangan |
|----|--------------|--------------|---|
| 1 | Id | Integer | Sebagai kunci utama (<i>primary key</i>) yang membuat data unik. |
| 2 | Nama_jurusan | Varchar(255) | Merupakan <i>field</i> untuk menyimpan data nama jurusan. |
| 3 | Kode_jurusan | Varchar(255) | Merupakan <i>field</i> untuk menyimpan data kode unik setiap jurusan. |

5. Tabel Agama

Tabel data agama berfungsi untuk menyimpan data-data yang terkait dengan agama. Detail dari struktur tabel agama pada Tabel 4. 62.

Tabel 4. 62 Tabel Data Agama

| No | Nama Kolom | Tipe Data | Keterangan |
|----|------------|-----------|--|
| 1 | Id | Integer | Sebagai kunci utama (<i>primary key</i>) yang membuat data unik. |
| 2 | Agama | Text(36) | Merupakan <i>field</i> untuk menyimpan nama agama. |

6. Tabel Kelas

Tabel data kelas berfungsi untuk menyimpan data-data yang terkait dengan data informasi kelas pada suatu tahun ajaran. Detail dari struktur tabel kelas pada Tabel 4. 63.

Tabel 4. 63 Tabel Data Kelas

| No | Nama Kolom | Tipe Data | Keterangan |
|----|------------|--------------|--|
| 1 | Id | Integer | Sebagai kunci utama (<i>primary key</i>) yang membuat data unik. |
| 2 | kode_kelas | Varchar(255) | Merupakan <i>field</i> untuk menyimpan data kode unik dari setiap kelas |
| 3 | nama_kelas | Varchar(255) | Merupakan <i>field</i> untuk menyimpan data nama kelas. |
| 4 | Id_jurusan | Integer | Sebagai <i>field</i> untuk menghubungkan dengan data pada tabel jurusan. |
| 5 | Wali_kelas | Integer | Sebagai <i>field</i> untuk menyimpan informasi wali kelas dengan menghubungkan data pada tabel guru. |

7. Tabel Data Riwayat Kelas

Tabel data riwayat kelas berfungsi untuk menyimpan data-data yang terkait dengan data riwayat kelas yang telah dilalui siswa. Detail dari struktur tabel riwayat kelas pada Tabel 4. 64.

Tabel 4. 64 Tabel Data Riwayat Kelas

| No | Nama Kolom | Tipe Data | Keterangan |
|----|------------|-------------|--|
| 1 | Id | Integer | Sebagai data unik dan kunci utama dari tabel data. |
| 2 | NISN | Varchar(50) | Sebagai <i>field</i> untuk menghubungkan dengan data pada tabel siswa. |

| No | Nama Kolom | Tipe Data | Keterangan |
|----|------------|-----------|---|
| 3 | Kelas_10 | Integer | Sebagai <i>field</i> untuk menyimpan informasi jenjang kelas 10 dengan menghubungkan data pada tabel kelas. |
| 4 | Kelas_11 | Integer | Sebagai <i>field</i> untuk menyimpan informasi jenjang kelas 11 dengan menghubungkan data pada tabel kelas. |
| 5 | Kelas_12 | Integer | Sebagai <i>field</i> untuk menyimpan informasi jenjang kelas 12 dengan menghubungkan data pada tabel kelas. |

8. Tabel Status Akademik

Tabel status akademik berfungsi untuk menyimpan data-data yang terkait dengan data status akademik siswa . Detail dari struktur tabel status akademik pada Tabel 4. 65.

Tabel 4. 65 Tabel Data Status Akademik

| No | Nama Kolom | Tipe Data | Keterangan |
|----|------------|-----------|--|
| 1 | Id | Integer | Sebagai kunci utama (<i>primary key</i>) yang membuat data unik. |
| 2 | Status | Text(36) | Merupakan <i>field</i> untuk menyimpan keterangan status siswa. |

9. Tabel Orang Tua

Tabel data orang tua berfungsi untuk menyimpan data-data yang terkait dengan informasi orang tua siswa. Detail dari struktur tabel orang tua pada Tabel 4. 66.

Tabel 4. 66 Tabel Data Orang Tua

| No | Nama Kolom | Tipe Data | Keterangan |
|----|----------------|--------------|--|
| 1 | NISN | Varchar(50) | Sebagai <i>field</i> untuk menghubungkan dengan data pada tabel siswa. |
| 2 | Nama_ayah | Varchar(255) | Sebagai <i>field</i> untuk menyimpan informasi nama ayah siswa. |
| 3 | Pekerjaan_ayah | Varchar(255) | Sebagai <i>field</i> untuk menyimpan informasi pekerjaan ayah siswa. |
| 4 | Telepon_ayah | varchar(15) | Sebagai <i>field</i> untuk menyimpan informasi telepon ayah siswa. |
| 5 | Nama_ibu | Varchar(255) | Sebagai <i>field</i> untuk menyimpan informasi nama ibu siswa. |
| 6 | Pekerjaan_ibu | Varchar(255) | Sebagai <i>field</i> untuk menyimpan informasi pekerjaan ibu siswa. |
| 7 | Telepon_ibu | Varchar(15) | Sebagai <i>field</i> untuk menyimpan informasi telepon ibu siswa. |

10. Tabel Sekolah Sebelumnya

Tabel sekolah sebelumnya berfungsi untuk menyimpan data-data yang terkait dengan data riwayat siswa diterima atau dipindahkan ke sekolah. Detail dari struktur tabel sekolah sebelumnya pada Tabel 4. 67.

Tabel 4. 67 Tabel Data Sekolah Sebelumnya

| No | Nama Kolom | Tipe Data | Keterangan |
|----|--------------|---------------|---|
| 1 | NISN | Varchar(50) | Sebagai <i>field</i> untuk menghubungkan dengan data pada tabel siswa. |
| 2 | Nama_sekolah | Varchar (255) | Sebagai <i>field</i> untuk menyimpan informasi nama sekolah sebelumnya. |

Tabel 4. 67 Tabel Data Sekolah Sebelumnya (Lanjutan)

| No | Nama Kolom | Tipe Data | Keterangan |
|----|------------------|--------------|--|
| 3 | Diterima_dikelas | Varchar (50) | Sebagai <i>field</i> untuk menyimpan informasi penerimaan siswa pada jenjang tertentu. |
| 4 | Tanggal | Date | Sebagai <i>field</i> untuk menyimpan tanggal penerimaan siswa. |

11. Tabel Mitra Industri

Tabel sekolah berfungsi untuk menyimpan data-data yang terkait dengan informasi industri yang bermitra dengan SMK Muhammadiyah 1 Kemlagi. Detail dari struktur tabel mitra industri pada Tabel 4. 68.

Tabel 4. 68 Tabel Data mitra Industri

| No | Nama Kolom | Tipe Data | Keterangan |
|----|----------------|---------------|--|
| 1 | Id | Integer | Sebagai kunci utama (<i>primary key</i>) yang membuat data unik. |
| 2 | Nama_industri | Varchar (100) | Sebagai <i>field</i> untuk menyimpan informasi nama industri. |
| 3 | Jenis Industri | Varchar (100) | Sebagai <i>field</i> untuk menyimpan informasi jenis industri. |
| 4 | Alamat | Varchar(100) | Sebagai <i>field</i> untuk menyimpan informasi alamat industri. |

12. Tabel Prakerin

Tabel prakerin berfungsi untuk menyimpan data-data yang terkait dengan kegiatan prakerin siswa. Detail dari struktur tabel prakerin pada Tabel 4. 69.

| No | Nama Kolom | Tipe Data | Keterangan |
|----|-------------|---------------|---|
| 1 | Id | Integer | Sebagai kunci utama (<i>primary key</i>) yang membuat data unik. |
| 2 | NISN | Varchar (25) | Sebagai <i>field</i> untuk menyimpan informasi NISN siswa. |
| 2 | Nama_tempat | Varchar (255) | Sebagai <i>field</i> untuk menyimpan informasi nama tempat pelaksanaan prakerin. |
| 6 | Alamat | Varchar(255) | Sebagai <i>field</i> untuk menyimpan informasi alamat tempat prakerin. |
| 7 | Deskripsi | Varchar(500) | Sebagai <i>field</i> untuk menyimpan informasi deskripsi tempat pelaksanaan prakerin. |
| 8 | Status | Varchar(50) | Sebagai <i>field</i> untuk menyimpan informasi status pelaksanaan prakerin siswa. |

13. Tabel Jadwal Pelajaran

Tabel jadwal pelajaran berfungsi untuk menyimpan data-data yang terkait dengan jadwal pelajaran setiap kelas. Detail dari struktur tabel jadwal pelajaran pada Tabel 4. 70.

Tabel 4. 70 Tabel Data Jadwal Pelajaran

| No | Nama Kolom | Tipe Data | Keterangan |
|----|------------|-------------|--|
| 1 | Id | Integer | Sebagai kunci utama (<i>primary key</i>) yang membuat data unik. |
| 2 | Hari | Varchar(10) | Sebagai <i>field</i> untuk menyimpan data hari. |

**Tabel 4. 70 Tabel Data Jadwal Pelajaran (Lanjutan)**

| No | Nama Kolom | Tipe Data | Keterangan |
|----|--------------|--------------|--|
| 3 | waktu | Varchar (50) | Sebagai <i>field</i> untuk menyimpan data waktu. |
| 4 | id_pelajaran | Int | Sebagai <i>field</i> untuk menghubungkan pada tabel mata pelajaran. |
| 5 | Id_kelas | Int | Sebagai <i>field</i> sita untuk menghubungkan data pada tabel kelas. |

14. Tabel Nilai

Tabel nilai berfungsi untuk menyimpan data-data yang terkait dengan nilai hasil evaluasi siswa. Detail dari struktur tabel nilai pada Tabel 4. 71.

Tabel 4. 71 Tabel Data Nilai

| No | Nama Kolom | Tipe Data | Keterangan |
|----|-------------------|--------------|--|
| 1 | Id | Integer | Sebagai <i>field</i> untuk menyimpan kunci utama yang menjadikan tabel unik. |
| 1 | NISN | Varchar(50) | Sebagai <i>field</i> untuk menghubungkan pada data di tabel siswa. |
| 2 | Id_pelajaran | Int | Sebagai <i>field</i> untuk menghubungkan pada data di tabel pelajaran. |
| 2 | Nilai_pengetahuan | Float | Sebagai <i>field</i> untuk menyimpan hasil nilai pengetahuan. |
| 3 | deskripsi_np | Varchar(500) | Sebagai <i>field</i> untuk menyimpan deskripsi pencapaian nilai pengetahuan. |
| 4 | Nilai_ketrampilan | Float | Sebagai <i>field</i> untuk menyimpan hasil nilai ketrampilan. |

Tabel 4. 71 Tabel Data Nilai (Lanjutan)

| No | Nama Kolom | Tipe Data | Keterangan |
|----|----------------|--------------|--|
| 5 | Deskripsi_nk | Varchar(500) | Sebagai <i>field</i> untuk menyimpan deskripsi pencapaian nilai ketrampilan. |
| 6 | Jenis_evaluasi | Varchar(50) | Sebagai <i>field</i> untuk menyimpan jenis evaluasi yang ditempuh. |

15. Tabel Kegiatan Akademik

Tabel kegiatan akademik berfungsi untuk menyimpan data-data yang terkait dengan kegiatan akademik tiap tahun ajaran baru. Detail dari struktur tabel kegiatan akademik pada Tabel 4. 72.

Tabel 4. 72 Tabel Data Kegiatan Akademik

| No | Nama Kolom | Tipe Data | Keterangan |
|----|---------------|--------------|--|
| 1 | id | Integer | Sebagai kunci utama (<i>primary key</i>) yang membuat data unik. |
| 2 | Nama_kegiatan | Varchar(255) | Sebagai <i>field</i> untuk menyimpan nama kegiatan. |
| 3 | Tanggal_mulai | Date | Sebagai <i>field</i> untuk menyimpan tanggal mulai kegiatan. |
| 4 | Tanggal_akhir | Date | Sebagai <i>field</i> untuk menyimpan tanggal akhir kegiatan. |

DBMS yang digunakan adalah MySQL yang merupakan basis data relasional, sehingga memungkinkan tabel-tabel data memiliki relasi satu dengan yang lain. Tahap normalisasi tabel basis data dilakukan untuk mencegah terjadinya anomali data pada basis data. Bentuk akhir dari relasi antar tabel pada basis data sesuai dengan normalisasi 3NF. Tahap normalisasi *database* adalah sebagai berikut:

A. Normalisasi 1NF

Pada tahap normalisasi bentuk pertama dilakukan pembentukan tabel data sehingga setiap tabel hanya memiliki nilai tunggal dan tidak

ada nilai atau atribut yang berulang-ulang. Tabel-tabel yang dinormalisasi pada tahap ini meliputi tabel siswa, tabel mata pelajaran, tabel jadwal, tabel kelas, tabel kegiatan akademik, tabel prakerin dan tabel nilai.

B. Normalisasi 2NF

Pada tahap normalisasi bentuk kedua dilakukan beberapa penyesuaian berupa penghapusan *functional dependency* pada beberapa tabel yang meliputi:

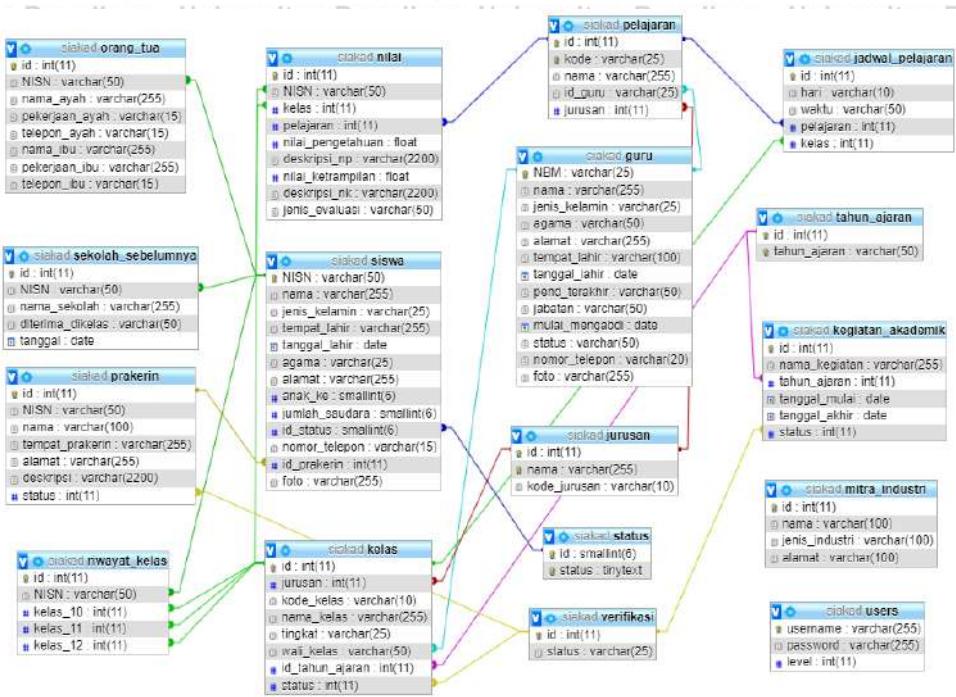
- Atribut pada tabel siswa dipecah dalam beberapa tabel yang meliputi tabel riwayat kelas, tabel orang tua, tabel sekolah sebelumnya, dan tabel nilai. Dimana masing-masing tabel tersebut dipecah berdasarkan *primary key* NISN siswa.
- Atribut pada tabel kegiatan akademik dipecah pada tabel tahun ajaran.

C. Normalisasi 3NF

Pada tahap normalisasi bentuk ketiga dilakukan beberapa penyesuaian berupa penghapusan *transitive dependency* pada beberapa tabel yang meliputi:

- Atribut pada tabel kelas dipecah dalam beberapa tabel yang meliputi tabel jurusan, tabel jadwal pelajaran, tabel guru dan tabel tahun ajaran.
- Atribut pada tabel pelajaran dipecah dalam beberapa tabel yang meliputi tabel jurusan, tabel tahun ajaran dan tabel guru.
- Atribut pada tabel jadwal pelajaran dipecah dalam beberapa tabel meliputi tabel pelajaran dan tabel kelas.

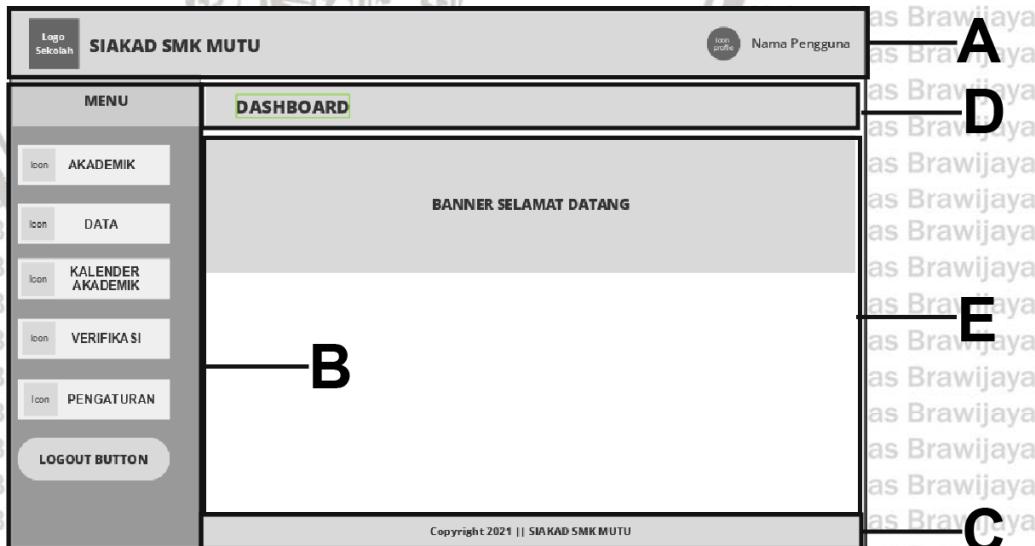
Dengan spesifikasi tabel-tabel yang telah didefinisikan maka diperoleh *physical data model* seperti pada Gambar 4. 124.



Gambar 4. 124 Physical Data Model Database Siakad

4.2.8 Perancangan Antarmuka

Perancangan antarmuka merupakan gambaran kasar dari antarmuka yang akan diimplementasikan pada sistem. Setiap halaman pada sistem informasi terdiri dari beberapa bagian seperti pada Gambar 4. 125.



Gambar 4. 125 Antarmuka Sistem informasi Default

- A. *Navbar* Sistem Informasi sebagai tempat menampilkan keterangan nama sistem informasi, logo sekolah dan profil pengguna yang saat ini mengakses sistem informasi.
- B. *Sidebar* sistem informasi sebagai bagian penempatan menu-menu yang ada pada sistem informasi, serta tombol untuk melakukan *logout* dari sistem.
- C. *Footer* sistem informasi sebagai penegas batas bawah dari sistem informasi dan bagian untuk menempatkan informasi yang berkaitan dengan pengembang sistem informasi.
- D. *Breadcrumb* sistem informasi berfungsi untuk menempatkan informasi mengenai judul halaman yang saat ini diakses oleh aktor.
- E. *Content* halaman berfungsi untuk menempatkan konten-konten dinamis sesuai dengan halaman yang diakses oleh aktor.

Rasio yang digunakan pada rancangan antarmuka berikut menerapkan rasio umum layar komputer, yaitu 1280 piksel banding 800 piksel. Berikut ini merupakan rancangan antarmuka dari sistem informasi akademik SMK Muhammadiyah 1 Kemlagi:

1. Halaman Login

Halaman *login* terdiri dari beberapa komponen pembentuk antarmuka, meliputi *Text* untuk menampilkan keterangan *field*, *Textbox* untuk menempatkan *username* dan *password*, *Image* untuk menampilkan gambar logo sekolah, dan *Button* untuk mengarahkan aktor pada halaman *dashboard*. Rancangan antarmuka halaman *login* pada Gambar 4. 126.

The image shows a wireframe design of a login page. At the top center, it says "SIAKAD SMK MUTU". Below that is a placeholder for "LOGO SMK MUTU 150px X 160 px". There are two input fields: one labeled "Masukkan username" and another labeled "Masukkan Password". Below these is a "Login" button. At the bottom, there is a link "Informasi CP Admin Sistem".

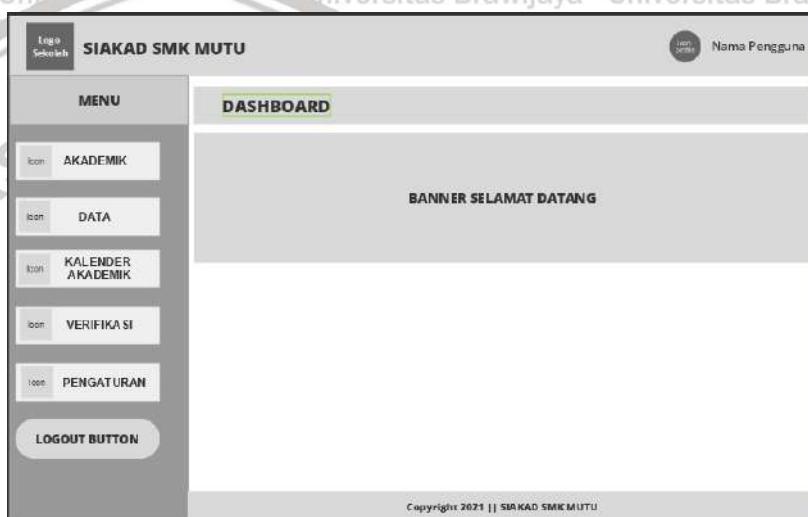
Gambar 4. 126 Rancangan Antarmuka Halaman Login

2. Halaman Dashboard

Karena Sistem Informasi menerapkan level akses maka tampilan halaman *dashboard* akan berbeda-beda menyesuaikan dengan hak akses aktor yang melakukan pengaksesan sistem informasi. Berikut merupakan daftar rancangan antarmuka halaman dashboard dari setiap aktor yang ada:

A. Dashboard Admin Sistem

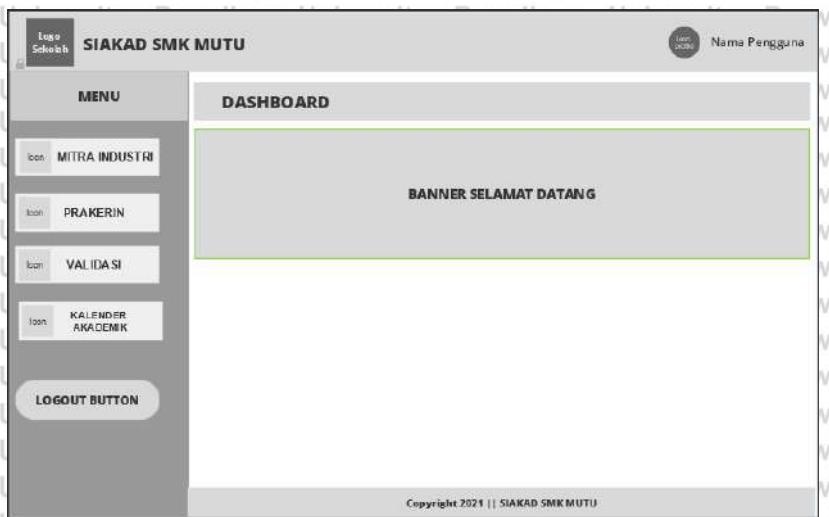
Menu yang dapat diakses pada *dashboard* admin sistem meliputi, menu akademik, menu data, menu kalender akademik, menu verifikasi, menu pengaturan dan tombol untuk logout. Rancangan tampilan *dashboard* admin sistem dapat dilihat pada Gambar 4. 127.



Gambar 4. 127 Rancangan Antarmuka Halaman *Dashboard* Admin Sistem

B. Dashboard Admin BKK

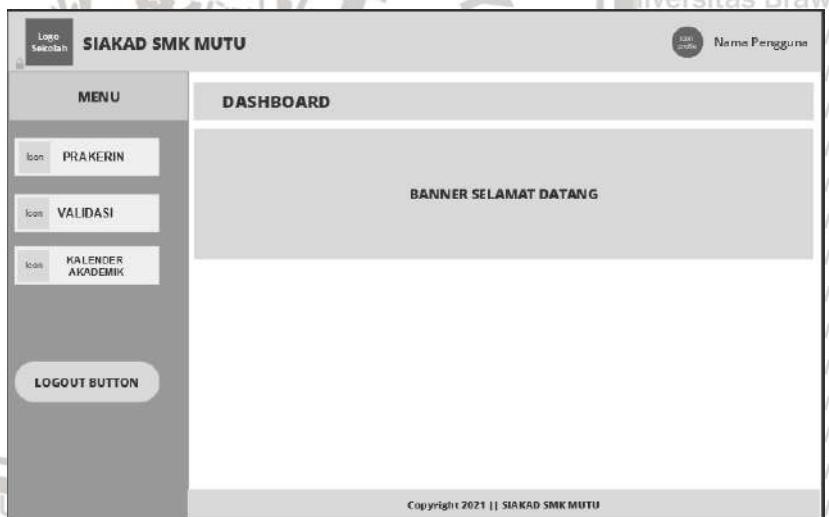
Menu yang dapat diakses pada *dashboard* admin BKK meliputi, menu mitra industri, menu prakerin, menu kalender akademik, menu validasi, dan tombol untuk logout. Rancangan tampilan *dashboard* admin BKK dapat dilihat pada Gambar 4. 128.



Gambar 4. 128 Rancangan Antarmuka Halaman *Dashboard* Admin BKK

C. *Dashboard* Waka Kurikulum

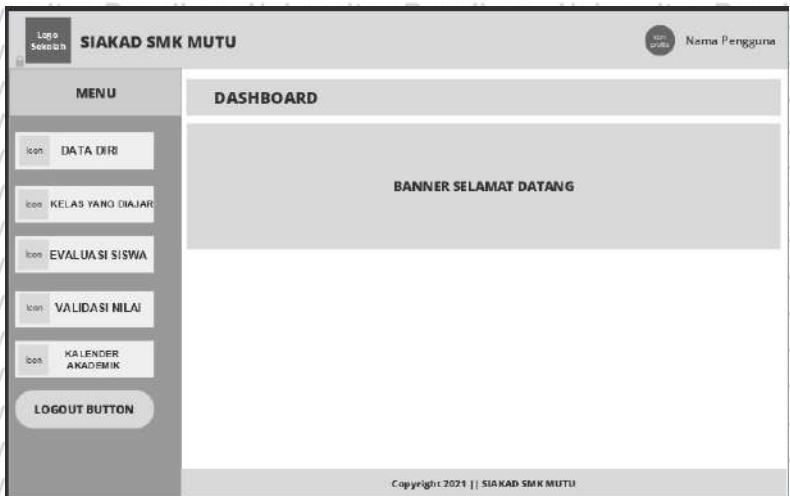
Menu yang dapat diakses pada *dashboard* waka kurikulum meliputi, menu prakerin, menu kalender akademik, menu validasi, dan tombol untuk logout. Rancangan tampilan *dashboard* waka kurikulum dapat dilihat pada Gambar 4. 129.



Gambar 4. 129 Rancangan Antarmuka Halaman *Dashboard* Waka Kurikulum

D. *Dashboard* Guru

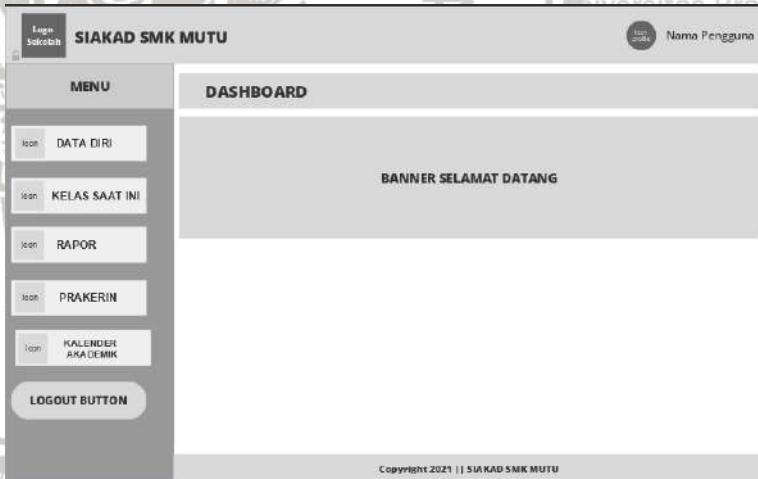
Menu yang dapat diakses pada *dashboard* guru meliputi, menu guru, menu kelas yang diajar, menu evaluasi siswa, menu validasi nilai, menu kalender akademik, dan tombol untuk logout. Rancangan tampilan *dashboard* admin BKK dapat dilihat pada Gambar 4. 130.



Gambar 4. 130 Rancangan Antarmuka Halaman **Dashboard Guru**

E. *Dashboard Siswa*

Menu yang dapat diakses pada *dashboard* siswa meliputi, menu data diri, menu kelas saat ini, menu rapor, menu prakerin, menu kalender akademik, dan tombol untuk logout. Rancangan tampilan *dashboard* siswa dapat dilihat pada Gambar 4. 131.



Gambar 4. 131 Rancangan Antarmuka Halaman **Dashboard Siswa**

3. Halaman Lihat Kalender Akademik

Halaman lihat kalender akademik memiliki tampilan yang sama untuk semua aktor yang ada di sistem informasi, rancangan antarmuka halaman lihat kalender akademik dapat dilihat pada Gambar 4. 132.

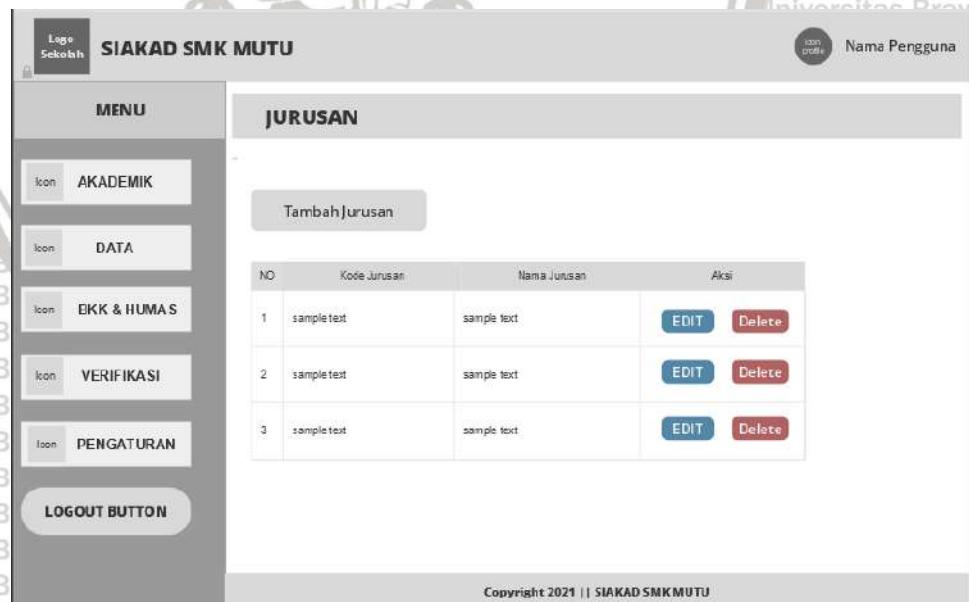


Gambar 4. 132 Rancangan Antarmuka Halaman Lihat Kalender Akademik

Komponen yang ada pada bagian *content* di halaman lihat kalender akademik berisi tabel yang memiliki 4 kolom, yaitu kolom nomor, kolom nama kegiatan, kolom tanggal mulai dan kolom tanggal selesai kegiatan.

4. Halaman Daftar Jurusan

Halaman daftar jurusan menampilkan daftar jurusan yang ada saat ini dalam bentuk tabel, rancangan antarmuka halaman lihat daftar jurusan dapat dilihat pada Gambar 4. 133.



Gambar 4. 133 Rancangan Antarmuka Halaman Daftar Jurusan

Komponen-komponen yang ada pada bagian *content* di halaman daftar jurusan meliputi tabel dengan empat kolom, yaitu kolom nomor, kolom kode jurusan, kolom nama jurusan dan kolom aksi. Selain itu juga

The screenshot shows the 'TAMBAH JURUSAN' (Add Program) screen of the SIAKAD SMK MUTU application. On the left, there is a vertical sidebar menu with five items: 'AKADEMIK' (selected), 'DATA', 'BKK & HUMAS', 'VERIFIKASI', and 'PENGATURAN'. Below the menu is a 'LOGOUT BUTTON'. The main content area is titled 'TAMBAH JURUSAN'. It contains two input fields: 'KODE JURUSAN' with placeholder text 'Masukkan Kode Jurusan' and 'NAMA JURUSAN' with placeholder text 'Masukkan Nama Jurusan'. Each field has a 'Simpan' button below it. In the top right corner of the main area, there is a user profile icon and the text 'Nama Pengguna'. At the bottom right of the main area, it says 'Copyright 2021 || SIAKAD SMK MUTU'.

Gambar 4. 134 Rancangan Antarmuka Halaman Tambah Data Jurusan

6. Halaman Ubah Jurusan

Halaman ubah jurusan menampilkan *field* yang sama dengan yang ada di halaman tambah jurusan, yang berbeda adalah pada halaman ubah jurusan *field* yang ada sudah terdapat nilai dengan data yang ada saat ini. *Button update* digunakan untuk menjalankan fungsi *update* data jurusan ke *database*. Rancangan antarmuka halaman ubah jurusan dapat dilihat pada Gambar 4. 135.



Gambar 4. 135 Rancangan Antarmuka Halaman *Update Data Jurusan*

7. Halaman Hapus Jurusan

Halaman hapus jurusan menampilkan keterangan berupa teks konfirmasi dan dua *button* fungsi. *Button* "Ya" digunakan untuk menjalankan fungsi *delete* data jurusan di *database*. *Button* "Tidak" digunakan untuk membatalkan fungsi hapus dan mengembalikan aktor pada halaman awal. Rancangan antarmuka halaman hapus jurusan dapat dilihat pada Gambar 4. 136.



Gambar 4. 136 Rancangan Antarmuka Halaman Hapus Data Jurusan

8. Halaman Data Kelas



Gambar 4. 137 Rancangan Antarmuka Halaman Daftar Kelas

Komponen-komponen yang ada pada bagian *content* di halaman kelas meliputi tabel dengan empat kolom, yaitu kolom nomor, kolom kode kelas, kolom nama kelas dan kolom aksi. Selain itu juga terdapat *button* untuk menambah jurusan saat ini, *button* untuk melakukan perubahan pada data jurusan, *button* untuk melihat details informasi kelas dan *button* untuk menghapus data kelas.

9. Halaman *Details* Kelas

Halaman *details* kelas, menampilkan informasi-informasi yang relevan dengan kelas, seperti kode kelas, nama kelas, wali kelas, tombol untuk melihat jadwal pelajaran kelas, dan daftar siswa kelas dalam bentuk tabel. Rancangan antarmuka dari halaman *details* kelas dapat dilihat pada Gambar 4. 138.

Logo Sekolah SIAKAD SMK MUTU

MENU

- Icon AKADEMIK
- Icon DATA
- Icon BKK & HUMAS
- Icon VERIFIKASI
- Icon PENGATURAN

Kode Kelas : X1-FRM-2

Nama Kelas : Sebelas Farmasi B

Wali Kelas : Oemar Bakri S.Pd

Lihat Jadwal

Daftar Siswa

| NO. AB SEN | Nama |
|------------|-------------|
| 1 | sample text |
| 2 | sample text |

Copyright 2021 || SIAKAD SMK MUTU

Gambar 4. 138 Rancangan Antarmuka Halaman Detail Kelas

10. Halaman Tambah Kelas

Halaman tambah kelas menampilkan *field* yang digunakan untuk memasukkan data-data kelas yang meliputi kode kelas dan nama kelas, untuk pemilihan wali kelas menggunakan komponen *dropdown* yang berisi daftar guru yang tersimpan di *database*. Tombol simpan digunakan untuk men jalankan fungsi simpan data kelas ke *database*. Rancangan antarmuka halaman tambah kelas dapat dilihat pada Gambar 4. 139.

Logo Sekolah SIAKAD SMK MUTU

MENU

- Icon AKADEMIK
- Icon DATA
- Icon BKK & HUMAS
- Icon VERIFIKASI
- Icon PENGATURAN

TAMBAH KELAS

KODE KELAS

Masukkan Kode Kelas

NAMA KELAS

Masukkan Nama Kelas

WALI KELAS

Pilih Wali Kelas

Simpan

Copyright 2021 || SIAKAD SMK MUTU

Gambar 4. 139 Rancangan Antarmuka Halaman Tambah Data Kelas

11. Halaman Ubah Kelas

Halaman ubah kelas menampilkan *field* yang sama dengan yang ada di halaman tambah kelas, yang berbeda adalah pada halaman ubah jurusan

field yang ada sudah terdapat nilai dengan data yang ada saat ini. *Button update* digunakan untuk menjalankan fungsi *update* data kelas ke *database*. rancangan antarmuka halaman ubah kelas dapat dilihat pada Gambar 4. 140.

Gambar 4. 140 Rancangan Antarmuka Halaman Ubah Data Kelas

12. Halaman Hapus Kelas

Halaman hapus kelas menampilkan keterangan berupa teks konfirmasi dan dua *button* fungsi. *Button* "Ya" digunakan untuk menjalankan fungsi *delete* data kelas di *database*. *Button* "Tidak" digunakan untuk membatalkan fungsi hapus dan mengembalikan aktor pada halaman awal. rancangan antarmuka halaman hapus kelas dapat dilihat pada Gambar 4. 141.

Gambar 4. 141 Rancangan Antarmuka Halaman Hapus Data Kelas

13. Halaman Mata Pelajaran

Halaman mata pelajaran menampilkan daftar mata pelajaran yang ada saat ini dalam bentuk tabel. Juga terdapat komponen *dropdown* yang berisi daftar jurusan yang ada di *database*. Rancangan antarmuka halaman mata pelajaran dapat dilihat pada Gambar 4. 142.

| NO | Kode Mata Pelajaran | Nama Mata Pelajaran | Guru Pengampu | Jurusan | Aksi |
|----|---------------------|---------------------|---------------|-------------|---|
| 1 | sample text | sample text | sample text | sample text | <button>EDIT</button> <button>Delete</button> |
| 2 | sample text | sample text | sample text | sample text | <button>EDIT</button> <button>Delete</button> |
| 3 | sample text | sample text | sample text | sample text | <button>EDIT</button> <button>Delete</button> |

Gambar 4. 142 Rancangan Antarmuka Halaman Daftar Mata Pelajaran

Komponen-komponen yang ada pada bagian *content* di halaman mata pelajaran meliputi tabel dengan enam kolom, yaitu kolom nomor, kolom kode mata pelajaran, kolom nama mata pelajaran, kolom guru pengampu, kolom jurusan, dan kolom aksi. Selain itu juga terdapat *button* untuk menambah mapel saat ini, *button* untuk melakukan pencarian mata pelajaran berdasarkan, *button* untuk mengubah data mata pelajaran dan *button* untuk menghapus data mata pelajaran.

14. Halaman Tambah Mata Pelajaran

Halaman tambah mata pelajaran menampilkan *field* yang digunakan untuk memasukkan data-data mata pelajaran yang meliputi kode mata pelajaran dan nama mata pelajaran, untuk pemilihan guru pengampu dan jurusan menggunakan komponen *dropdown* yang berisi daftar guru dan jurusan yang tersimpan di *database*. Tombol simpan digunakan untuk menjalankan fungsi simpan data mata pelajaran ke *database*. Rancangan antarmuka halaman tambah mata pelajaran dapat dilihat pada Gambar 4. 143.

Copyright 2021 || SIAKAD SMK MUTU

Gambar 4. 143 Rancangan Antarmuka Halaman Tambah Mata Pelajaran

15. Halaman Ubah Mata Pelajaran

Halaman ubah mata pelajaran menampilkan *field* yang sama dengan yang ada di halaman tambah mata pelajaran, yang berbeda adalah pada halaman ubah mata pelajaran adalah *field* yang ada sudah terdapat nilai dengan data yang ada saat ini. *Button update* digunakan untuk menjalankan fungsi *update* data mata pelajaran ke *database*. Rancangan antarmuka halaman ubah mata pelajaran dapat dilihat pada Gambar 4. 144.

Copyright 2021 || SIAKAD SMK MUTU

Gambar 4. 144 Rancangan Antarmuka Halaman Ubah Mata Pelajaran

16. Halaman Hapus Mata Pelajaran

Halaman hapus mata pelajaran menampilkan keterangan berupa teks konfirmasi dan dua button fungsi. Button “Ya” digunakan untuk menjalankan fungsi *delete* data mata pelajaran di *database*. Button “Tidak” digunakan untuk membatalkan fungsi hapus dan mengembalikan aktor pada halaman awal. Rancangan antarmuka halaman hapus mata pelajaran dapat dilihat pada Gambar 4. 145.



Gambar 4. 145 Rancangan Antarmuka Halaman Hapus Data Mata Pelajaran

17. Halaman Daftar Guru

Halaman data guru menampilkan daftar guru yang ada saat ini dalam bentuk tabel, rancangan antarmuka halaman guru dapat dilihat pada Gambar 4. 146.

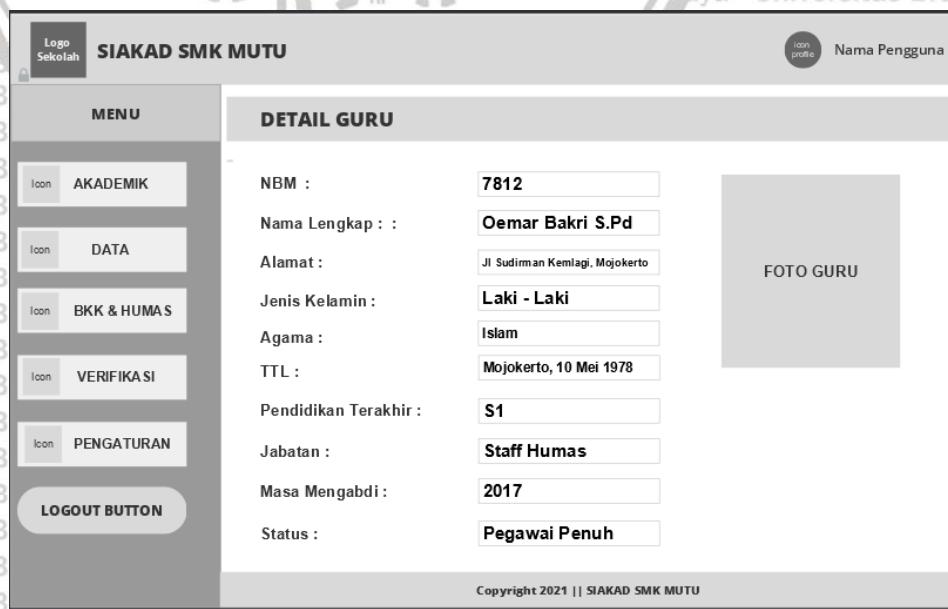


Gambar 4. 146 Rancangan Antarmuka Halaman Daftar Guru

Komponen-komponen yang ada pada bagian *content* di halaman guru meliputi tabel dengan lima kolom, yaitu kolom nomor, kolom NBM, kolom nama guru, jenis kelamin dan kolom aksi. Selain itu juga terdapat button untuk menambah data guru saat ini, button untuk melakukan perubahan pada data guru, button untuk melihat details informasi guru dan button untuk menghapus data guru.

18. Halaman Detail Guru

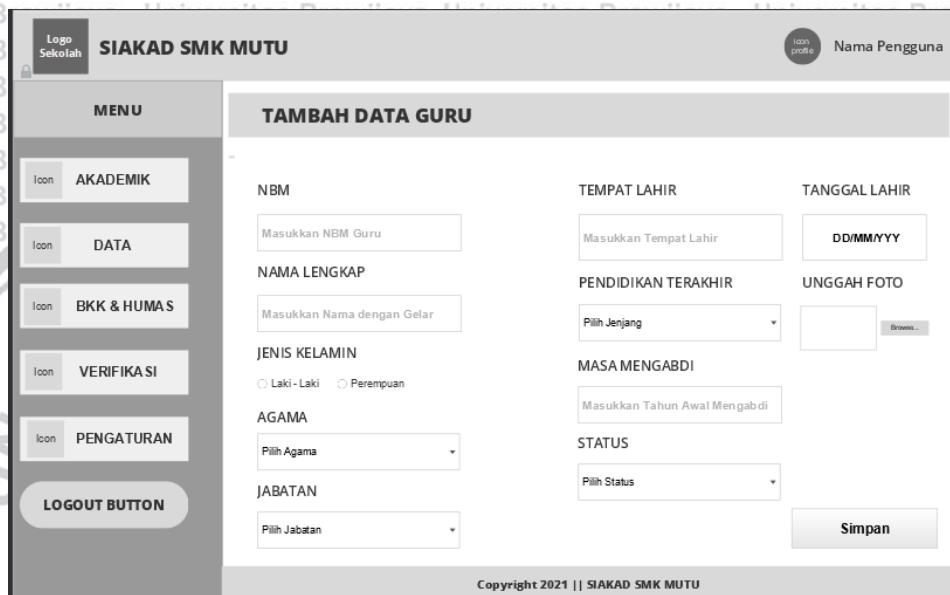
Halaman detail guru menampilkan informasi-informasi mengenai data diri guru, seperti NBM, nama lengkap, alamat, jenis kelamin, agama, tempat tanggal lahir, pendidikan terakhir, jabatan, masa mengabdi, status dan foto guru. Rancangan antarmuka dari halaman detail guru dapat dilihat pada Gambar 4. 147.



19. Halaman Tambah Guru

Halaman tambah guru menampilkan *field* yang digunakan untuk memasukkan data diri guru yang meliputi NBM, nama lengkap, alamat, jenis kelamin, agama, tempat tanggal lahir, pendidikan terakhir, jabatan, masa mengabdi, status dan tombol unggah foto. Tombol simpan digunakan untuk menjalankan fungsi simpan data guru ke *database*. Rancangan antarmuka halaman tambah guru dapat dilihat pada Gambar 4.

148.



Gambar 4.148 menunjukkan rancangan antarmuka halaman tambah data guru. Halaman ini berjudul "SIAKAD SMK MUTU" di bagian atas. Di sisi kiri ada sidebar dengan menu: Logo Sekolah, MENU (AKADEMIK, DATA, BKK & HUMAS, VERIFIKASI, PENGATURAN), dan LOGOUT BUTTON. Di bagian tengah ada judul "TAMBAH DATA GURU". Formulir tambah data guru mencakup beberapa bagian: NBM (Masukkan NBM Guru), TEMPAT LAHIR (Masukkan Tempat Lahir) dan TANGGAL LAHIR (DD/MM/YYYY), NAMA LENGKAP (Masukkan Nama dengan Gelar), PENDIDIKAN TERAKHIR (Pilih Jenjang), PENDIDIKAN TERAKHIR (UNGGAH FOTO), JENIS KELAMIN (Laki-Laki / Perempuan), AGAMA (Pilih Agama), MASA MENGABDI (Masukkan Tahun Awal Mengabdi), JABATAN (Pilih Jabatan), STATUS (Pilih Status), dan tombol Simpan. Di bagian bawah formulir terdapat tanda Copyright 2021 || SIAKAD SMK MUTU.

Gambar 4. 148 Rancangan Antarmuka Halaman Tambah Data Guru

20. Halaman Update Data Guru

Halaman *update* data guru menampilkan *field* yang sama dengan yang ada di halaman tambah data guru, yang berbeda adalah pada halaman *update* data guru, *field* yang ada sudah terisi nilai dengan data saat ini. *Button update* digunakan untuk menjalankan fungsi *update* data guru ke *database*. Rancangan antarmuka halaman *update* data guru dapat dilihat pada Gambar 4.149.

Gambar 4. 149 Rancangan Antarmuka Halaman *Update Data Guru*

21. Halaman Hapus Data Guru

Halaman hapus data guru menampilkan keterangan berupa teks konfirmasi dan dua *button* fungsi. *Button* "Ya" digunakan untuk menjalankan fungsi *delete* data guru di *database*. *Button* "Tidak" digunakan untuk membatalkan fungsi hapus dan mengembalikan aktor pada halaman awal. Rancangan antarmuka halaman hapus data guru dapat dilihat pada Gambar 4. 150.

Gambar 4. 150 Rancangan Antarmuka Halaman Hapus Data Guru

22. Halaman Daftar Siswa

Halaman data siswa menampilkan daftar guru yang ada saat ini dalam bentuk tabel. Komponen-komponen yang ada pada bagian *content*



Gambar 4. 151 Rancangan Antarmuka Halaman Daftar Siswa

23. Halaman Detail Siswa

Rancangan halaman detail siswa bagian satu pada Gambar 4. 152 menampilkan informasi-informasi mengenai data diri siswa dan status siswa, data-data tersebut meliputi seperti NISN, nama lengkap, alamat, jenis kelamin, agama, tempat tanggal lahir, nomor telepon, jurusan , anak ke, status akademik, status prakerin dan foto siswa.

Gambar 4. 152 Rancangan Antarmuka Halaman Detail Siswa Bagian Satu

Rancangan halaman detail siswa bagian dua pada Gambar 4.153 menampilkan informasi-informasi mengenai data diri siswa dan riwayat studi siswa, data-data tersebut meliputi daftar riwayat kelas, sekolah sebelumnya, dan informasi orang tua.



Logo Sekolah SIAKAD SMK MUTU

Nama Pengguna

MENU

AKADEMIK

DATA

BKK & HUMAS

VERIFIKASI

PENGATURAN

DETAIL SISWA

RIWAYAT KELAS

| | |
|------------|-----------|
| Kelas X: | X-TEI-A |
| Kelas XI: | XI-TEI-B |
| Kelas XII: | XII-TEI-A |

SEKOLAH SEBELUMNYA

| | |
|---------------------|------------|
| Nama Sekolah: | SMP Soeko |
| Diterima di Kelas: | X |
| Tanggal Penerimaan: | 14-02-2017 |

INFORMASI ORANG TUA

| | |
|---------------------|-----------------|
| Nama Ayah: | Glyono |
| Nama Ibu: | Nurul |
| Pekerjaan Ayah: | Tukang Cat |
| Pekerjaan Ibu: | RT |
| Nomor Telepon Ayah: | 0899-5070-8066 |
| Nomor Telepon Ibu: | - |
| Alamat orang tuas: | Mengelo Selatan |

LOGOUT BUTTON

Copyright 2021 || SIAKAD SMK MUTU

Gambar 4. 153 Rancangan Antarmuka Halaman Detail Siswa Bagian Dua

24. Halaman Tambah Siswa

Halaman tambah siswa menampilkan *field* yang digunakan untuk memasukkan data diri siswa yang meliputi NISN, nama lengkap, alamat, jenis kelamin, agama, tempat tanggal lahir, jurusan, jabatan, informasi anak ke, dan tombol unggah foto. Rancangan antarmuka halaman tambah siswa bagian pertama dapat dilihat pada Gambar 4. 154.

The screenshot shows a user interface for 'SIAKAD SMK MUTU'. On the left is a vertical menu bar with icons for AKADEMIK, DATA, BKK & HUMAS, VERIFIKASI, and PENGATURAN, along with a LOGOUT BUTTON. The main area is titled 'TAMBAH DATA SISWA' and contains several input fields: 'NISN' (Masukkan NISN SISWA), 'TEMPAT LAHIR' (Masukkan Tempat Lahir), 'NAMA LENGKAP' (Masukkan Nama Siswa), 'TANGGAL LAHIR' (DD/MM/YYYY), 'JENIS KELAMIN' (radio buttons for Laki - Laki and Perempuan), 'AGAMA' (dropdown menu for Pilih Agama), 'ANAK KE' (dropdown menu for Masukkan Informasi siswa anak no), 'UNGGAH FOTO' (button for Cari di Komputer), and 'JURUSAN' (dropdown menu for Pilih Jurusan). At the bottom right is a copyright notice: 'Copyright 2021 || SIAKAD SMK MUTU'.

Gambar 4. 154 Rancangan Antarmuka Halaman Tambah Data Siswa Bagian Satu

Pada halaman tambah siswa bagian dua berisi informasi kontak siswa, informasi sekolah siswa sebelumnya, riwayat kelas dan informasi orang tua. Tombol simpan digunakan untuk menjalankan fungsi simpan data siswa ke *database*. Rancangan antarmuka halaman tambah data siswa bagian dua dapat dilihat pada Gambar 4. 155.

This screenshot shows the second part of the 'TAMBAH DATA SISWA' form. It includes fields for 'NO NOMOR TELEPON' (Masukkan NBM Guru), 'RIWAYAT KELAS' (dropdown menus for Kelas X, XI, and XII each with sample text), 'SEKOLAH SEBELUMNYA' (dropdown menus for Nama Sekolah, Diterima di Kelas, and Tanggal Penerimaan, each with sample text), and 'INFORMASI ORANG TUA' (table with columns for Nama Ayah, Nama Ibu, Pekerjaan Ayah, Pekerjaan Ibu, Nomor Telepon Ayah, Nomor Telepon Ibu, and Alamat orang tua, all with sample text). A 'Simpan' button is located at the bottom right. The footer displays the copyright notice: 'Copyright 2021 || SIAKAD SMK MUTU'.

Gambar 4. 155 Rancangan Antarmuka Halaman Tambah Data Siswa Bagian Dua

25. Halaman Update Siswa

Halaman ubah data siswa menampilkan *field* yang sama dengan yang ada di halaman tambah data siswa, yang berbeda adalah pada

halaman *update* data siswa *field* yang ada sudah terisi nilai dengan data saat ini. *Button* perbarui digunakan untuk menjalankan fungsi *update* data siswa ke *database*. Rancangan antarmuka halaman *update* data siswa dapat dilihat pada Gambar 4. 156 dan pada Gambar 4. 157.

Gambar 4. 156 Rancangan Antarmuka Halaman *Update* Data Siswa Bagian Satu

Gambar 4. 157 Rancangan Antarmuka Halaman *Update* Data Siswa Bagian Dua

26. Halaman Hapus Siswa

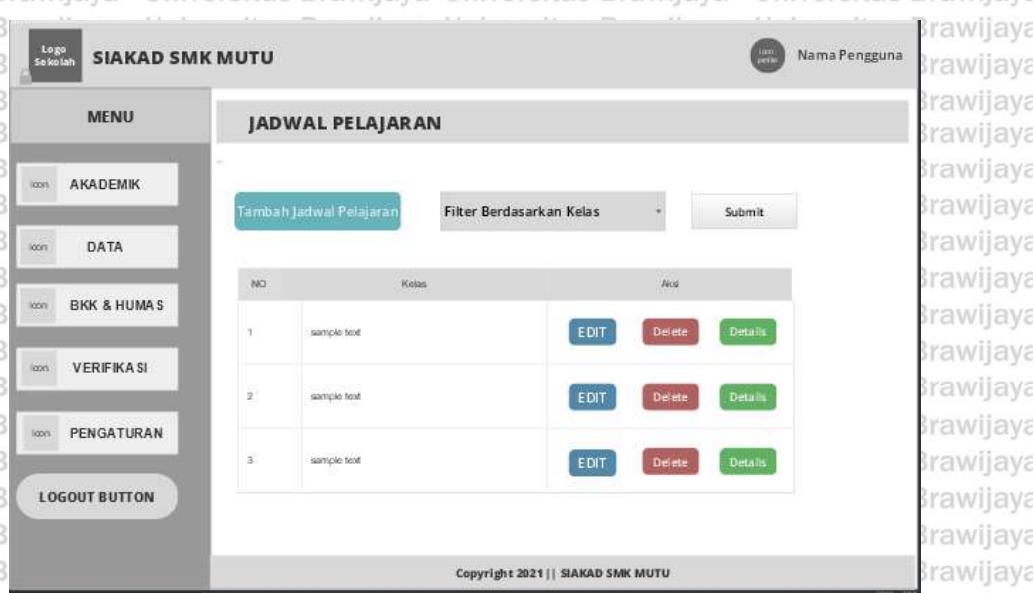
Halaman hapus data siswa menampilkan keterangan berupa teks konfirmasi dan dua *button* fungsi. *Button* “Ya” digunakan untuk



Gambar 4. 158 Rancangan Antarmuka Halaman Hapus Data Siswa

27. Halaman Jadwal Pelajaran

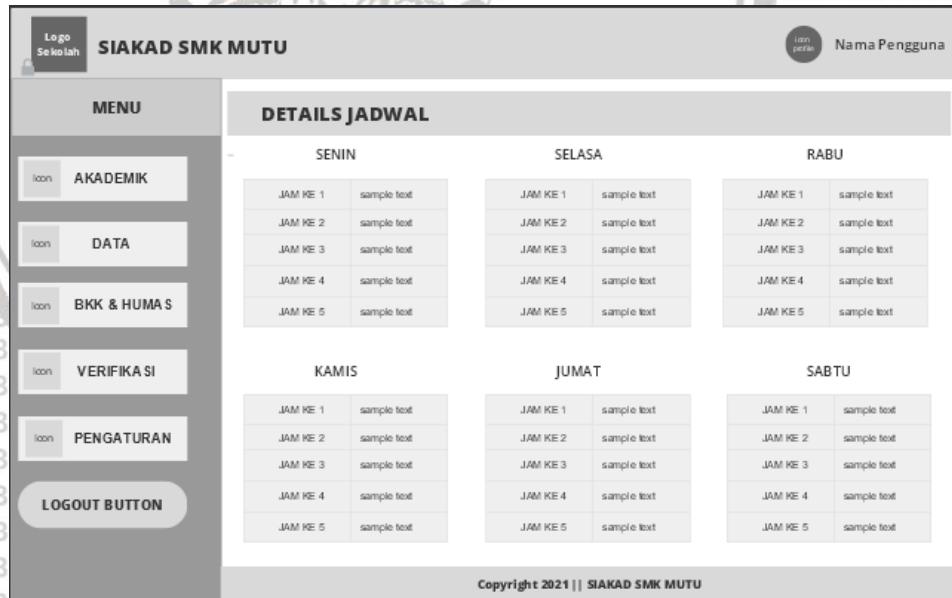
Halaman data jadwal pelajaran menampilkan daftar jadwal pelajaran yang ada saat ini dalam bentuk tabel. Komponen-komponen yang ada pada bagian *content* di halaman jadwal pelajaran meliputi tabel dengan tiga kolom, yaitu kolom nomor, kolom kelas, dan kolom aksi. Selain itu juga terdapat *button* untuk menambah data jadwal pelajaran, *button* untuk melihat details informasi jadwal pelajaran dan *button* untuk menghapus data jadwal pelajaran. Rancangan antarmuka halaman siswa dapat dilihat pada Gambar 4. 159.



Gambar 4. 159 Rancangan Antarmuka Halaman Daftar Jadwal Pelajaran

28. Halaman Detail Jadwal Pelajaran

Halaman detail jadwal pelajaran menampilkan informasi-informasi jadwal pelajaran dari hari senin sampai dengan sabtu dalam bentuk tabel. Rancangan antarmuka dari halaman detail jadwal pelajaran dapat dilihat pada Gambar 4. 160.



Gambar 4. 160 Rancangan Antarmuka Halaman Detail Jadwal Pelajaran

29. Halaman Tambah Jadwal Pelajaran

Halaman tambah jadwal pelajaran menampilkan *field* yang digunakan untuk memasukkan data mengenai jadwal pelajaran perhari, *field* yang ada terdiri dari *dropdown* yang daftar nilainya menyesuaikan

Gambar 4. 161 Rancangan Antarmuka Halaman Tambah Jadwal Pelajaran

30. Halaman Ubah Jadwal Pelajaran

Halaman ubah data jadwal pelajaran menampilkan *field* yang sama dengan yang ada di halaman tambah data jadwal pelajaran, yang berbeda adalah pada halaman ubah data jadwal pelajaran *field* yang ada sudah terisi nilai dengan data saat ini. *Button* perbarui digunakan untuk menjalankan fungsi *update* data jadwal pelajaran ke *database*. rancangan antarmuka halaman *update* data jadwal pelajaran dapat dilihat pada Gambar 4. 162.

Logo Sekolah Nama Pengguna

SIAKAD SMK MUTU

MENU

- AKADEMIK
- DATA
- BKK & HUMAS
- VERIFIKA SI
- PENGATURAN
- LOGOUT BUTTON

UBAH JADWAL PELAJARAN

KELAS : XI-TEI-B HARI : Senin

Mata Pelajaran Jam Ke 1 : Fisika Listrik Mata Pelajaran Jam Ke 3 : Kewirausahaan

Mata Pelajaran Jam Ke 2 : Bahasa Inggris Mata Pelajaran Jam Ke 4 : Matematika

Mata Pelajaran Jam Ke 5 : Sejarah Indonesia

Perbarui

Copyright 2021 || SIAKAD SMK MUTU

Gambar 4. 162 Rancangan Antarmuka Halaman Ubah Jadwal Pelajaran

31. Halaman Hapus Jadwal Pelajaran

Halaman hapus jadwal pelajaran menampilkan keterangan berupa teks konfirmasi dan dua button fungsi. Button “Ya” digunakan untuk menjalankan fungsi *delete* data jadwal pelajaran di *database*. Button “Tidak” digunakan untuk membatalkan fungsi hapus dan mengembalikan aktor pada halaman awal. Rancangan antarmuka halaman hapus data jadwal pelajaran dapat dilihat pada Gambar 4. 163.

Logo Sekolah Nama Pengguna

SIAKAD SMK MUTU

MENU

- AKADEMIK
- DATA
- BKK & HUMAS
- VERIFIKA SI
- PENGATURAN
- LOGOUT BUTTON

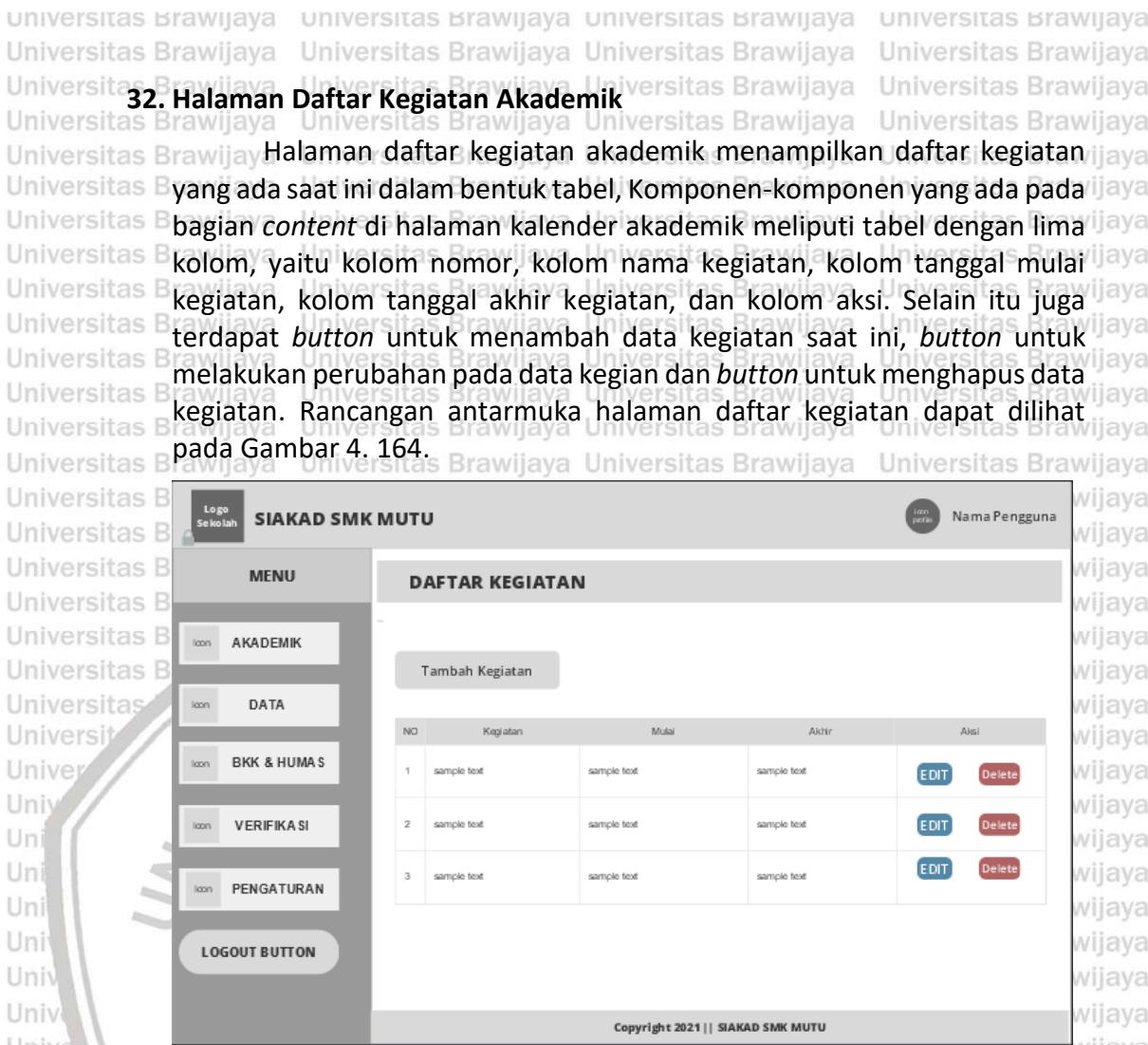
HAPUS DATA JADWAL

Yakin Ingin Menghapus informasi Jadwal ini dari Database?

Ya Tidak

Copyright 2021 || SIAKAD SMK MUTU

Gambar 4. 163 Rancangan Antarmuka Halaman Hapus Jadwal Pelajaran



Gambar 4. 164 Rancangan Antarmuka Halaman Daftar Kegiatan Akademik

33. Halaman Tambah Kegiatan Akademik

Halaman tambah kegiatan akademik menampilkan *field* yang digunakan untuk memasukkan data-data kegiatan akademik yang meliputi tanggal mulai kegiatan, tanggal akhir kegiatan, dan nama kegiatan. Tombol simpan digunakan untuk menjalankan fungsi simpan data kegiatan akademik ke *database*. Rancangan antarmuka halaman tambah kegiatan akademik dapat dilihat pada Gambar 4. 165.

SIAKAD SMK MUTU

MENU

- AKADEMIK**
- DATA**
- BKK & HUMAS**
- VERIFIKA SI**
- PENGATURAN**
- LOGOUT BUTTON**

TAMBAH KEGIATAN

MULAI KEGIATAN
Masukkan Tanggal Mulai Kegiatan

AKHIR KEGIATAN
Masukkan Tanggal Akhir Kegiatan

Keterangan Kegiatan
Masukkan Keterangan Kegiatan

Simpan

Copyright 2021 || SIAKAD SMK MUTU

Gambar 4. 165 Rancangan Antarmuka Halaman Tambah Kegiatan Akademik

34. Halaman Ubah Kegiatan Akademik

Halaman ubah kegiatan akademik menampilkan *field* yang sama dengan yang ada di halaman tambah kegiatan akademik, yang berbeda adalah pada halaman ubah informasi kegiatan *field* yang ada sudah terisi nilai dengan data saat ini. *Button* perbarui digunakan untuk menjalankan fungsi *update* data kegiatan ke *database*. Rancangan antarmuka halaman *update* informasi kegiatan dapat dilihat pada Gambar 4. 166.

SIAKAD SMK MUTU

MENU

- AKADEMIK**
- DATA**
- BKK & HUMAS**
- VERIFIKA SI**
- PENGATURAN**
- LOGOUT BUTTON**

UBAH INFORMASI KEGIATAN

MULAI KEGIATAN
12 Februari 2021

AKHIR KEGIATAN
27 Maret 2021

Keterangan Kegiatan
Daftar Ulang Siswa

Perbarui

Copyright 2021 || SIAKAD SMK MUTU

Gambar 4. 166 Rancangan Antarmuka Halaman Ubah Data Kegiatan Akademik

35. Halaman Hapus Kegiatan Akademik

Halaman hapus data kegiatan akademik menampilkan keterangan berupa teks konfirmasi dan dua *button* fungsi. *Button* "Ya" digunakan untuk menjalankan fungsi *delete* data kegiatan di *database*. *Button* "Tidak" digunakan untuk membatalkan fungsi hapus dan mengembalikan aktor pada halaman awal. Rancangan antarmuka halaman hapus data kegiatan dapat dilihat pada Gambar 4. 167.



Gambar 4. 167 Rancangan Antarmuka Halaman Hapus Data Kegiatan Akademik

36. Halaman Validasi Pelaksanaan Prakerin

Halaman validasi pelaksanaan prakerin menampilkan daftar pengajuan prakerin oleh siswa dalam bentuk tabel. Komponen-komponen yang ada pada bagian *content* di halaman validasi pelaksanaan prakerin meliputi tabel dengan lima kolom, yaitu kolom nomor, kolom ketua kelompok, kolom tempat prakerin, kolom jenis industri, dan kolom aksi. Pada kolom aksi terdapat *button* ACC yang berfungsi untuk memverifikasi nilai pengajuan penuntutan prakerin dan tombol tolak untuk mengajukan peninjauan ulang data dikirimkan. Rancangan antarmuka halaman validasi pelaksanaan prakerin dapat dilihat pada Gambar 4. 168.



Gambar 4. 168 Rancangan Antarmuka Halaman Validasi Pelaksanaan Prakerin

37. Halaman Mitra Industri

Halaman mitra industri menampilkan daftar mitra industri yang ada saat ini dalam bentuk tabel. Komponen-komponen yang ada pada bagian content di halaman mitra industri meliputi tabel dengan lima kolom, yaitu kolom nomor, kolom nama industri, kolom jenis industri, kolom alamat, dan kolom aksi. Selain itu juga terdapat button untuk menambah data mitra industri saat ini, button untuk melakukan perubahan pada data mitra industri dan button untuk menghapus data mitra industri. Rancangan antarmuka halaman bisa dilihat pada Gambar 4. 169.



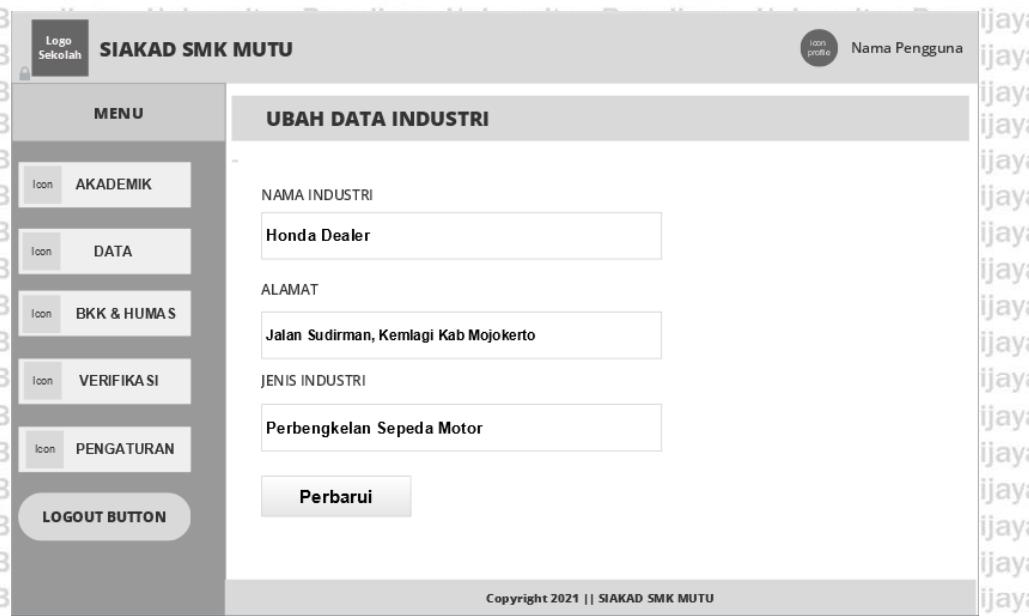
Gambar 4. 169 Rancangan Antarmuka Halaman Daftar Mitra Industri**38. Halaman Tambah Mitra Industri**

Halaman tambah mitra industri menampilkan *field* yang digunakan untuk memasukkan data-data mitra industri yang meliputi nama industri, alamat industri, dan jenis industri. Tombol simpan digunakan untuk menjalankan fungsi simpan data mitra industri ke *database*. Rancangan antarmuka halaman tambah mitra industri dapat dilihat pada Gambar 4. 170.

Gambar 4. 170 Rancangan Antarmuka Halaman Tambah Data Mitra Industri

39. Halaman Ubah Mitra Industri

Halaman ubah mitra industri menampilkan *field* yang sama dengan yang ada di halaman tambah mitra industri, yang berbeda adalah pada halaman ubah mitra industri *field* yang ada sudah terisi nilai dengan data saat ini. *Button* perbarui digunakan untuk menjalankan fungsi *update* data mitra industri ke *database*. Rancangan antarmuka halaman ubah data industri dapat dilihat pada Gambar 4. 171.



Gambar 4. 171 Rancangan Antarmuka Halaman Ubah Data Mitra Industri

40. Halaman Hapus Mitra Industri

Halaman hapus data mitra industri menampilkan keterangan berupa teks konfirmasi dan dua button fungsi. Button “Ya” digunakan untuk menjalankan fungsi *delete* data mitra industri di *database*. Button “Tidak” digunakan untuk membatalkan fungsi hapus dan mengembalikan aktor pada halaman awal. Rancangan antarmuka halaman hapus data mitra industri dapat dilihat pada Gambar 4. 172.



Gambar 4. 172 Rancangan Antarmuka Halaman Hapus Data Mitra Industri

41. Halaman Informasi Pelaksanaan Prakerin

Halaman daftar pelaksanaan prakerin menampilkan daftar pengajuan yang telah diajukan seluruh siswa. Komponen-komponen yang

ada pada bagian *content* di halaman nilai pelaksanaan prakerin siswa meliputi tabel dengan lima kolom, yaitu kolom nomor, kolom nama ketua kelompok, kolom tempat prakerin, kolom jenis industri dan kolom status. Pada kolom status terdapat *box* status pengajuan. Rancangan antarmuka halaman daftar pelaksanaan prakerin dapat dilihat pada Gambar 4. 173.

| NO | Ketua Kelompok | Tempat Prakerin | Jenis Industri | Status |
|----|----------------|-----------------|----------------|---------|
| 1 | sample text | sample text | sample text | ACC |
| 2 | sample text | sample text | sample text | Ditolak |
| 3 | sample text | sample text | sample text | ACC |

Gambar 4. 173 Rancangan Antarmuka Halaman Pelaksanaan Prakerin

42. Halaman Validasi Kalender Akademik

Halaman validasi kalender akademik berisi komponen tabel daftar kegiatan yang telah dimasukkan oleh admin sistem dan dua *button* fungsi yang terdiri dari *button* ACC yang digunakan untuk melakukan validasi kalender akademik, dan tombol tolak untuk mengirimkan permintaan peninjauan ulang terhadap daya yang dimasukkan. Rancangan antarmuka halaman validasi kegiatan akademik dapat dilihat pada Gambar 4. 174.

The screenshot shows a user interface for 'SIAKAD SMK MUTU'. At the top right is a profile icon and the text 'Nama Pengguna'. On the left, there's a sidebar titled 'MENU' with three items: 'PRAKERIN', 'VALIDASI', and 'KALENDER AKADEMIK'. Below the sidebar is a 'LOGOUT BUTTON'. The main content area is titled 'VERIFIKASI KALENDER AKADEMIK'. It contains a table with four rows and four columns. The columns are labeled 'NO', 'Kegiatan', 'Tanggal Mulai', and 'Tanggal Selesai'. The rows contain sample text. At the bottom of the table are two buttons: a green 'ACC' button and a red 'Tolak' button. At the very bottom of the page is the text 'Copyright 2021 || SIAKAD SMK MUTU'.

| NO | Kegiatan | Tanggal Mulai | Tanggal Selesai |
|----|-------------|---------------|-----------------|
| 1 | sample text | sample text | sample text |
| 2 | sample text | sample text | sample text |
| 3 | sample text | sample text | sample text |
| 4 | sample text | sample text | sample text |

Gambar 4. 174 Rancangan Antarmuka Halaman Validasi Kalender Akademik

43. Halaman Validasi Penetapan Wali Kelas

Halaman validasi penetapan wali kelas menampilkan daftar pembagian wali kelas dalam bentuk tabel. Komponen-komponen yang ada pada bagian *content* di halaman validasi penetapan wali kelas meliputi tabel dengan lima kolom, yaitu kolom nomor, kolom kelas, kolom jurusan, kolom wali kelas, dan kolom aksi. Pada kolom aksi terdapat button ACC yang berfungsi untuk memverifikasi nilai penetapan wali kelas dan tombol tolak untuk mengajukan peninjauan ulang terhadap penetapan wali kelas. Rancangan antarmuka halaman validasi penetapan wali kelas dapat dilihat pada Gambar 4. 175.

| NO | KELAS | JURUSAN | WALI KELAS | Aksi |
|----|-------------|-------------|-------------|---|
| 1 | sample text | sample text | sample text | <button>ACC</button> <button>Tolak</button> |
| 2 | sample text | sample text | sample text | <button>ACC</button> <button>Tolak</button> |
| 3 | sample text | sample text | sample text | <button>ACC</button> <button>Tolak</button> |

Copyright 2021 || SIAKAD SMK MUTU

Gambar 4. 175 Rancangan Antarmuka Halaman Validasi Wali Kelas

44. Halaman Perbarui Data Diri

Halaman perbarui data dibagi menjadi dua, yaitu halaman perbarui data diri untuk guru dan halaman perbarui data diri untuk siswa. Halaman perbarui data diri siswa, dibatasi hanya pada beberapa data yaitu data nama, data jenis kelamin, data agama, data tempat tanggal lahir, data nomor telepon, data nomor kelahiran di keluarga, dan data informasi orang tua. Rancangan halaman perbarui data diri siswa dapat dilihat pada Gambar 4. 176.

Copyright 2021 || SIAKAD SMK MUTU

Gambar 4. 176 Rancangan Antarmuka Halaman Perbarui Data Diri Siswa

Halaman perbarui data diri guru juga dibatasi hanya pada beberapa data yaitu data nama, data jenis kelamin, data agama, data tempat tanggal

The screenshot shows a user interface for updating personal data. On the left, there's a sidebar with a logo and several menu options: DATA DIRI, KELAS YANG DIAJAR, EVALUASI SISWA, VALIDASI NILAI, and KALENDER AKADEMIK. The main area is titled 'PERBARUI DATA DIRI'. It contains the following fields:

- NAMA LENGKAP:** Oemar Bakrie S.Pd
- TANGGAL LAHIR:** 10/05/1978
- JENIS KELAMIN:** Laki - Laki (radio button selected)
- PENDIDIKAN TERAKHIR:** SI
- AGAMA:** Islam
- MASA MENGABDI:** 2011
- TEMPAT LAHIR:** Mojokerto
- STATUS:** Pegawai Penuh

At the bottom right, it says 'Copyright 2021 || SIAKAD SMK MUTU'.

Gambar 4. 177 Rancangan Antarmuka Halaman Perbarui Data Diri Guru

45. Halaman Kelas Yang Diwakili

Halaman kelas yang diwakili terdiri dari beberapa komponen yang meliputi tabel kelas yang diajar dan tabel kelas yang diwakili. Tabel kelas yang diajar terdiri atas lima kolom yaitu kolom nomor, kolom nama kelas, kolom jurusan kelas, kolom wali kelas dan kolom aksi. Pada kolom aksi tabel kelas yang diajar terdapat dua *button*, yaitu *button* detail untuk melihat informasi kelas secara detail dan *button* masukkan nilai untuk memasukkan nilai hasil evaluasi siswa. Tabel kelas yang diwakili memiliki kolom yang sama dengan tabel kelas yang diwakili, perbedaan terdapat pada *button* lihat nilai yang berfungsi untuk melihat dan melakukan validasi nilai. Rancangan antarmuka halaman kelas yang diwakili dapat dilihat pada Gambar 4. 178.

Gambar 4. 178 Rancangan Antarmuka Halaman Kelas Yang Diwakili**46. Halaman Masukkan Nilai Hasil Evaluasi**

Halaman tambah kegiatan akademik menampilkan *field* yang digunakan untuk memasukkan data-data untuk nilai hasil evaluasi yang meliputi nama siswa, nilai pengetahuan, deskripsi nilai pengetahuan, nilai ketrampilan dan deskripsi nilai ketrampilan. Tombol simpan digunakan untuk menjalankan fungsi simpan data nilai ke *database*. Rancangan antarmuka halaman masukkan nilai hasil evaluasi dapat dilihat pada Gambar 4. 179.

Gambar 4. 179 Rancangan Antarmuka Halaman Nilai Hasil Evaluasi

47. Halaman Daftar Nilai Hasil Evaluasi

Halaman data nilai hasil evaluasi menampilkan daftar hasil evaluasi yang telah ditempuh siswa dalam bentuk tabel. Komponen-komponen yang ada pada bagian *content* di halaman nilai hasil evaluasi meliputi tabel dengan lima kolom, yaitu kolom nomor, kolom mata pelajaran, kolom guru pengampu, kolom tipe evaluasi dan kolom aksi. Pada kolom aksi terdapat *button details* yang berfungsi untuk melihat hasil evaluasi secara detail. Rancangan antarmuka halaman nilai hasil evaluasi dapat dilihat pada Gambar 4. 180.

| NO | MATA PELAJARAN | GURU PENGAMPU | TIPE EVALUASI | AKSI |
|----|----------------|---------------|---------------|-------------------------|
| 1 | sample text | sample text | sample text | Details |
| 2 | sample text | sample text | sample text | Details |
| 3 | sample text | sample text | sample text | Details |

Copyright 2021 || SIAKAD SMK MUTU

Gambar 4. 180 Rancangan Antarmuka Halaman Daftar Nilai Hasil Evaluasi

48. Halaman Ubah Nilai Hasil Evaluasi

Halaman ubah nilai hasil evaluasi menampilkan *field* yang sama dengan yang ada di halaman masukkan nilai hasil evaluasi, yang berbeda adalah pada halaman ubah nilai hasil evaluasi *field* yang ada sudah terisi nilai dengan data saat ini. *Button* perbarui digunakan untuk menjalankan fungsi *update* data nilai hasil evaluasi ke *database*. Rancangan antarmuka halaman ubah data nilai hasil evaluasi dapat dilihat pada Gambar 4. 181.

SIAKAD SMK MUTU

MENU

- Icon DATA DIRI
- Icon KELAS YANG DIAJAR
- Icon EVALUASI SISWA
- Icon VALIDASI NILAI
- Icon KALENDER AKADEMIK

LOGOUT BUTTON

UBAH NILAI HASIL EVALUASI

PILIH SISWA

Taftiyah Ilham

NILAI PENGETAHUAN

85

DESKRIPSI

Siswa memiliki pemahaman cukup baik mengenai materi pelajaran

NILAI KETRAMPILAN

90

DESKRIPSI

Siswa dapat mempraktikkan ilmu yang diperoleh dalam kegiatan prakt

Perbarui

Copyright 2021 || SIAKAD SMK MUTU

Gambar 4. 181 Rancangan Antarmuka Halaman Ubah Nilai Hasil Evaluasi

49. Halaman Daftar Rapor

Halaman daftar rapor menampilkan daftar hasil rapor siswa dalam bentuk tabel. Komponen-komponen yang ada pada bagian *content* halaman nilai daftar rapor siswa meliputi tabel dengan lima kolom, yaitu kolom nomor, kolom tahun ajaran, kolom semester, kolom tipe evaluasi dan kolom aksi. Pada kolom aksi terdapat *button* details yang berfungsi untuk melihat rapor secara lebih detail. Rancangan antarmuka halaman nilai daftar rapor dapat dilihat pada Gambar 4. 182.

SIAKAD SMK MUTU

MENU

- Icon DATA DIRI
- Icon KELAS SAAT INI
- Icon RAPOR
- Icon PRAKERIN
- Icon KALENDER AKADEMIK

LOGOUT BUTTON

DAFTAR RAPOR SISWA

| NO | Tahun Ajaran | Semester | Tipe Evaluasi | AKSI |
|----|--------------|-------------|---------------|----------------|
| 1 | sample text | sample text | sample text | DETAILS |
| 2 | sample text | sample text | sample text | DETAILS |
| 3 | sample text | sample text | sample text | DETAILS |

Copyright 2021 || SIAKAD SMK MUTU

Gambar 4. 182 Rancangan Antarmuka Halaman Daftar Rapor

50. Halaman Nilai Detail Rapor

Halaman nilai hasil evaluasi terdiri dari komponen teks untuk informasi evaluasi dan komponen tabel yang menampilkan nilai hasil evaluasi siswa. Rancangan antarmuka halaman nilai hasil evaluasi dapat dilihat pada Gambar 4. 183.

| NO | Mata Pelajaran | Nilai Pengetahuan | Deskripsi | Nilai Keterampilan | Deskripsi | Nilai Akhir |
|----|----------------|-------------------|-------------|--------------------|-------------|--------------|
| 1 | sample text | Sample Value | sample text | Sample Value | sample text | Sample Value |
| 2 | sample text | Sample Value | sample text | Sample Value | sample text | Sample Value |
| 3 | sample text | Sample Value | sample text | Sample Value | sample text | Sample Value |
| 4 | sample text | Sample Value | sample text | Sample Value | sample text | Sample Value |
| 5 | sample text | Sample Value | sample text | Sample Value | sample text | Sample Value |

Copyright 2021 || SIAKAD SMK MUTU

Gambar 4. 183 Rancangan Antarmuka Halaman Detail Rapor

51. Halaman Pengajuan Penuntasan Pelaksanaan Prakerin

Halaman pengajuan pelaksanaan prakerin menampilkan *field* yang digunakan untuk memasukkan data-data pelaksanaan prakerin yang meliputi tanggal ketua kelompok, NISN siswa, tempat prakerin dan jenis industri tempat prakerin. Tombol simpan digunakan untuk menjalankan fungsi simpan data pengajuan prakerin ke *database*. Rancangan antarmuka halaman pengajuan prakerin dapat dilihat pada Gambar 4. 184.

PENGAJUAN PENUNTASAN PRAKERIN

KETUA KELOMPOK
Masukkan nama ketua kelompok anda

NISN KETUA KELOMPOK
Masukkan NISN ketua kelompok anda

TEMPAT PRAKERIN
Masukkan nama tempat anda prakerin

JENIS INDUSTRI TEMPAT PPRAKERIN
Masukkan Jenis Industri Tempat Anda Prakerin

AJUKAN

Copyright 2021 || SIAKAD SMK MUTU

Gambar 4. 184 Rancangan Antarmuka Halaman Pengajuan Pelaksanaan Prakerin

52. Halaman Daftar Pengajuan Prakerin

Halaman daftar pengajuan prakerin menampilkan daftar pengajuan yang telah diajukan siswa. Komponen-komponen yang ada pada bagian content di halaman nilai hasil evaluasi meliputi tabel dengan lima kolom, yaitu kolom nomor, kolom nama kelompok, kolom tempat prakerin, kolom jenis industri dan kolom status. Pada kolom status terdapat box status pengajuan. Rancangan antarmuka halaman daftar pengajuan penuntasan prakerin dapat dilihat pada Gambar 4. 185.

| NO | Ketua Kelompok | Tempat Prakerin | Jenis Industri | Status |
|----|----------------|-----------------|----------------|--------------|
| 1 | sample text | sample text | sample text | Menunggu ACC |
| 2 | sample text | sample text | sample text | Ditolak |

Gambar 4. 185 Rancangan Antarmuka Halaman Daftar Pengajuan Prakerin

The screenshot shows a user interface for managing practical training applications. The main title is 'SIAKAD SMK MUTU'. On the left, there's a sidebar with a logo and a 'LOGOUT BUTTON'. The main content area is titled 'UBAH PENGAJUAN PENUNTASAN PRAKERIN'. It contains several input fields: 'KETUA KELOMPOK' (filled with 'Taftiyah Ilham'), 'NISN KETUA KELOMPOK' (filled with '7812'), 'TEMPAT PRAKERIN' (filled with 'PT. Mertex Indonesia'), and 'JENIS INDUSTRI TEMPAT PPRAKERIN' (filled with 'Industri Garmen'). Below these fields is a 'Perbarui' button.

Gambar 4. 186 Rancangan Antarmuka Halaman Ubah Pengajuan Pelaksanaan Prakerin

4.3 Implementasi

Tahap implementasi merupakan penerapan hasil pemodelan yang ada pada tahap perancangan kedalam kode program. Pada tahap implementasi sistem digunakan beberapa perangkat lunak dan *framework* untuk memudahkan proses pengembangan sistem informasi, sehingga dapat memenuhi fungsionalitas yang telah dispesifikasi.

4.3.1 Batasan Implementasi

Implementasi sistem informasi akademik SMK Muhammadiyah 1 Kemlagi dibatasi pada beberapa hal sebagai berikut:

1. Sistem dirancang untuk bekerja secara optimal pada perangkat *desktop* seperti laptop atau komputer pribadi.
2. Pada penelitian ini sistem diproyeksikan dijalankan selama kurun waktu satu semester (6 bulan).

3. Sistem kompatibel dengan peramban web versi terbaru seperti Google Chrome dan Mozilla Firefox versi baru, untuk peramban web lain tidak dilakukan pengujian.

4.3.2 Spesifikasi Perangkat Lunak

Perangkat lunak berikut adalah perangkat lunak yang digunakan untuk membantu pengembangan sistem informasi akademik SMK Muhammadiyah 1 Kemlagi. Daftar perangkat lunak dan spesifikasi dapat dilihat pada Tabel 4.73.

Tabel 4.73 Daftar Spesifikasi Perangkat Lunak

| No | Perangkat Lunak | Kategori | Keterangan |
|----|-----------------|----------------|--|
| 1 | Windows 10 | Sistem Operasi | Versi 64-bit berfungsi untuk penghubung dengan perangkat keras, sehingga perangkat lunak lain bisa dijalankan. |
| 2 | Apache | Web Server | Berfungsi untuk memproses berkas PHP dan menjalankan sistem informasi. |
| 3 | MySQL | DBMS | Berfungsi sebagai tempat penyimpanan data akademik, yang dihubungkan secara relasional. |
| 4 | VS Code | Code Editor | Berfungsi untuk melakukan pembuatan dan penyuntingan berkas-berkas kode program. |
| 5 | Codeigniter | Framework | Berfungsi untuk menyediakan fungsi-fungsi yang dibutuhkan dalam pengembangan sistem. |
| 6 | Bizagi Modeler | BPMN Modeler | Berfungsi untuk membantu pemodelan proses bisnis yang ada saat ini. |

Tabel 4. 73 Daftar Spesifikasi Perangkat Lunak (Lanjutan)

| No | Perangkat Lunak | Kategori | Keterangan |
|----|-----------------|-------------------|--|
| 7 | Justinmind | Prototype Builder | Berfungsi untuk membuat rancangan antarmuka sistem informasi. |
| 8 | Visual Paradigm | UML Designer | Berfungsi untuk membuat diagram-diagram sistem, yang meliputi <i>sequence diagram</i> , <i>class diagram</i> , <i>use case diagram</i> , <i>activity diagram</i> . |
| 9 | SB Admin 2 | Admin Template | Berfungsi untuk menyediakan komponen-komponen antarmuka yang siap pakai. |
| 10 | Google Chrome | Web Browser | Berfungsi untuk mengakses data-data dari <i>web server</i> . |

4.3.3 Implementasi Basis Data

Bagian ini membahas mengenai implementasi terhadap rancangan basis data yang telah didefinisikan sebelumnya, melalui *Data Definition Language (DDL)* yang merupakan perintah-perintah SQL untuk mengimplementasikan rancangan menjadi objek basis data. Implementasi melalui DDL pada masing-masing tabel basis data sistem informasi akademik SMK Muhammadiyah 1 Kemlagi adalah sebagai berikut:

Tabel 4. 74 DDL Tabel Jurusan

| Tabel : jurusan |
|---|
| <pre>CREATE TABLE `jurusan` (`id` int(11) NOT NULL AUTO_INCREMENT, `nama` varchar(255) NOT NULL, `kode_jurusan` varchar(10) NOT NULL) ALTER TABLE `jurusan` ADD PRIMARY KEY (`id`);</pre> |

Tabel 4. 75 DDL Tabel Kelas**Tabel : kelas**

```

CREATE TABLE `kelas` (
  `id` int(11) NOT NULL,
  `jurusan` int(11) DEFAULT NULL,
  `kode_kelas` varchar(10) NOT NULL,
  `nama_kelas` varchar(255) NOT NULL,
  `tingkat` varchar(25) NOT NULL,
  `wali_kelas` varchar(50) DEFAULT NULL,
  `id_tahun_ajaran` int(11) DEFAULT NULL,
  `status` int(11) DEFAULT NULL
)
ALTER TABLE `kelas`
  ADD PRIMARY KEY (`id`),
  ADD KEY `Kelas_fk0` (`wali_kelas`),
  ADD KEY `kelas_ibfk_1` (`jurusan`),
  ADD KEY `kelas_ibfk_2` (`id_tahun_ajaran`),
  ADD KEY `status_fk` (`status`);

ALTER TABLE `kelas`
  ADD CONSTRAINT `kelas_ibfk_1` FOREIGN KEY (`jurusan`)
  REFERENCES `jurusan` (`id`) ON DELETE SET NULL ON UPDATE CASCADE;
ADD CONSTRAINT `Kelas_fk0` FOREIGN KEY (`wali_kelas`) REFERENCES
`guru` (`NBM`) ON DELETE SET NULL ON UPDATE CASCADE;
ADD CONSTRAINT `Kelas_ibfk2` FOREIGN KEY (`id_tahun_ajaran`)
REFERENCES `tahun_ajaran` (`id`) ON DELETE SET NULL ON UPDATE
CASCADE;
ADD CONSTRAINT `status_fk` FOREIGN KEY (`status`) REFERENCES
`status` (`id`) ON DELETE SET NULL ON UPDATE CASCADE;

```

Tabel 4. 76 DDL Tabel Mata Pelajaran**Tabel : pelajaran**

```

CREATE TABLE `pelajaran` (
  `id` int(11) NOT NULL,
  `kode` varchar(25) NOT NULL,
  `nama` varchar(255) NOT NULL,
  `id_guru` varchar(25) DEFAULT '-',
  `jurusan` int(11) DEFAULT 0
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

ALTER TABLE `pelajaran`

```

```
    ADD CONSTRAINT `Pelajaran_fk0` FOREIGN KEY (`id_guru`) REFERENCES `guru`(`NBM`) ON DELETE SET NULL ON UPDATE CASCADE,  
    ADD CONSTRAINT `Pelajaran_fk1` FOREIGN KEY (`jurusan`) REFERENCES `jurusan`(`id`) ON DELETE SET NULL ON UPDATE CASCADE;
```

Tabel 4. 77 DDL Tabel Jadwal Pelajaran

Tabel : jadwal_pelajaran

```
CREATE TABLE `jadwal_pelajaran` (
  `id` int(11) NOT NULL,
  `hari` varchar(10) NOT NULL,
  `waktu` varchar(50) NOT NULL,
  `pelajaran` int(11) DEFAULT NULL,
  `kelas` int(11) DEFAULT NULL
)
ALTER TABLE `jadwal_pelajaran`
  ADD PRIMARY KEY (`id`),
  ADD KEY `fk_pelajaran` (`pelajaran`),
  ADD KEY `fk_kelas` (`kelas`);
ALTER TABLE `jadwal_pelajaran`
  ADD CONSTRAINT `fk_kelas` FOREIGN KEY (`kelas`)
    REFERENCES `kelas`(`id`) ON DELETE SET NULL ON UPDATE CASCADE;
```

Tabel 4. 78 DDL Tabel Guru

Tabel : guru

```
CREATE TABLE `guru` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `NBM` varchar(25) NOT NULL,
  `nama` varchar(255) NOT NULL,
  `jenis_kelamin` varchar(25) NOT NULL,
  `agama` varchar(50) NOT NULL,
  `alamat` varchar(255) NOT NULL,
  `tempat_lahir` varchar(100) NOT NULL,
  `tanggal_lahir` date NOT NULL,
  `pend_terakhir` varchar(50) NOT NULL,
  `jabatan` varchar(50) NOT NULL,
  `mulai_mengabdi` date NOT NULL,
  `status` varchar(50) NOT NULL,
  `nomor_telepon` varchar(20) NOT NULL,
```

```
foto` varchar(255) NOT NULL  
)  
ALTER TABLE `guru`  
ADD PRIMARY KEY (`NBM`),  
ADD UNIQUE KEY `id_guru`(`id`);
```

Tabel 4. 79 DDL Tabel Siswa

Tabel : siswa

```
CREATE TABLE `siswa` (
  `NISN` varchar(50) NOT NULL,
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nama` varchar(255) NOT NULL,
  `jenis_kelamin` varchar(25) NOT NULL,
  `tempat_lahir` varchar(255) NOT NULL,
  `tanggal_lahir` date NOT NULL,
  `agama` varchar(25) NOT NULL,
  `alamat` varchar(255) NOT NULL,
  `anak_ke` smallint(6) NOT NULL,
  `jumlah_saudara` smallint(6) NOT NULL,
  `id_status` smallint(6) DEFAULT NULL,
  `nomor_telepon` varchar(15) NOT NULL,
  `foto` varchar(255) NOT NULL
)
ALTER TABLE `siswa`
ADD PRIMARY KEY (`NISN`),
ADD UNIQUE KEY `id` (`id`),
ADD KEY `Siswa_fk1` (`id_status`);
ALTER TABLE `siswa`
ADD CONSTRAINT `Siswa_fk1` FOREIGN KEY (`id_status`) REFERENCES
`status` (`id`) ON DELETE SET NULL ON UPDATE CASCADE;
```

Tabel 4. 80 DDL Tabel Orang Tua

Tabel : orang_tua

```
CREATE TABLE `orang_tua` (
  `id` int(11) NOT NULL,
  `NISN` varchar(50) DEFAULT NULL,
  `nama_ayah` varchar(255) NOT NULL,
```

```

`pekerjaan_ayah` varchar(15) NOT NULL,
`telepon_ayah` varchar(15) NOT NULL,
`nama_ibu` varchar(255) NOT NULL,
`pekerjaan_ibu` varchar(255) NOT NULL,
`telepon_ibu` varchar(15) NOT NULL
);
ALTER TABLE `orang_tua`
ADD PRIMARY KEY (`id`),
ADD KEY `Orang_tua_fk0` (`NISN`);
ALTER TABLE `orang_tua`
ADD CONSTRAINT `Orang_tua_fk0` FOREIGN KEY (`NISN`)
REFERENCES `siswa`(`NISN`) ON DELETE CASCADE ON UPDATE CASCADE;

```

Tabel 4. 81 DDL Tabel Sekolah Sebelumnya**Tabel : sekolah_sebelumnya**

```

CREATE TABLE `sekolah_sebelumnya` (
`id` int(11) NOT NULL,
`NISN` varchar(50) DEFAULT NULL,
`nama_sekolah` varchar(255) NOT NULL,
`diterima_dikelas` varchar(50) NOT NULL,
`tanggal` date NOT NULL
);
ALTER TABLE `sekolah_sebelumnya`
ADD PRIMARY KEY (`id`),
ADD KEY `sekolah_sebelumnya_fk0` (`NISN`); ALTER TABLE
`orang_tua`;
ALTER TABLE `sekolah_sebelumnya`
ADD CONSTRAINT `sekolah_sebelumnya_fk0` FOREIGN KEY (`NISN`)
REFERENCES `siswa`(`NISN`) ON DELETE CASCADE ON UPDATE CASCADE;

```

Tabel 4. 82 DDL Tabel Riwayat Kelas**Tabel : riwayat_kelas**

```

CREATE TABLE `riwayat_kelas` (
`id` int(11) NOT NULL,
`NISN` varchar(50) DEFAULT NULL,
`kelas_10` int(11) DEFAULT NULL,
`kelas_11` int(11) DEFAULT NULL,
`kelas_12` int(11) DEFAULT NULL

```

```

) )
ALTER TABLE `riwayat_kelas`
ADD PRIMARY KEY (`id`),
ADD KEY `Riwayat_Kelas_fk0`(`NISN`),
ADD KEY `Riwayat_Kelas_fk1`(`kelas_10`),
ADD KEY `Riwayat_Kelas_fk2`(`kelas_11`),
ADD KEY `Riwayat_Kelas_fk3`(`kelas_12`);

ALTER TABLE `riwayat_kelas`
ADD CONSTRAINT `Riwayat_Kelas_fk0` FOREIGN KEY (`NISN`)
REFERENCES `siswa`(`NISN`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `Riwayat_Kelas_fk1` FOREIGN KEY (`kelas_10`)
REFERENCES `kelas`(`id`) ON DELETE SET NULL ON UPDATE CASCADE,
ADD CONSTRAINT `Riwayat_Kelas_fk2` FOREIGN KEY (`kelas_11`)
REFERENCES `kelas`(`id`) ON DELETE SET NULL ON UPDATE CASCADE,
ADD CONSTRAINT `Riwayat_Kelas_fk3` FOREIGN KEY (`kelas_12`)
REFERENCES `kelas`(`id`) ON DELETE SET NULL ON UPDATE CASCADE;

ADD CONSTRAINT `sekolah_sebelumnya_fk0` FOREIGN KEY(`NISN`)
REFERENCES `siswa`(`NISN`) ON DELETE CASCADE ON UPDATE CASCADE;

```

Tabel 4. 83 DDL Tabel Kegiatan Akademik

| Tabel : kegiatan_akademik |
|--|
| <pre> CREATE TABLE `kegiatan_akademik` (`id` int(11) NOT NULL, `nama_kegiatan` varchar(255) NOT NULL, `tahun_ajaran` int(11) DEFAULT NULL, `tanggal_mulai` date NOT NULL, `tanggal_akhir` date NOT NULL, `status` int(11) DEFAULT NULL) ALTER TABLE `kegiatan_akademik` ADD PRIMARY KEY(`id`), ADD KEY `status_fk_ka`(`status`), ADD KEY `tahun_ajaran_FK`(`tahun_ajaran`); ALTER TABLE `kegiatan_akademik` ADD CONSTRAINT `status_fk_ka` FOREIGN KEY(`status`) REFERENCES `verifikasi`(`id`) ON DELETE SET NULL ON UPDATE CASCADE; ADD CONSTRAINT `tahun_ajaran_fk` FOREIGN KEY(`tahun_ajaran`) REFERENCES `tahun_ajaran`(`id`) ON DELETE SET NULL ON UPDATE CASCADE; </pre> |

Tabel 4. 84 DDL Tabel Mitra Industri

```
Tabel : mitra_industri  
CREATE TABLE `mitra_industri` (  
    `id` int(11) NOT NULL,  
    `nama` varchar(100) NOT NULL,  
    `jenis_industri` varchar(100) NOT NULL,  
    `alamat` varchar(100) NOT NULL)  
ALTER TABLE `mitra_industri`  
ADD PRIMARY KEY (`id`);
```

Tabel 4. 85 DDL Tabel Nilai

Tabel : nilai

```
CREATE TABLE `nilai` (
  `id` int(11) NOT NULL,
  `NISN` varchar(50) DEFAULT NULL,
  `kelas` int(11) DEFAULT NULL,
  `pelajaran` int(11) DEFAULT NULL,
  `nilai_pengetahuan` float NOT NULL,
  `deskripsi_np` varchar(2200) NOT NULL,
  `nilai_ketrampilan` float NOT NULL,
  `deskripsi_nk` varchar(2200) NOT NULL,
  `jenis_evaluasi` varchar(50) NOT NULL
)
ALTER TABLE `nilai`
  ADD PRIMARY KEY (`id`),
  ADD KEY `Nilai_fk0` (`NISN`),
  ADD KEY `Nilai_fk1` (`kelas`),
  ADD KEY `Nilai_fk2` (`pelajaran`);

ALTER TABLE `nilai`
  ADD CONSTRAINT `Nilai_fk0` FOREIGN KEY (`NISN`) REFERENCES `siswa`(`NISN`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `Nilai_fk1` FOREIGN KEY (`kelas`) REFERENCES `kelas`(`id`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `Nilai_fk2` FOREIGN KEY (`pelajaran`) REFERENCES `pelajaran`(`id`) ON DELETE CASCADE ON UPDATE CASCADE;
```

Tabel 4. 86 DDL Tabel Prakerin

Tabel : mitra_industri

```
CREATE TABLE `prakerin` (
```

```

CREATE TABLE `prakerin` (
  `id` int(11) NOT NULL,
  `NISN` varchar(50) NOT NULL,
  `nama` varchar(100) NOT NULL,
  `tempat_prakerin` varchar(255) NOT NULL,
  `alamat` varchar(255) NOT NULL,
  `deskripsi` varchar(2200) NOT NULL,
  `status` int(11) NOT NULL
);
ALTER TABLE `prakerin`
  ADD PRIMARY KEY (`id`),
  ADD KEY `status_fk_prakerin`(`status`);

```

Tabel 4. 87 DDL Tabel Status**Tabel : status**

```

CREATE TABLE `status` (
  `id` smallint(6) NOT NULL,
  `status` tinytext NOT NULL
)
ALTER TABLE `status`
  ADD PRIMARY KEY (`id`),
  ADD UNIQUE KEY `id` (`id`),
  ADD UNIQUE KEY `status` (`status`) USING HASH;

```

Tabel 4. 88 DDL Tabel Tahun Ajaran**Tabel : tahun_ajaran**

```

CREATE TABLE `tahun_ajaran` (
  `id` int(11) NOT NULL,
  `tahun_ajaran` varchar(50) NOT NULL
)
ALTER TABLE `tahun_ajaran`
  ADD PRIMARY KEY (`id`),
  ADD UNIQUE KEY `tahun_ajaran` (`tahun_ajaran`);

```

Tabel 4. 89 DDL Tabel Users**Tabel : users**

```
CREATE TABLE `users` (
```

```
 `username` varchar(255) NOT NULL,  
 `password` varchar(255) NOT NULL,  
 `level` int(11) NOT NULL  
  
ALTER TABLE `users`  
ADD PRIMARY KEY ( `username` );
```

Tabel 4. 90 DDL Tabel Verifikasi

Tabel : verifikasi

```
CREATE TABLE `verifikasi` (
  `id` int(11) NOT NULL,
  `status` varchar(25) NOT NULL
);
ALTER TABLE `verifikasi`
  ADD PRIMARY KEY (`id`);
```

4.3.4 Implementasi Fungsi Utama

Pada bagian ini merupakan beberapa potongan kode program yang terdapat pada fungsi utama sistem, yang meliputi proses verifikasi *login* sistem, proses *logout* sistem, pembacaan data, penambahan data, pengubahan data, penghapusan data, pengubahan status data (verifikasi), dan pengolahan data berformat gambar.

1. Render View

Proses *render view* dilakukan di bagian *controller*, proses *render view* akan melakukan render terhadap antarmuka yang ada pada bagian *view*. *Render view* terdiri atas dua jenis, yaitu *render* dengan membawa data dari *database*, atau tanpa membawa data (hanya menampilkan *view*). Pada Tabel 4. 91 merupakan *render* tanpa membawa data dan merupakan *render* dengan membawa data.

Tabel 4. 91 Kode Fungsi Utama Render View

| No | Kode Program |
|----|---|
| 1 | public function nama_function(){ |
| 2 | \$this->load- |
| 3 | >view('nama_package/templates/header'); |
| 4 | \$this->load- |
| 5 | >view('nama_package/templates/sidebar'); |
| 6 | \$this->load->view('nama_package/nama_view'); |
| 7 | \$this->load- |
| | >view('nama_package/templates/footer');} |

Kode Program pada Tabel 4. 91 merupakan sebuah fungsi untuk merender sebuah view minimal merender tiga templates yaitu header, sidebar dan footer, ketiga *templates* tersebut diperlukan agar sistem informasi dapat ditampilkan dengan baik, karena 3 file view tersebut mengandung kode untuk memuat file CSS, dan Javascript, serta antarmuka umum dari sistem informasi seperti yang terdapat pada bagian perancangan antarmuka. Pada baris kode ke 6-7 merupakan kode untuk memuat *view* antarmuka secara spesifik, misalnya untuk memuat tampilan *daftar jurusan* maka bagian *nama_view* diubah menjadi *daftar_jurusan*.

Tabel 4. 92 Kode Fungsi Utama Render View Dengan Data

| No | Kode Program |
|----|--|
| 1 | public function <i>nama_function</i> () |
| 2 | { |
| 3 | \$data = \$this->user_model->getData(\$this->session->userdata['username']); |
| 4 | \$data = array(|
| 5 | 'username' => \$data->username, |
| 6 | 'level' => \$data->level |
| 7 |); |
| 8 | \$data["guru"] = \$this->model_guru->getAll(); |
| 9 | \$this->load->view('nama_package/templates/header'); |
| 10 | \$this->load->view('nama_package/templates/sidebar'); |
| 11 | \$this->load->view('nama_package/nama_view'); |
| 12 | \$this->load->view('nama_package/templates/footer'); |
| 13 | } |

Kode program pada Tabel 4. 92 merupakan contoh sebuah *render view* yang membawa data, data yang diperoleh itu dengan memanggil method pada bagian model, yang selanjutnya akan dikirim pada view oleh *controller*, pada sistem informasi akademik SMK Muhammadiyah 1 Kemlagi, semua pemuatan tampilan selain fungsi *login sistem*, menggunakan render view dengan data, karena data *login user* akan disimpan dan dipakai dalam identifikasi pengguna di sistem.

2. Login Sistem Informasi

Fungsi utama untuk melakukan *login* pada sistem informasi dapat dilihat pada Tabel 4. 93.

Tabel 4. 93 Kode Program Fungsi Utama Login Sistem Informasi

| No | Kode Program |
|----|---|
| 1 | public function proses_login() |
| 2 | { |
| 3 | \$this->form_validation->set_rules('username', 'username', 'required', ['required' => 'Username wajib diisi']); |
| 4 | \$this->form_validation->set_rules('password', 'password', 'required', ['required' => 'Password wajib diisi']); |
| 5 | if (\$this->form_validation->run() == FALSE) { |
| 6 | \$this->load->view('administrator/templates-administrator/header'); |
| 7 | \$this->load->view('administrator/login'); |
| 8 | \$this->load->view('administrator/templates-administrator/footer'); |
| 9 | } else { |
| 10 | \$username = \$post['username']; |
| 11 | \$password = \$post['password']; |
| 12 | \$user = \$username; |
| 13 | \$pass = \$password; |
| 14 | \$ccek = \$this->login_model->cek_login(\$user, \$pass); |
| 15 | if (\$cek->num_rows() > 0) { |
| 16 | foreach (\$cek->result() as \$ck) { |
| 17 | \$sess_data['username'] = \$ck->username; |
| 18 | // \$sess_data['email'] = \$ck->email; |
| 19 | \$sess_data['level'] = \$this->session->set_userdata(\$sess_data); |
| 20 | } |

Tabel 4. 93 Kode Program Fungsi Utama Login Sistem Informasi (Lanjutan)

| No | Kode Program |
|----|---|
| 34 | if (\$sess_data['level'] == 1) { |
| 35 | redirect('administrator/dashboard'); |
| 36 | } else if (\$sess_data['level'] == 2) |
| 37 | { |
| 38 | redirect('bkk/dashboard'); |
| 39 | } else if (\$sess_data['level'] == 3) |
| 40 | { |
| 41 | redirect('wakakur/dashboard'); |
| 42 | } else if (\$sess_data['level'] == 4) |
| 43 | { |
| 44 | redirect('guru/dashboard'); |
| 45 | } else if (\$sess_data['level'] == 5) |
| 46 | redirect('siswa/dashboard'); |
| 47 | } else { |
| 48 | \$this->session- |
| 49 | >set_flashdata('pesan', '<div class="alert alert- |
| 50 | danger alert-dismissible fade show" role="alert">Maaf |
| 51 | username atau password yang anda masukkan tidak |
| 52 | valid<button type="button" class="close" data- |
| 53 | dismiss="alert" aria-label="Close"><span aria- |
| 54 | hidden="true">×</button></div>'); |
| 55 | redirect('administrator/auth'); |
| 56 | } |
| 57 | } else { |
| 58 | \$this->session- |
| 59 | >set_flashdata('pesan', '<div class="alert alert- |
| 60 | danger alert-dismissible fade show" role="alert">Maaf |
| 61 | username atau password yang anda masukkan tidak |
| 62 | valid<button type="button" class="close" data- |
| 63 | dismiss="alert" aria-label="Close"><span aria- |
| 64 | hidden="true">×</button></div>'); |
| 65 | redirect('administrator/auth'); |

Kode diatas akan melakukan beberapa kali pengecekan dan verifikasi, apabila data yang dimasukkan oleh pengguna tidak lengkap maka sistem akan memunculkan informasi bahwa sebuah data perlu dilengkapi, apabila data yang dimasukkan oleh user tidak terdaftar di database maka sistem akan menampilkan informasi bahwa data tidak

terdaftar. Apabila data yang dimasukkan dinyatakan valid maka akan dilakukan pengecekan level akses yang dimiliki oleh *user*. Hasil akhir apabila *user* berhasil masuk akan diarahkan pada halaman *dashboard*.

3. Logout Sistem

Fungsi *logout* sistem memungkinkan pengguna untuk menghapus sesi informasi masuk pada peramban web yang digunakan.

Implementasi fungsi *logout* sistem dapat dilihat pada Tabel 4.94.

Tabel 4. 94 Kode Program Fungsi Utama Logout Sistem Informasi

| No | Kode Program |
|----|----------------------------------|
| 1 | public function logout() |
| 2 | { |
| 3 | \$this->session->sess_destroy(); |
| 4 | redirect('administrator/auth'); |
| 5 | } |

Kode diatas berfungsi untuk menghapus data *session login* dari pengguna, dan mengarahkan pengguna pada halaman *login*.

4. Pembacaan Daftar Data

Fungsi pembacaan daftar data memungkinkan pengguna untuk dapat melihat data dalam jumlah yang banyak, sesuai dengan kategori tertentu. Implementasi fungsi pembacaan daftar data terdiri atas dua bagian, yaitu melakukan *query* secara langsung pada tabel tertentu, serta menerapkan *JOIN table* untuk mendapatkan data pada *table* lain.

Tabel 4. 95 Kode Program Fungsi Ambil Daftar Data

| No | Kode Program |
|----|---------------------------------------|
| 1 | public function show_data() |
| 2 | { |
| 3 | return \$this->db->get('nama_tabel'); |
| 4 | } |

Kode pada Tabel 4. 95 berfungsi untuk menampilkan data dari database sesuai dengan nama tabel yang dispesifikasikan. Apabila data yang ingin dicari terdapat pada lebih dari satu tabel yang berelasi satu sama lain, maka contoh fungsi yang digunakan adalah seperti pada Tabel 4. 96.

Tabel 4. 96.

| No | Kode Program |
|----|--|
| 1 | public function show_data() |
| 2 | { |
| 3 | \$query = "SELECT jurusan.nama AS |
| 4 | nama_jurusan,kelas.id, kelas.kode_kelas, |
| | kelas.nama_kelas as nama_kelas, guru.nama as |
| | nama_guru, tahun_ajaran.tahun_ajaran, tahun_ajaran.id as id_ta, verifikasi.status FROM |
| | kelas JOIN jurusan ON kelas.jurusan = jurusan.id JOIN |
| | guru ON guru.NBM = kelas.wali_kelas JOIN tahun_ajaran |
| | ON kelas.id_tahun_ajaran = tahun_ajaran.id JOIN |
| | verifikasi ON verifikasi.id = kelas.status ORDER BY |
| | guru.NBM" |
| | return \$this->db->query(\$query); |
| | } |

5. Pembacaan Detail Data

Fungsi pembacaan detail data memungkinkan pengguna untuk dapat melihat data pada baris atau *id* yang spesifik. Contoh Implementasi fungsi pembacaan detail data dapat dilihat pada Tabel 4.97.

Tabel 4. 97 Kode Program Fungsi Utama Lihat Detail Data

| No | Kode Program |
|----|---|
| 1 | public function getDetailKelas(\$id) |
| 2 | { |
| 3 | \$sql = "SELECT kelas.id, kelas.kode_kelas, |
| 4 | kelas.nama_kelas, kelas.tingkat, guru.nama AS |
| 5 | wali_kelas, tahun_ajaran.tahun_ajaran FROM kelas JOIN |
| 6 | guru ON kelas.wali_kelas = guru.NBM JOIN tahun_ajaran |
| 7 | ON kelas.id_tahun_ajaran = tahun_ajaran.id WHERE |
| 8 | kelas.id = \$id"; |
| | return \$this->db->query(\$sql) ->row(); |
| | } |

Fungsi pembacaan detail data menggunakan parameter \$id untuk menentukan data yang dirujuk. Pada akhir *function* memanggil fungsi *row()* sehingga data yang diambil dari *database* akan berbentuk *array* dengan satu buah data, sehingga dalam menampilkan data pada *view*, perulangan tidak diperlukan lagi.

Universitas Brawijaya

6. Penambahan Data

Fungsi penambahan data memungkinkan pengguna untuk dapat menambahkan data baru pada *database*. Contoh implementasi fungsi penambahan data dapat dilihat pada Tabel 4. 98.

Tabel 4. 98 Kode Program Fungsi Utama *Input Data (Controller)*

| No | Kode Program |
|----|--|
| 1 | public function input_aksi() |
| 2 | { |
| 3 | \$this->_rules(); |
| 4 | if (\$this->form_validation->run() == FALSE) |
| 5 | \$this->input(); |
| 6 | } else { |
| 7 | \$data = array(|
| 8 | 'kode' => \$this->input->post('kode_pelajaran', TRUE), |
| 9 | 'nama' => \$this->input->post('nama', TRUE), |
| 10 | 'id_guru' => \$this->input->post('id_guru', TRUE), |
| 11 | 'jurusan' => \$this->input->post('jurusan', TRUE), |
| 12 |); |
| 13 | \$this->mata_pelajaran_model->input_data(\$data); |
| 14 | \$this->session->set_flashdata('pesan', '<div class="alert alert-danger alert-dismissible fade show" role="alert">Data berhasil ditambahkan<button type="button" class="close" data-dismiss="alert" aria-label="Close">×</button></div>'); |
| 15 | redirect('administrator/mata_pelajaran'); |
| 16 | } |
| 17 | } |

Semua data yang akan ditambahkan dalam *database* diproses melalui *form* dan dikirim melalui *method post*. Sebelum data dikirim akan dilakukan pengecekan data, apakah data yang dimasukkan pengguna memenuhi syarat atau tidak, apabila memenuhi syarat maka data yang dikirim melalui *method post* di *form* akan dijadikan satu dalam bentuk *array* pada variable \$data. Selanjutnya \$data akan diproses oleh fungsi pada bagian model yang digunakan untuk input data ke database. Contoh fungsi dalam model untuk melakukan input data ke database dapat dilihat pada Tabel 4. 99.

universitas brawijaya universitas brawijaya universitas brawijaya universitas brawijaya
 Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya Universitas Brawijaya
Tabel 4. 99 Kode Program Fungsi Utama Input Data (Model)

| No | Kode Program |
|----|---|
| 1 | public function input_data(\$data) |
| 2 | { |
| 3 | \$this->db->insert('nama_table', \$data); |
| 4 | } |

Kode diatas akan menambahkan data *array* yang dikirimkan oleh *controller* pada *table* yang diinginkan. Operasi ini hanya akan berhasil apabila terdapat kecocokan antara elemen *array* dengan struktur tabel yang dituju.

7. Pengubahan Data

Fungsi pengubahan data memungkinkan pengguna untuk dapat melakukan perubahan terhadap data yang sudah ada di *database*. Implementasi fungsi pengubahan data dapat dilihat pada Tabel 4. 100.

Tabel 4. 100 Kode Program Fungsi Utama Ubah Data (Controller)

| No | Kode Program |
|----|---|
| 1 | public function update_aksi() |
| 2 | { |
| 3 | \$id = \$this->input->post('id'); |
| 4 | \$kode_kelas = \$this->input->post('kode_kelas'); |
| 5 | \$nama_kelas = \$this->input->post('nama_kelas'); |
| 6 | \$tahun_ajaran = \$this->input->post('tahun_ajaran'); |
| 7 | \$wali_kelas = \$this->input->post('wali_kelas'); |
| 8 | \$id_jurusan = \$this->input->post('jurusan'); |
| 9 | \$status = \$this->input->post('status'); |
| 10 | \$data = array(|
| 11 | 'kode_kelas' => \$kode_kelas, |
| 12 | 'nama_kelas' => \$nama_kelas, |
| 13 | 'id_tahun_ajaran' => \$tahun_ajaran, |
| 14 | 'wali_kelas' => \$wali_kelas, |
| 15 | 'jurusan' => \$id_jurusan, |
| 16 | 'status' => \$status |
| 17 |); |
| 18 | \$where = array('id' => \$id); |
| 19 | \$this->kelas_model->update_data(\$where, |
| 20 | \$data, 'kelas'); |
| 21 | \$this->kelas_model->update_data(\$where, |
| 22 | \$data, 'kelas'); |

Tabel 4. 100 Kode Program Fungsi Utama Ubah Data (*Controller*) (Lanjutan)

| No | Kode Program |
|----|---|
| 23 | \$this->session->set_flashdata('pesan', ' <div class="alert alert-successs alert-dismissible fade show" role="alert">Data berhasil diupdate<button type="button" class="close" data-dismiss="alert" aria-label="Close">&times;</button></div>');</div> |
| 24 | |
| 25 | |
| 26 | |
| 27 | |
| 28 | } |

Semua data yang akan diproses melalui *form* dan dikirim melalui *method post*. Sebelum data dikirim akan dilakukan pengecekan data, apakah data yang dimasukkan pengguna memenuhi syarat atau tidak, apabila memenuhi syarat maka data yang dikirim melalui *method post* di *form* akan dijadikan satu dalam bentuk *array* pada variable \$data. Selanjutnya \$data akan diproses oleh fungsi pada bagian model yang digunakan untuk update data ke database. Contoh fungsi dalam model untuk melakukan update data ke database dapat dilihat pada Tabel 4. 101.

Tabel 4. 101 Kode Program Fungsi Utama Ubah Data (*Model*)

| No | Kode Program |
|----|---|
| 1 | public function update_data(\$where, \$data, \$table) |
| 2 | { |
| 3 | \$this->db->where(\$where); |
| 4 | \$this->db->update(\$table, \$data); |
| 5 | } |

Proses pengubahan data perlu merujuk pada data secara spesifik, sehingga perlu dispesifikasi dengan pasti, id unik dari data menggunakan *array* \$where, lalu dispesifikasi \$table yang merujuk pada tabel tempat data berada dan \$data yang merupakan data baru pengganti data lama.

8. Penghapusan Data

Fungsi penghapusan data memungkinkan pengguna untuk dapat melakukan penghapusan terhadap data yang sudah ada di *database*. Implementasi fungsi penghapusan data dapat dilihat pada Tabel 4. 102.

| No | Kode Program |
|----|--|
| 1 | public function hapus_data(\$where, \$table) |
| 2 | { |
| 3 | \$this->db->where(\$where); |
| 4 | \$this->db->delete(\$table); |
| 5 | } |

Proses penghapusan data perlu merujuk pada data secara spesifik, sehingga perlu dispesifikasi dengan pasti, id unik dari data menggunakan array \$where, lalu dispesifikasi \$table yang merujuk pada tabel tempat data berada.

9. Pengubahan Status Data

Fungsi pengubahan status data memungkinkan pengguna untuk dapat melakukan pengubahan terhadap status (ACC atau Tolak) data yang sudah ada di *database*. Implementasi fungsi pengubahan status data dapat dilihat pada Tabel 4. 103.

Tabel 4. 103 Kode Program Fungsi Utama ACC Data

| No | Kode Program |
|----|------------------------------------|
| 1 | public function setAcc(\$id) |
| 2 | { |
| 3 | \$query = "UPDATE prakerin |
| 4 | SET status = 2 WHERE id = \$id"; |
| 5 | return \$this->db->query(\$query); |

Kode diatas akan mengubah nilai pada kolom status yang ada pada tabel prakerin dengan nilai 2 yang merupakan kode untuk status ACC, apabila pengguna merasa masih terdapat kesalahan pada data maka pengguna dapat memilih pilihan tolak, sehingga kode yang dijalankan seperti pada Tabel 4. 104.

Tabel 4. 104 Kode Program Fungsi Utama Tolak Data

| No | Kode Program |
|----|------------------------------------|
| 1 | public function setTolak(\$id) |
| 2 | { |
| 3 | \$query = "UPDATE prakerin |
| 4 | SET status = 3 WHERE id = \$id"; |
| 5 | return \$this->db->query(\$query); |

Nilai 3 untuk kolom status merupakan kode yang menyatakan bahwa data yang dikirimkan statusnya oleh pengguna lain ditolak oleh pengguna saat ini.

10. Pengunggahan (*Upload*) Gambar

Karena tidak disimpan secara langsung (hanya informasi nama *file* yang disimpan) pada *database*, data dengan format gambar perlu diproses secara terpisah dengan data lain yang disimpan di *database* secara langsung. Implementasi fungsi *upload* gambar dapat dilihat pada Tabel 4. 105.

Tabel 4. 105 Kode Program Fungsi Utama Unggah Gambar

| No | Kode Program |
|----|---|
| 1 | private function _uploadImage() |
| 2 | { |
| 3 | \$config['upload_path'] = './assets/uploads'; |
| 4 | \$config['allowed_types'] = 'gif jpg png'; |
| 5 | \$config['file_name'] = \$this->NBM; |
| 6 | \$config['overwrite'] = true; |
| 7 | \$config['max_size'] = 2048; |
| 8 | |
| 9 | \$this->load->library('upload', \$config); |
| 10 | if (\$this->upload->do_upload('foto')){ |
| 11 | return \$this->upload->data("file_name"); |
| 12 | } |
| 13 | print_r(\$this->upload->display_errors()); |
| 14 | return "default.jpg"; |
| 15 | } |

Pada kode diatas, akan dilakukan konfigurasi terlebih dahulu terhadap gambar yang akan diunggah yang meliputi tempat gambar akan disimpan setelah diunggah, tipe yang dapat diunggah, nama *file* saat gambar diunggah, penimpaan *file* apabila terdapat perubahan terhadap data baru, dan ukuran maksimum dari file yang diunggah. Setelah itu akan dilakukan *load* sebuah *library* untuk mengunggah gambar sesuai dengan konfigurasi yang telah disepfikasikan, apabila pengunggahan gambar sukses maka nilai yang dikembalikan berupa *string* nama *file* yang telah diunggah, apabila proses pengunggahan gagal maka akan menampilkan pesan *eror* dan data yang dikembalikan berupa *string* nama file gambar bawaan (*default*).

11. Penghapusan Gambar

Fungsi penghapusan gambar memungkinkan data gambar akan ikut terhapus apabila data yang berkaitan dilakukan penghapusan. Implementasi penghapusan gambar dapat dilihat pada Tabel 4. 106.

Tabel 4. 106 Kode Program Fungsi Utama Hapus Data Gambar

| No | Kode Program |
|----|---|
| 1 | private function deleteImage(\$id) |
| 2 | { |
| 3 | \$guru = \$this->getById(\$id); |
| 4 | if (\$guru->foto != "default.jpg") { |
| 5 | \$filename = explode(".", \$guru->foto)[0]; |
| 6 | return array_map('unlink', glob(FCPATH . |
| 7 | "assets/uploads/\$filename.*")); |
| 8 | } |
| 9 | } |

Kode diatas akan mengambil data berdasarkan id untuk memperoleh informasi mengenai nama file gambar yang akan dihapus, selanjutnya apabila nama file gambar bukan merupakan nama *default* maka akan dilakukan penghapusan terhadap data gambar yang tersimpan di *web server*.

4.3.5 Implementasi Antarmuka

Bagian ini membahas mengenai hasil dari implementasi rancangan antarmuka yang telah dibuat pada tahap sebelumnya. Hasil implementasi antarmuka sistem informasi adalah sebagai berikut:

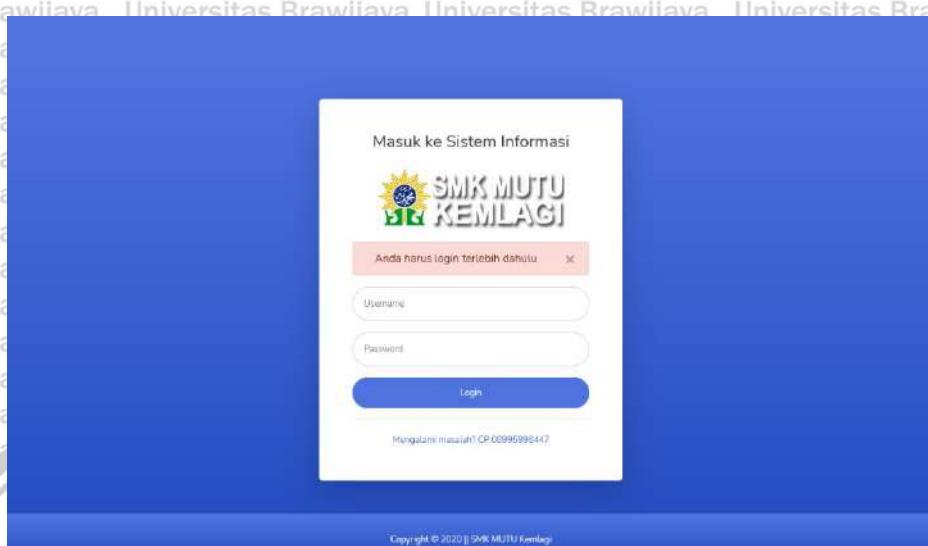
1. Login Sistem Informasi

Pada Gambar 4. 187 merupakan implementasi antarmuka untuk fungsi *login* sistem, yang memungkinkan aktor-aktor yang ada dapat melakukan akses pada fungsionalitas yang ada pada sistem informasi.

**Gambar 4. 187 Antarmuka Login Sistem**

2. Logout Sistem Informasi

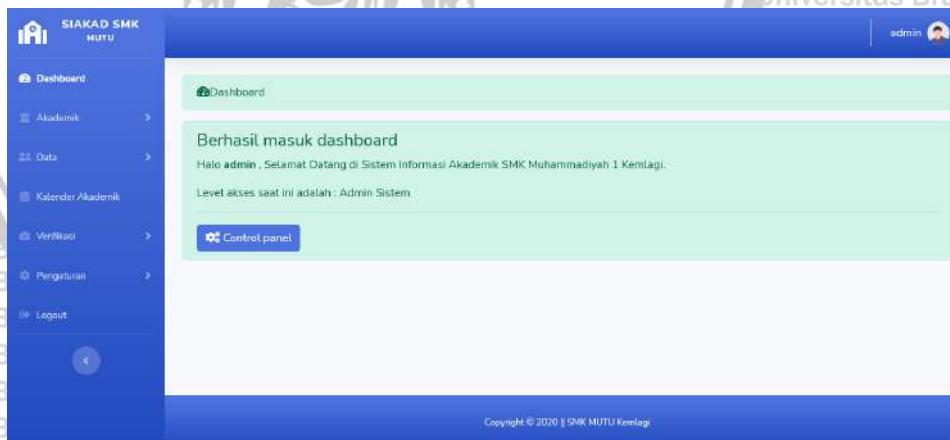
Pada Gambar 4. 188 merupakan implementasi antarmuka untuk fungsi *logout* sistem, yang memungkinkan aktor-aktor untuk keluar dari akses sistem informasi untuk sementara.



Gambar 4. 188 Antarmuka Logout Sistem

3. Halaman Dashboard

Pada Gambar 4. 189 merupakan implementasi antarmuka untuk halaman *dashboard*, yang memungkinkan aktor-aktor dapat mengakses fitur-fitur yang ada sesuai dengan level aksesnya.



Gambar 4. 189 Antarmuka Dashboard

4. Lihat Kalender Akademik

Pada Gambar 4. 190 merupakan implementasi antarmuka untuk fungsi lihat kalender akademik, yang memungkinkan aktor-aktor dapat mengakses kalender akademik pada tahun ajaran saat ini.

**Gambar 4. 190 Antarmuka Kalender Akademik****5. Lihat Daftar Kegiatan Akademik**

Pada Gambar 4. 191 merupakan implementasi antarmuka untuk fungsi lihat daftar kegiatan akademik, yang memungkinkan aktor (admin sistem) dapat mengakses daftar kegiatan akademik yang tersimpan.

The screenshot shows a blue-themed web application titled 'SIKAKAD SMK MUTU'. On the left sidebar, there are menu items: Dashboard, Akademik, Data, Kalender Akademik, Verifikasi, Pengeluaran, and Logout. The main content area is titled 'Kegiatan Akademik' and displays a table of academic activities:

| No | Nama Kegiatan | Tanggal Mulai | Tanggal Akhir | Tahun Ajaran | Status | Aksi |
|----|------------------------------|---------------|---------------|--------------|-------------------|------|
| 1 | Daftar Ulang Siswa | 20-05-2021 | 08-06-2021 | 2018/2019 | Menunggu Validasi | |
| 2 | Ujian Tengah Semester Ganjil | 20-05-2021 | 27-05-2021 | 2018/2019 | Diterima | |
| 3 | Ujian Akhir Semester | 19-05-2021 | 29-05-2021 | 2017/2018 | Menunggu Validasi | |

Copyright © 2020 | SMK MUTU Kertajati

Gambar 4. 191 Antarmuka Daftar Kegiatan Akademik**6. Tambah Data Kegiatan Akademik**

Pada Gambar 4. 192 merupakan implementasi antarmuka untuk fungsi tambah data kegiatan akademik, yang memungkinkan aktor (admin sistem) dapat melakukan penambahan data kegiatan akademik.

The screenshot shows a blue-themed user interface for an academic activity management system. On the left, a sidebar menu lists 'Dashboard', 'Akademik', 'Data', 'Kalender Akademik', 'Verifikasi', 'Pengaturan', and 'Logout'. The main content area has a title 'Form Tambah Data Kegiatan Akademik'. It contains input fields for 'Nama Kegiatan' (activity name), 'Tanggal Mulai' (start date), 'Tanggal Akhir' (end date), and 'Tahun Ajaran' (school year). A 'Simpan' (Save) button is at the bottom.

Gambar 4. 192 Antarmuka Tambah Kegiatan Akademik

7. Ubah Data Kegiatan Akademik

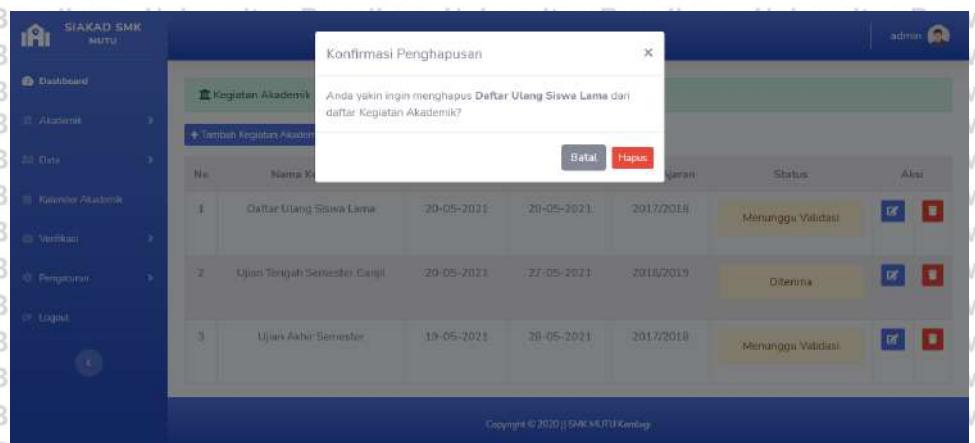
Pada Gambar 4. 193 merupakan implementasi antarmuka untuk fungsi ubah data kegiatan akademik, yang memungkinkan aktor (admin sistem) dapat melakukan perubahan terhadap data kegiatan akademik.

This screenshot shows the update form for academic activities. The sidebar menu is identical to the previous one. The main form is titled 'Update Data Kegiatan Akademik'. It features a dropdown menu for 'ID Jurusan Nama Kegiatan' containing 'Daftar Utang Siswa Lama'. Other fields include 'Tanggal Mulai' (20/05/2021), 'Tanggal Akhir' (20/05/2021), and 'Tahun Ajaran' (2017/2018). A 'Simpan' (Save) button is at the bottom.

Gambar 4. 193 Antarmuka Ubah Data Kegiatan Akademik

8. Hapus Data Kegiatan Akademik

Pada Gambar 4. 194 merupakan implementasi antarmuka untuk fungsi tambah data kegiatan akademik, yang memungkinkan aktor (admin sistem) dapat melakukan penghapusan data kegiatan akademik.



Gambar 4. 194 Antarmuka Hapus Data Kegiatan Akademik

9. Lihat Daftar Jurusan

Pada Gambar 4. 195 merupakan implementasi antarmuka untuk fungsi lihat daftar jurusan, yang memungkinkan aktor (admin sistem) dapat mengakses daftar kegiatan jurusan yang tersimpan.



Gambar 4. 195 Antarmuka Daftar Jurusan

10. Tambah Data Jurusan

Pada Gambar 4. 196 merupakan implementasi antarmuka untuk fungsi tambah data jurusan, yang memungkinkan aktor (admin sistem) dapat melakukan penambahan data jurusan.



Gambar 4. 196 Antarmuka Tambah Data Jurusan

11. Ubah Data Jurusan

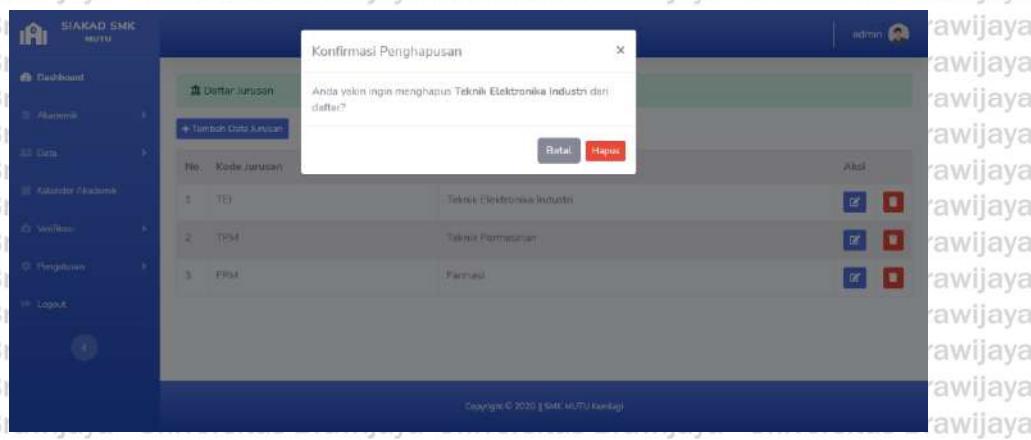
Pada Gambar 4. 197 merupakan implementasi antarmuka untuk fungsi ubah data jurusan, yang memungkinkan aktor (admin sistem) dapat melakukan perubahan terhadap data jurusan.



Gambar 4. 197 Antarmuka Ubah Data Jurusan

12. Hapus Data Jurusan

Pada Gambar 4. 198 merupakan implementasi antarmuka untuk fungsi hapus data jurusan, yang memungkinkan aktor (admin sistem) dapat melakukan penghapusan data jurusan.



13. Lihat Data Kelas

Pada Gambar 4. 199 merupakan implementasi antarmuka untuk fungsi lihat daftar data kelas, yang memungkinkan aktor (admin sistem) dapat melihat daftar kelas yang tersimpan.

| No | Kode Kelas | Nama Kelas | Wali Kelas | Jurusan | Tahun Ajaran | Status | Akta | |
|----|------------|--|-------------|-----------------------------|--------------|-------------------|------|--|
| 1 | XII-TEI-1 | Dua Belas Teknik Elektronika Industri Satu | Oemar Patek | Teknik Elektronika Industri | 2017/2018 | Menunggu Validasi | | |
| 2 | XI-FRM-2 | Sebelas Farmasi Dua | Oemar Bakri | Farmasi | 2018/2019 | Menunggu Validasi | | |

Gambar 4. 199 Antarmuka Daftar Kelas**14. Details Data Kelas**

Pada Gambar 4. 200 merupakan implementasi antarmuka untuk fungsi lihat detail data kelas, yang memungkinkan aktor (admin sistem, guru dan siswa) dapat melihat informasi kelas secara lebih detail.

| Informasi Kelas | | Daftar Siswa | |
|-----------------|-----------------|--------------|------------------------------|
| No | Nama | No | Nama |
| 1 | Imam Buchon | 1 | Muhammad Taftian Ilham Akbar |
| 2 | Salsabila Putri | 2 | Salsabila Putri |
| 3 | | | |

Gambar 4. 200 Antarmuka Detail Data Kelas**15. Tambah Data Kelas**

Pada Gambar 4. 201 merupakan implementasi antarmuka untuk fungsi tambah data kelas, yang memungkinkan aktor (admin sistem) dapat melakukan penambahan data kelas.

Form Tambah Kelas

Kode Kelas: XII-TEI-1

Nama Kelas: Dua Belas Teknik Elektronika Industri Seturi

Wali Kelas: Demar Bakri

Tahun Ajaran: 2017/2018

Id Jurusan: Teknik Elektronika Industri

Simpan

Gambar 4. 201 Antarmuka Tambah Data Kelas

16. Ubah Data Kelas

Pada Gambar 4. 202 merupakan implementasi antarmuka untuk fungsi ubah data kelas, yang memungkinkan aktor (admin sistem) dapat melakukan perubahan terhadap data kelas.

Form Update Kelas

Kode Kelas: XII-TEI-1

Nama Kelas: Dua Belas Teknik Elektronika Industri Seturi

Wali Kelas: Demar Bakri

Tahun Ajaran: 2017/2018

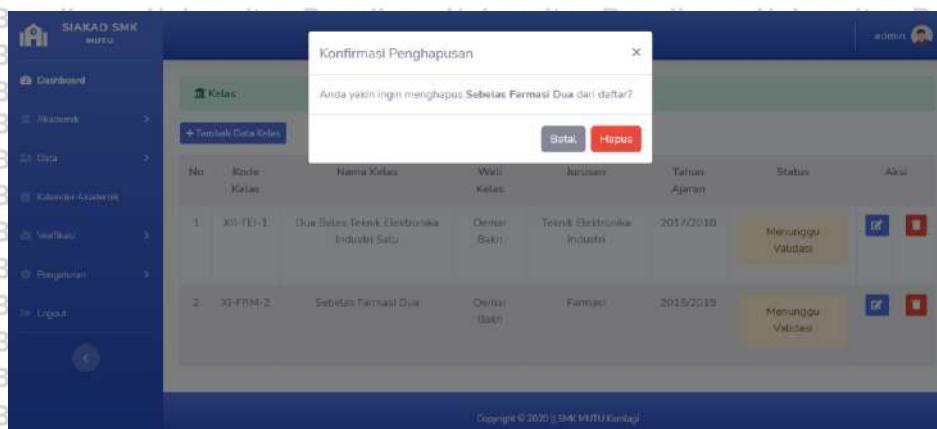
Id Jurusan: Teknik Elektronika Industri

Perbarui

Gambar 4. 202 Antarmuka Ubah Data Kelas

17. Hapus Data Kelas

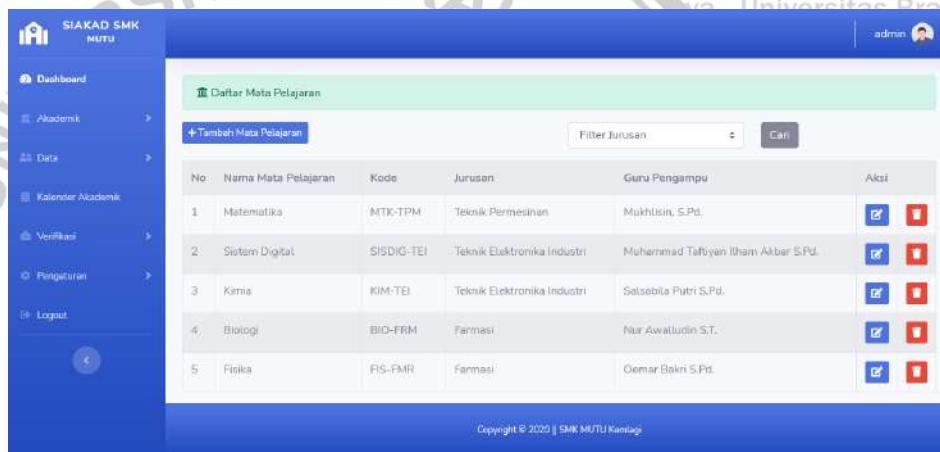
Pada Gambar 4. 203 merupakan implementasi antarmuka untuk fungsi hapus data kelas, yang memungkinkan aktor (admin sistem) dapat melakukan penghapusan data kelas.



Gambar 4. 203 Antarmuka Hapus Data Kelas

18. Lihat Data Mata Pelajaran

Pada Gambar 4. 204 merupakan implementasi antarmuka untuk fungsi lihat daftar data mata pelajaran, yang memungkinkan aktor (admin sistem) dapat melihat daftar mata pelajaran yang tersimpan.



Gambar 4. 204 Antarmuka Data Mata Pelajaran

19. Filter Mata Pelajaran

Pada Gambar 4. 205 merupakan implementasi antarmuka untuk filter data jadwal pelajaran, yang memungkinkan aktor (admin sistem) dapat melihat daftar mata pelajaran berdasarkan jurusan yang dipilih.



Gambar 4. 205 Antarmuka Filter Mata Pelajaran

20. Tambah Data Mata Pelajaran

Pada Gambar 4. 206 merupakan implementasi antarmuka untuk fungsi tambah data mata pelajaran, yang memungkinkan aktor (admin sistem) dapat melakukan penambahan data mata pelajaran.

The screenshot shows a form titled 'Form Tambah Data Mata Pelajaran'. It includes fields for 'Kode Mata Pelajaran' (with placeholder 'Masukkan kode pelajaran'), 'Nama Mata Pelajaran' (with placeholder 'Masukkan nama mata pelajaran'), 'Guru Pengampu' (with placeholder 'Pilihan Guru'), 'Jurusan' (with placeholder 'Pilihan Jurusan'), and a 'Simpan' button at the bottom. The sidebar on the left is identical to the one in the previous screenshot.

Copyright © 2020 | SMK MUTU Kemagi

Gambar 4. 206 Antarmuka Tambah Data Mata Pelajaran

21. Ubah Data Mata Pelajaran

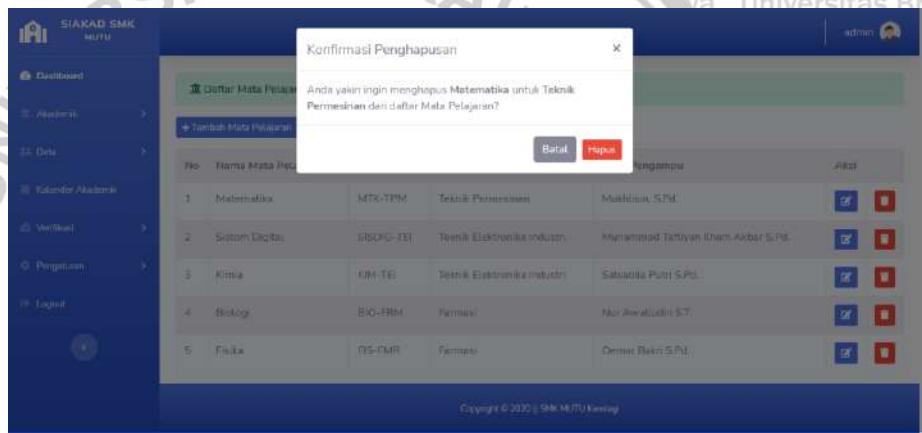
Pada Gambar 4. 207 merupakan implementasi antarmuka untuk fungsi ubah data mata pelajaran, yang memungkinkan aktor (admin sistem) dapat melakukan perubahan terhadap data mata pelajaran.



Gambar 4. 207 Antarmuka Ubah Data Mata Pelajaran

22. Hapus Data Mata Pelajaran

Pada Gambar 4. 208 merupakan implementasi antarmuka untuk fungsi hapus data mata pelajaran, yang memungkinkan aktor (admin sistem) dapat melakukan penghapusan data mata pelajaran.



Gambar 4. 208 Antarmuka Hapus Data Mata Pelajaran

23. Lihat Data Jadwal Pelajaran

Pada Gambar 4. 209 merupakan implementasi antarmuka untuk fungsi lihat daftar data jadwal pelajaran, yang memungkinkan aktor (admin sistem) dapat melihat daftar jadwal pelajaran yang tersimpan.



Gambar 4. 209 Antarmuka Daftar Jadwal Pelajaran

24. Lihat Detail Jadwal Pelajaran

Pada Gambar 4. 210 merupakan implementasi antarmuka untuk fungsi lihat detail jadwal pelajaran untuk aktor siswa dan guru yang memungkinkan para aktor tersebut dapat melihat jadwal pelajaran kelas.

The screenshot shows a table titled 'Detail Jadwal Pelajaran' (Detailed Class Schedule) for a week. It has four sections: Senin, Selasa, Rabu, and Kamis. Each section has two tables: 'Senin' and 'Selasa' share one table; 'Rabu' and 'Kamis' share another. The columns are 'Waktu' and 'Pelajaran'.

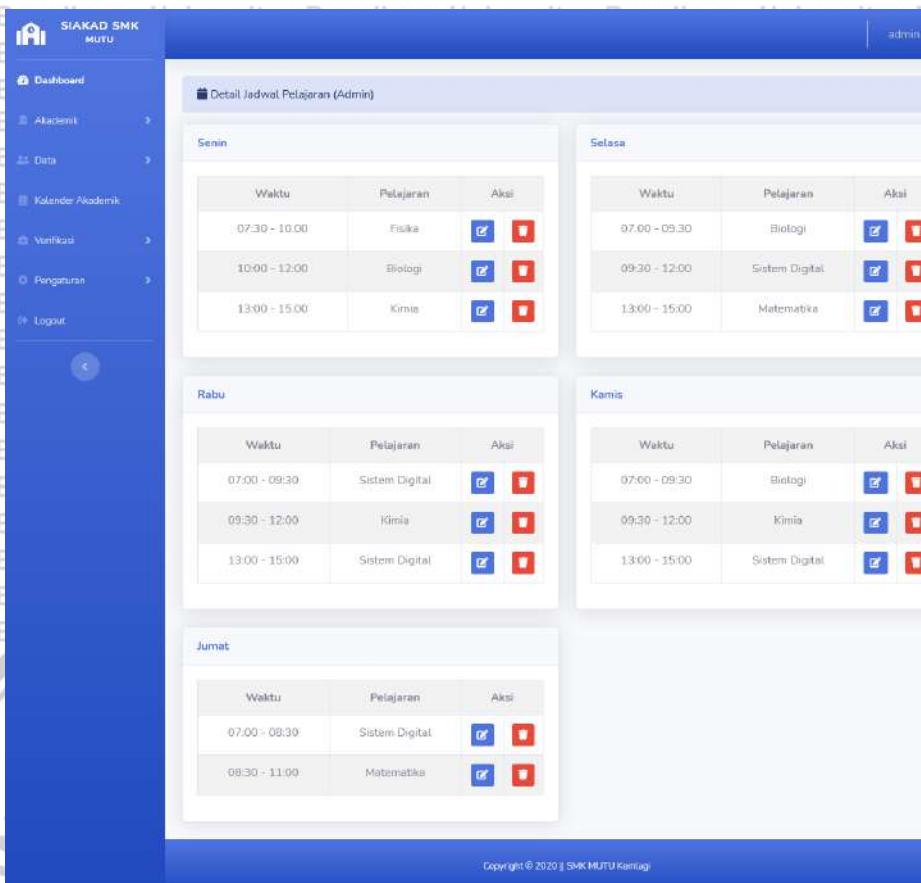
| Senin | | Selasa | |
|---------------|-----------|---------------|----------------|
| Waktu | Pelajaran | Waktu | Pelajaran |
| 07:30 - 10:00 | Fisika | 07:00 - 09:30 | Biologi |
| 10:00 - 12:00 | Biologi | 09:30 - 12:00 | Sistem Digital |
| 13:00 - 15:00 | Kimia | 13:00 - 15:00 | Matematika |

| Rabu | | Kamis | |
|---------------|----------------|---------------|----------------|
| Waktu | Pelajaran | Waktu | Pelajaran |
| 07:00 - 09:30 | Sistem Digital | 07:00 - 09:30 | Biologi |
| 09:30 - 12:00 | Kimia | 09:30 - 12:00 | Kimia |
| 13:00 - 15:00 | Sistem Digital | 13:00 - 15:00 | Sistem Digital |

| Jumat | |
|---------------|----------------|
| Waktu | Pelajaran |
| 07:00 - 08:30 | Sistem Digital |
| 08:30 - 11:00 | Matematika |

Gambar 4. 210 Antarmuka Detail Jadwal Pelajaran

Pada Gambar 4. 211 merupakan implementasi antarmuka untuk fungsi detail data jadwal pelajaran untuk aktor admin sistem yang memungkinkan aktor tersebut dapat melihat jadwal pelajaran dengan pilihan aksi terhadap data jadwal.



Gambar 4. 211 Antarmuka Detail Jadwal Pelajaran (Admin)

25. Tambah Data Jadwal Pelajaran

Pada Gambar 4. 212 merupakan implementasi antarmuka untuk fungsi tambah data jadwal pelajaran, yang memungkinkan aktor (admin sistem) dapat melakukan penambahan data jadwal pelajaran.

Pilih Kelas
Pilihan Kelas

Pilih Hari
Pilih Hari

Pilih Jam
Pilih Jam

Pilih Mata Pelajaran
Pilihan Mata Pelajaran

Gambar 4. 212 Antarmuka Tambah Data Jadwal Pelajaran

26. Ubah Data Jadwal Pelajaran

Pada Gambar 4. 213 merupakan implementasi antarmuka untuk fungsi ubah data jadwal pelajaran, yang memungkinkan aktor (admin sistem) dapat melakukan perubahan terhadap data jadwal pelajaran.

Form Update Data Jadwal

Pilih Kelas: Dua Belas Farmasi Satu

Pilihan Hari: Senin

Pilihan Jam: 07:00 - 08:30

Pilihan Mata Pelajaran: MTK-TPM - Matematika

Simpan

Gambar 4. 213 Antarmuka Ubah Data Jadwal Pelajaran

27. Hapus Data Jadwal Pelajaran

Pada Gambar 4. 214 merupakan implementasi antarmuka untuk fungsi hapus data jadwal pelajaran, yang memungkinkan aktor (admin sistem) dapat melakukan penghapusan data jadwal pelajaran.

Konfirmasi Penghapusan

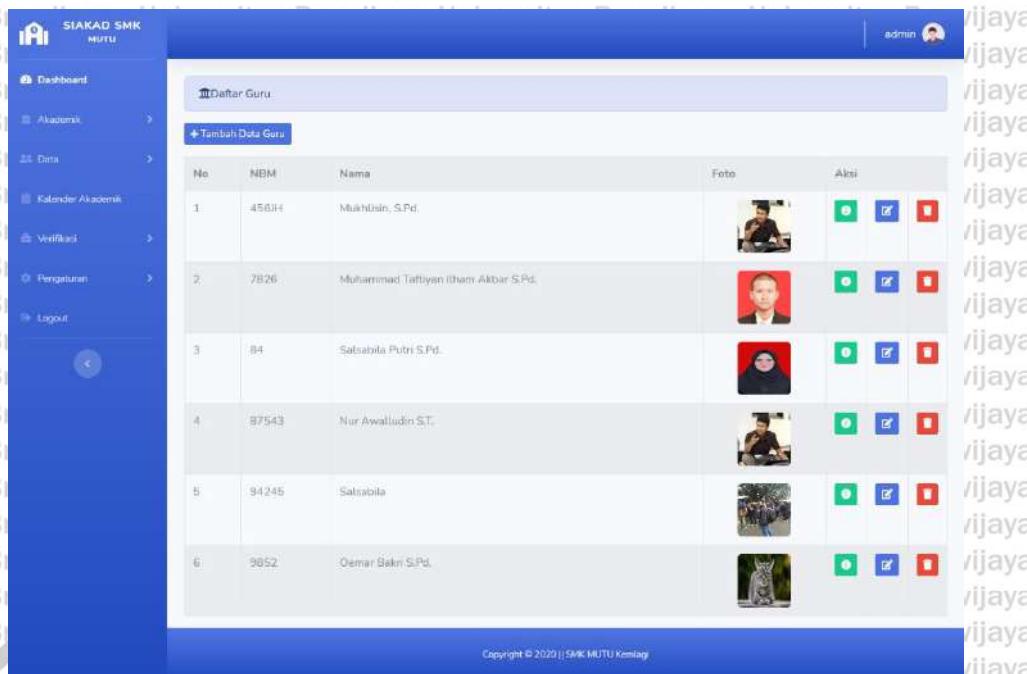
Anda yakin ingin menghapus Fisika dari daftar Jadwal Pelajaran?

Batalkan **Hapus**

Gambar 4. 214 Antarmuka Hapus Data Jadwal Pelajaran

28. Lihat Daftar Guru

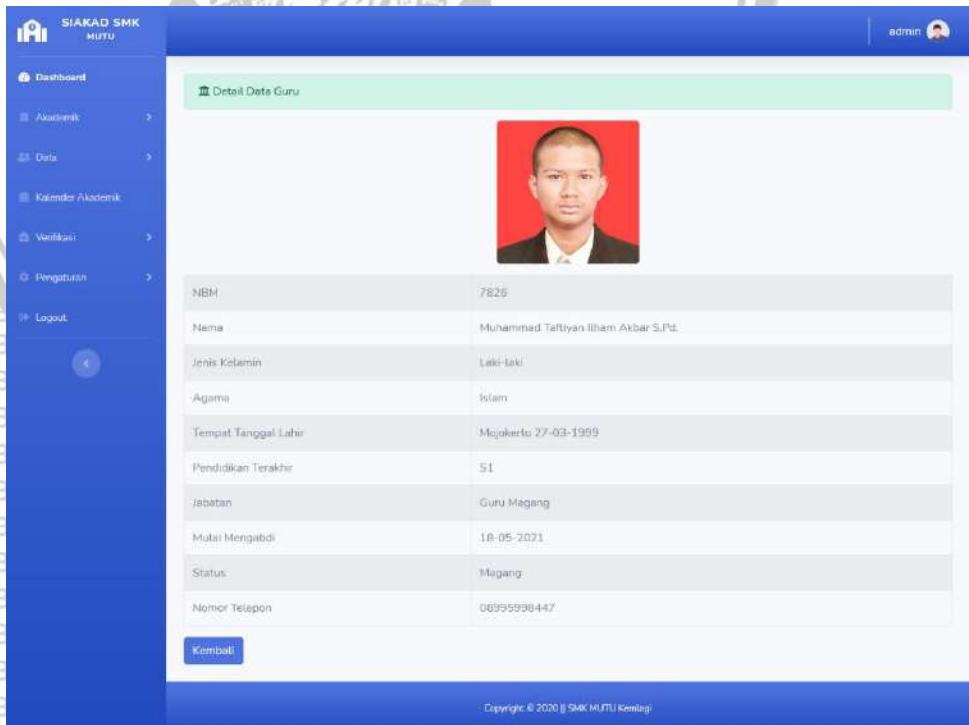
Pada Gambar 4. 215 merupakan implementasi antarmuka untuk fungsi lihat daftar data guru, yang memungkinkan aktor (admin sistem) dapat melihat daftar data guru yang tersimpan.



Gambar 4. 215 Antarmuka Daftar Guru

29. Lihat Detail Data Guru

Pada Gambar 4. 216 merupakan implementasi antarmuka untuk fungsi lihat detail data guru, yang memungkinkan aktor (admin sistem) dapat melihat informasi guru secara lebih detail.



Gambar 4. 216 Antarmuka Detail Data Guru

30. Tambah Data Guru

Pada Gambar 4. 217 merupakan implementasi antarmuka untuk fungsi tambah data guru, yang memungkinkan aktor (admin sistem) dapat melakukan penambahan guru.

Gambar 4. 217 Antarmuka Tambah Data Guru

31. Ubah Data Guru

Pada Gambar 4. 218 merupakan implementasi antarmuka untuk fungsi ubah data guru yang memungkinkan aktor (admin sistem) dapat melakukan perubahan terhadap data guru.

Form Update Data Guru

NBM
7826

Nama Guru
Muhammad Taftiyen Iham Akbar S.Pd.

Jenis Kelamin
Laki-laki

Agama
Islam

Alamat
Mengelo Selatan RT 2 RW 11

Tempat Lahir
Mojokerto

Tanggal Lahir
27/03/1999

Pendidikan Terakhir
S1

Jabatan/Peran
Guru Magang

Awal Mengabdi
18/05/2021

Status
Magang

Nomor Telepon
08995998447

Pilih Foto
Teksuri... Tidak ada berkas dipilih.

Simpan

Copyright © 2020 | JI SMK MUTU Kediri

Gambar 4. 218 Antarmuka Ubah Data Guru

32. Hapus Data Guru

Pada Gambar 4. 219 merupakan implementasi antarmuka untuk fungsi hapus data guru, yang memungkinkan aktor (admin sistem) dapat melakukan penghapusan data guru.

Konfirmasi Penghapusan

Anda yakin ingin menghapus data bapak/bu Demar Bakri S.Pd. dari daftar Guru?

Batal Hapus

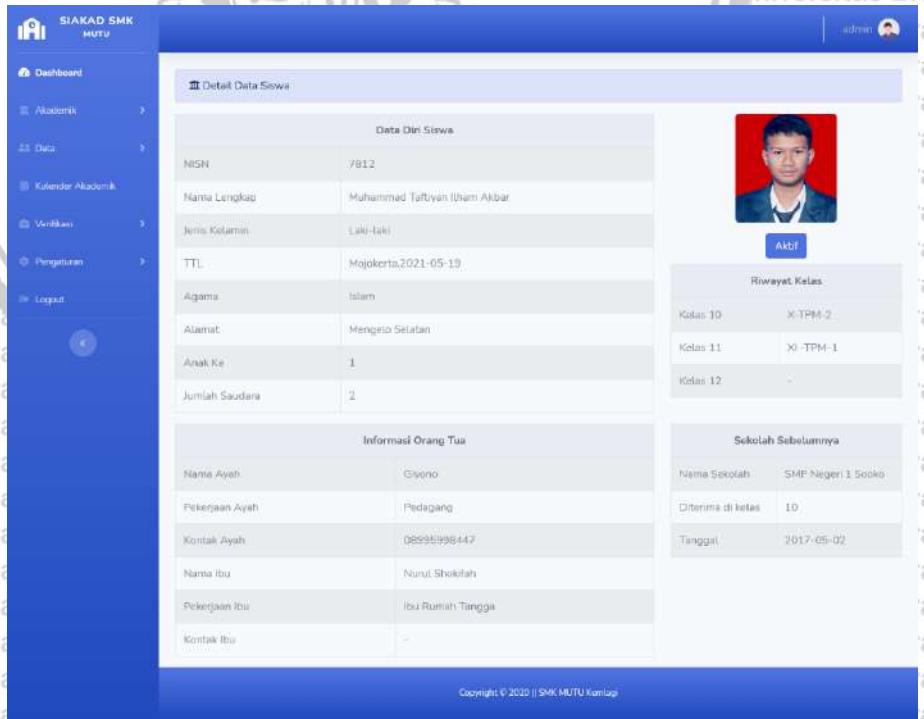
| No | NBM | Poto | Aksi |
|----|-------|------------------------------------|------------------|
| 1 | 48848 | Muhammad Taftiyen Iham Akbar S.Pd. | [Action Buttons] |
| 2 | 7826 | Muhammad Taftiyen Iham Akbar S.Pd. | [Action Buttons] |
| 3 | 94 | Setiajaya Putri S.Pd. | [Action Buttons] |
| 4 | 82543 | Nur Azzizatun S.T. | [Action Buttons] |

33. Lihat Daftar Siswa

Pada Gambar 4. 220 merupakan implementasi antarmuka untuk fungsi lihat daftar data siswa, yang memungkinkan aktor (admin sistem) dapat melihat daftar data siswa yang tersimpan.

**Gambar 4. 220 Antarmuka Daftar Siswa****34. Lihat Detail Siswa**

Pada Gambar 4. 221 merupakan implementasi antarmuka untuk fungsi lihat detail data siswa, yang memungkinkan aktor (admin sistem) dapat melihat informasi siswa secara lebih detail.

**Gambar 4. 221 Antarmuka Detail Data Siswa**

35. Cari Data Siswa**Gambar 4. 222 Antarmuka Cari Data Siswa****36. Tambah Data Siswa**

Pada Gambar 4. 223 merupakan implementasi antarmuka untuk fungsi tambah data siswa, yang memungkinkan aktor (admin sistem) dapat melakukan penambahan data siswa.

The screenshot shows a blue-themed web application interface. On the left is a sidebar with a navigation menu:

- SIAKAD SMK MUTU**
- Dashboard**
- Akademik**
 - Data
 - Kalender Akademik
- Verifikasi**
- Pengaturan**
- Logout**

The main content area is titled "Form Tambah Data Siswa". It contains the following fields:

- NISN: Masukkan NISN siswa
- Nama Lengkap Siswa: Masukkan nama lengkap siswa
- Jenis Kelamin: Pilih Jenis Kelamin
- Agama: Pilih Agama
- Alamat: Masukkan alamat
- Tempat Lahir: Masukkan tempat lahir
- Tanggal Lahir: hh/bb/ttt
- Analik: Masukkan nomor kelahiran siswa dalam keluarga
- Jumlah Seudara: Masukkan jumlah saudara siswa
- Pilih Foto: Tersedia... Tidak ada berkas dipilih.
- Nama Ayah: Masukkan nama ayah dari siswa yang bersangkutan
- Pekerjaan Ayah: Masukkan pekerjaan ayah dari siswa yang bersangkutan
- Kontak/Telepon Ayah: Masukkan kontak/telepon ayah dari siswa yang bersangkutan
- Nama Ibu: Masukkan nama ibu dari siswa yang bersangkutan
- Pekerjaan Ibu: Masukkan pekerjaan ibu dari siswa yang bersangkutan
- Kontak/Telepon Ibu: Masukkan kontak/telepon ibu dari siswa yang bersangkutan
- Sekolah sebelumnya: Masukkan nama sekolah siswa sebelum masuk ke SMK Muhammadiyah 1 Kemlagi
- Otentica dikelas: Masukkan jenjang penerimaan siswa
- Tanggal Penerimaan: hh/bb/ttt
- Simpan**

At the bottom right of the form area, there is a small copyright notice: Copyright © 2020 || SMK MUTU Kemlagi.

Gambar 4. 223 Antarmuka Tambah Data Siswa**37. Ubah Data Siswa**

Pada Gambar 4. 224 merupakan implementasi antarmuka untuk fungsi ubah data siswa, yang memungkinkan faktor (admin sistem) dapat melakukan perubahan terhadap data siswa.

Form Update Data Siswa

NISN
7812

Nama Lengkap Siswa
Muhammad Taftiyah Ilham Akber

Jenis Kelamin
Laki-Laki

Agama
Islam

Alamat
Mengelo Selatan RT 2 RW 11 Mojokerto

Tempat Lahir
Mojokerto

Tanggal Lahir
hh / bb / tttt

Anak Ke
1

Jumlah Seudara
2

Pilih Foto
Telusuri... Tidak ada berkas dipilih.

Nama Ayah
Gayono

Pekerjaan Ayah
Pedagang

Kontak/Telepon Ayah
08995998447

Nama Ibu
Nurul Sholehah

Pekerjaan Ibu
Ibu Rumah Tangga

Kontak/Telepon ibu
-

Sekolah sebelumnya
SMP 1 Soekarno

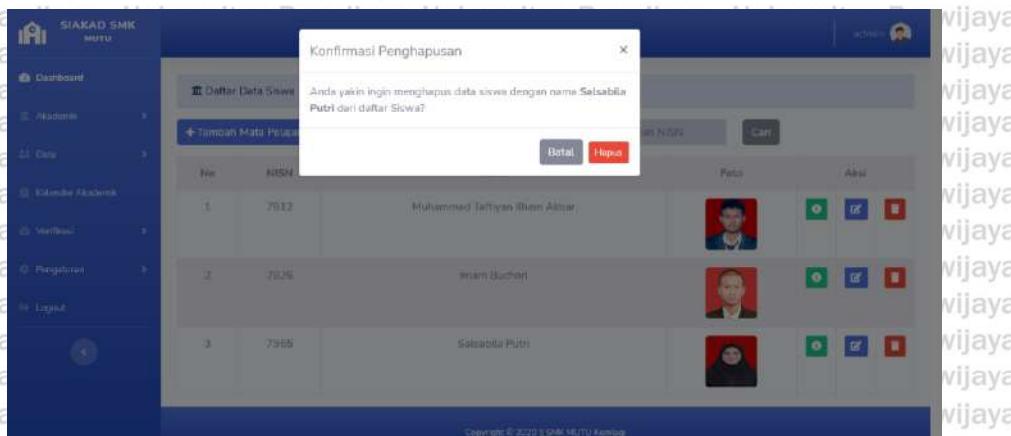
Diterima di kelas
11

Tanggal Penerimaan
hh / bb / tttt

Simpan

Gambar 4. 224 Antarmuka Ubah Data Siswa**38. Hapus Data Siswa**

Pada Gambar 4. 225 merupakan implementasi antarmuka untuk fungsi hapus data siswa, yang memungkinkan aktor (siswa) dapat melakukan penghapusan data siswa.



Gambar 4. 225 Antarmuka Hapus Data Siswa

39. Lihat Daftar Mitra Industri

Pada Gambar 4. 226 merupakan implementasi antarmuka untuk fungsi lihat daftar data mitra industri, yang memungkinkan aktor (admin sistem) dapat melihat daftar data mitra industri yang tersimpan.

| No. | Nama Industri | Jenis Industri | Alamat | Aksi |
|-----|--|--------------------|---|---|
| 1 | Fixtoner | IT | Mengelo Selatan RT 02/RW 11 | <input checked="" type="checkbox"/> Delete |
| 2 | PT Mertex Indonesia | Garmen | Jl. By Pass Mojokerto No.Desa. Lengkong, Kec. Mojoanyar; Mojokerto, Jawa Timur 61364 | <input checked="" type="checkbox"/> Delete |
| 3 | PT. Ajinomoto Indonesia, Tbk | Bahan Makanan | Jl. Raya Mirip No.110, Gedong, Mirip, Kec. Jetis, Mojokerto, Jawa Timur 61352 | <input checked="" type="checkbox"/> Delete |
| 4 | PT. Yakult Indonesia Persada Ngoko-Mojokerto | Minuman | Ngoro Industri, Ngoro, Jarang Sari, Lolawang, Kec. Ngoro, Mojokerto, Jawa Timur 61385 | <input checked="" type="checkbox"/> Delete |
| 5 | PT. Surabaya Autocomp Indonesia | Peralatan Otomotif | Ngoro Industrial Park Kaw T-1, Jarang Sari, Lolawang, Kec. Ngoro, Mojokerto, Jawa Timur 61385 | <input checked="" type="checkbox"/> Delete |

Gambar 4. 226 Antarmuka Daftar Mitra Industri

40. Tambah Data Mitra Industri

Pada Gambar 4. 227 merupakan implementasi antarmuka untuk fungsi tambah data mitra industri, yang memungkinkan aktor (admin BKK) dapat melakukan penambahan data mitra industri.

The screenshot shows a blue-themed web interface for adding industry partners. On the left is a sidebar with navigation links: Dashboard, Mitra Industri (selected), Validasi, Kalendar Akademik, and Logout. The main content area has a header 'Form Mitra Industri'. It contains three input fields: 'Nama Industri' (Name of Industry) with placeholder 'Masukkan nama industri', 'Jenis Industri' (Industry Type) with placeholder 'Masukkan jenis industri', and 'Alamat' (Address) with placeholder 'Masukkan alamat industri'. A 'Simpan' (Save) button is at the bottom.

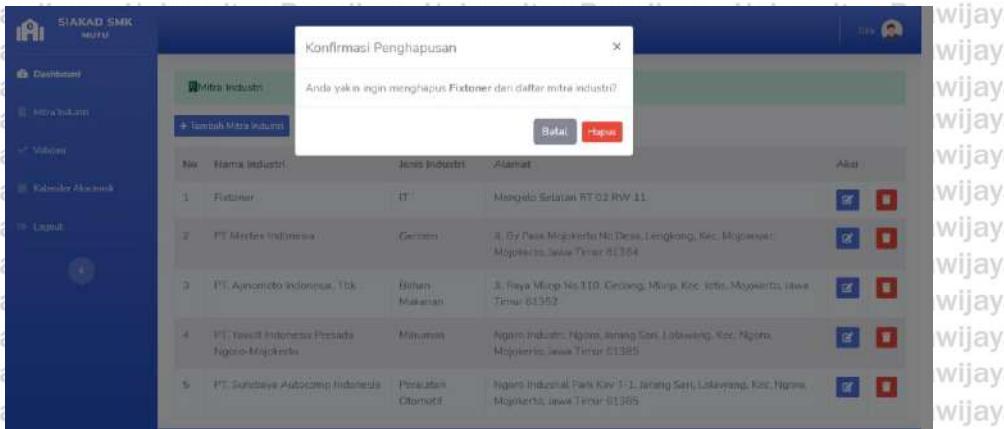
Gambar 4. 227 Antarmuka Tambah Mitra Industri**41. Ubah Data Mitra Industri**

Pada Gambar 4. 228 merupakan implementasi antarmuka untuk fungsi ubah data mitra industri, yang memungkinkan aktor (admin BKK) dapat melakukan perubahan terhadap data mitra industri.

This screenshot shows the update form for industry partners. The sidebar and header are identical to the add form. The main content area has a header 'Update Mitra Industri'. It contains four input fields: 'ID Mitra' (ID Partner) with placeholder 'Fixtner', 'Nama Industri' (Name of Industry) with placeholder 'Fixtner', 'Jenis Industri' (Industry Type) with placeholder 'IT', and 'Alamat' (Address) with placeholder 'Mengelo Selatan RT 02 RW 11'. A 'Simpan' (Save) button is at the bottom.

Gambar 4. 228 Antarmuka Ubah Data Mitra Industri**42. Hapus Data Mitra Industri**

Pada Gambar 4. 229 merupakan implementasi antarmuka untuk fungsi hapus data mitra industri, yang memungkinkan aktor (admin BKK) dapat melakukan penghapusan data mitra industri.

**Gambar 4. 229 Antarmuka Hapus Data Mitra Industri****43. Verifikasi Prakerin Siswa**

Pada Gambar 4. 230 merupakan implementasi antarmuka untuk fungsi verifikasi prakerin siswa, yang memungkinkan aktor (admin BKK) dapat melakukan verifikasi terhadap pengajuan penuntasan prakerin siswa.

The screenshot shows a table titled "Daftar Pengajuan Prakerin" (List of Work Placement Applications) displaying three entries:

| No | NISN | Nama | Tempat Prakerin | Alamat | Deskripsi | Aksi |
|----|-------|------------------------------|---------------------------------|--|---|------|
| 1 | 7812 | Muhammad Taftiyah Iham Akbar | PT Ajinomoto Indonesia | Jl. Raya Mirip No.110, Gedong, Mirip, Kec. Jetis, Mojokerto, Jawa Timur 61352 | Mulai bulan Desember 2020 berakhir pada tanggal 12 Februari 2021. | |
| 2 | 53624 | Taufiqi Iham | PT. Surabaya Autocomp Indonesia | Ngoro Industri, Park Kav 1-1, Jaring Sari, Lalawang, Kec. Ngoro, Mojokerto, Jawa Timur 61385 | Mulai bulan Desember 2020 berakhir pada tanggal 12 Februari 2021. | |
| 3 | 7812 | Muhammad Taftiyah Iham Akbar | PT Ajinomoto Indonesia | Jl. Raya Mirip No.110, Gedong, Mirip, Kec. Jetis, Mojokerto, Jawa Timur 61352 | Mulai bulan Desember 2020 berakhir pada tanggal 14 Februari 2021. | |

Gambar 4. 230 Antarmuka Verifikasi Prakerin Siswa**44. Verifikasi Penempatan Wali Kelas**

Pada Gambar 4. 231 merupakan implementasi antarmuka untuk fungsi verifikasi penempatan wali kelas, yang memungkinkan aktor (waka kurikulum) dapat melakukan verifikasi terhadap penempatan wali kelas.

| Daftar Penempatan Wali Kelas | | | | | | | |
|------------------------------|------------|--|------------------------------------|-----------------------------|--------------|-------------------|---|
| No | Kode Kelas | Nama Kelas | Wali Kelas | Jurusan | Tahun Ajaran | Status | Aksi |
| 1 | XII-TEI-1 | Dua Belas Teknik Elektronika Industri Satu | Oemar Bekti S.Pd. | Teknik Elektronika Industri | 2017/2018 | Menunggu Validasi | <button>ACC</button> <button>Tolak</button> |
| 2 | XII-FRM-1 | Dua Belas Formasi Satu | Muhammad Taffyan Ilham Akbar S.Pd. | Farmasi | 2020/2021 | Menunggu Validasi | <button>ACC</button> <button>Tolak</button> |
| 3 | XO-TPM-1 | Sembelas Teknik Permesinan Satu | Nur Awaludin S.T. | Teknik Permesinan | 2020/2021 | Menunggu Validasi | <button>ACC</button> <button>Tolak</button> |

Gambar 4. 231 Antarmuka Verifikasi Wali Kelas**45. Verifikasi Kalender Akademik**

Pada Gambar 4. 232 merupakan implementasi antarmuka untuk fungsi verifikasi kalender akademik, yang memungkinkan aktor (waka kurikulum) dapat melakukan verifikasi terhadap penerbitan kalender akademik.

| Daftar Rencana Waktu Pelaksanaan Akademik | | | | |
|---|-------------------------------------|---------------------------|--------------|---|
| No | Name Kegiatan | Pelaksanaan | Tahun Ajaran | Aksi |
| 1 | Daftar Ulang Siswa Lama | 07-01-2021 s/d 14-01-2021 | 2020/2021 | <button>ACC</button> <button>Tolak</button> |
| 2 | Ujian Akhir Semester Ganjil | 01-11-2021 s/d 11-11-2021 | 2020/2021 | <button>ACC</button> <button>Tolak</button> |
| 3 | Pendaftaran Prakerin Untuk Kelas 11 | 01-03-2021 s/d 15-03-2021 | 2020/2021 | <button>ACC</button> <button>Tolak</button> |
| 4 | UKK Untuk Kelas 12 | 03-11-2021 s/d 10-11-2021 | 2020/2021 | <button>ACC</button> <button>Tolak</button> |
| 5 | Libur UT斯 | 28-05-2021 s/d 04-06-2021 | 2020/2021 | <button>ACC</button> <button>Tolak</button> |

Gambar 4. 232 Antarmuka Verifikasi Kalender Akademik**46. Perbarui Data Diri Siswa**

Pada Gambar 4. 233 merupakan implementasi antarmuka untuk fungsi perbarui data diri siswa, yang memungkinkan aktor (siswa) dapat melakukan pembaharuan terhadap data diri masing-masing.

The screenshot shows a user profile for 'Muhammad Taftiyan Ilham Akbar'. The fields filled are:

- NISN: 7812
- Nama: Muhammad Taftiyan Ilham Akbar
- Alamat: Mengelo Selatan
- Tempat Lahir: Mojokerto
- Tanggal Lahir: 19/05/2021
- Nomor Telepon: 08995998447
- Agama: Islam
- Anak Ke: 1
- Jumlah Saudara: 2

At the bottom right, it says "Copyright © 2020 || SMK MUTU Kemtagi".

Gambar 4. 233 Antarmuka Perbarui Data Diri Siswa

47. Perbarui Data Diri Guru

Pada Gambar 4. 234 merupakan implementasi antarmuka untuk fungsi perbarui data diri guru, yang memungkinkan aktor (guru) dapat melakukan pembaharuan terhadap data diri masing-masing.

The screenshot shows a user profile for 'Nur Awaliudin S.T.'. The fields filled are:

- NIDN: 87543
- Nama: Nur Awaliudin S.T.
- Alamat: Kementren Mojokerto
- Tempat Lahir: Mojokerto
- Tanggal Lahir: 19/05/1976
- Nomor Telepon: 08995998447
- Agama: Islam
- Pendidikan Terakhir: S1
- Jabatan: Kepala BIOK

At the bottom right, it says "Copyright © 2020 || SMK MUTU Kemtagi".

Gambar 4. 234 Antarmuka Perbarui Data Diri Guru

48. Lihat Kelas Yang Diwakili

Pada Gambar 4. 235 merupakan implementasi antarmuka untuk fungsi lihat daftar kelas yang diwakili, yang memungkinkan aktor (guru) dapat melihat daftar data kelas yang diwakili oleh aktor.

**Gambar 4. 235 Antarmuka Kelas Yang Diwakili****49. Lihat Data Nilai**

Pada Gambar 4. 236 merupakan implementasi antarmuka untuk fungsi lihat daftar nilai yang diperoleh siswa di tiap kelas, yang memungkinkan aktor (guru) dapat melihat daftar data nilai siswa pada mata pelajaran tertentu.

| No | NISN | Name Siswa | Aksi |
|----|------|------------------------------|---|
| 1 | 7812 | Muhammad Taufyan Ilham Akbar | <button>Nilai UTS</button> <button>Nilai UAS</button> |
| 2 | 7826 | Imam Buchori | <button>Nilai UTS</button> <button>Nilai UAS</button> |
| 3 | 7965 | Sababila Putri | <button>Nilai UTS</button> <button>Nilai UAS</button> |

Gambar 4. 236 Antarmuka Data Nilai**50. Lihat Detail Nilai Siswa**

Pada Gambar 4. 237 merupakan implementasi antarmuka untuk fungsi lihat detail hasil nilai siswa, yang memungkinkan aktor (guru) dapat melihat informasi hasil nilai secara lebih detail.

| No | Mata Pelajaran | Nilai Pengetahuan | Deskripsi Nilai Pengetahuan | Nilai Keterampilan | Deskripsi Nilai Pengetahuan | Aksi |
|----|----------------|-------------------|--|--------------------|---|--|
| 1 | Biologi | 87,5 | Peserta didik memiliki pengetahuan yang baik dalam penguasaan mata pelajaran secara teori. | 90 | Peserta didik dapat menerapkan dengan sangat baik pengetahuan yang diperoleh dalam kegiatan belajar mengajar. | <input checked="" type="checkbox"/> <input type="checkbox"/> |
| 2 | Fisika | 80 | Peserta didik memiliki pengetahuan yang baik dalam penguasaan mata pelajaran secara teori. | 85 | Peserta didik dapat menerapkan dengan sangat baik pengetahuan yang diperoleh dalam kegiatan belajar mengajar. | <input checked="" type="checkbox"/> <input type="checkbox"/> |
| 3 | Kimia | 89 | Peserta didik memiliki pengetahuan yang baik dalam penguasaan mata pelajaran secara teori. | 90 | Peserta didik dapat menerapkan dengan sangat baik pengetahuan yang diperoleh dalam kegiatan belajar mengajar. | <input checked="" type="checkbox"/> <input type="checkbox"/> |
| 4 | Matematika | 87,5 | Peserta didik memiliki pengetahuan yang baik dalam penguasaan mata pelajaran secara teori. | 85 | Peserta didik dapat menerapkan dengan sangat baik pengetahuan yang diperoleh dalam kegiatan belajar mengajar. | <input checked="" type="checkbox"/> <input type="checkbox"/> |
| 5 | Sistem Digital | 89 | Peserta didik memiliki pengetahuan yang baik dalam penguasaan mata pelajaran secara teori. | 87,5 | Peserta didik dapat menerapkan dengan sangat baik pengetahuan yang diperoleh dalam kegiatan belajar mengajar. | <input checked="" type="checkbox"/> <input type="checkbox"/> |

Copyright © 2020 || SMK MUTU Kertag.

Gambar 4. 237 Antarmuka Detail Nilai Siswa**51. Tambah Data Nilai Siswa**

Pada Gambar 4. 238 merupakan implementasi antarmuka untuk fungsi tambah data nilai siswa, yang memungkinkan aktor (guru) dapat melakukan penambahan data nilai siswa.

Pilih Siswa

Pilih Mata Pelajaran

Nilai Pengetahuan

Deskripsi Nilai Pengetahuan

Nilai Keterampilan

Deskripsi Nilai Keterampilan

Pilih Jenis Evaluasi

Simpan

Copyright © 2020 || SMK MUTU Kertag.

Gambar 4. 238 Antarmuka Tambah Data Nilai Siswa

SIAKAD SMK
MUTU

Form Update Nilai Siswa

Pilih Siswa
Muhammad Taftiyen Ilhem Akbar

Pilih Mata Pelajaran
MTK-TPM-Matematika

Nilai Pengetahuan
87.5

Deskripsi Nilai Pengetahuan
Peserta didik memiliki pengetahuan yang baik dalam penguasaan mata pelajaran secara teori.

Nilai Ketrampilan
90

Deskripsi Nilai Ketrampilan
Peserta didik memiliki pengetahuan yang baik dalam penguasaan mata pelajaran secara teori.

Pilih Jenis Evaluasi
Ujian Tengah Semester

Simpan

Copyright © 2020 || SMK MUTU Kediri

Gambar 4. 239 Antarmuka Ubah Data Nilai Siswa

53. Lihat Data Rapor

Pada Gambar 4. 240 merupakan implementasi antarmuka untuk fungsi lihat daftar data rapor, yang memungkinkan aktor (siswa) dapat melihat daftar data rapor yang tersimpan.

SIAKAD SMK
MUTU

Daftar Rapor Siswa

Menampilkan Daftar Rapor Siswa, Klik tombol detail untuk info lebih lengkapnya.

| No | NISN | Jenis Evaluasi | Aksi |
|----|------|----------------|-----------------|
| 1 | 7812 | UTS | Lihat Detail |

Copyright © 2020 || SMK MUTU Kediri

Gambar 4. 240 Antarmuka Data Rapor

54. Lihat Detail Rapor

Pada Gambar 4. 241 merupakan implementasi antarmuka untuk fungsi lihat detail rapor, yang memungkinkan aktor (siswa) dapat melihat informasi rapor secara lebih detail.

| No | Mata Pelajaran | Nilai Pengetahuan | Deskripsi Nilai Pengetahuan | Nilai Keterampilan | Deskripsi Nilai Pengetahuan |
|----|----------------|-------------------|--|--------------------|---|
| 1 | Biologi | 87,5 | Peserta didik memiliki pengetahuan yang baik dalam penguasaan mata pelajaran secara teori. | 90 | Peserta didik dapat menerapkan dengan sangat baik pengetahuan yang diperoleh dalam kegiatan belajar mengajar. |
| 2 | Fisika | 80 | Peserta didik memiliki pengetahuan yang baik dalam penguasaan mata pelajaran secara teori. | 85 | Peserta didik dapat menerapkan dengan sangat baik pengetahuan yang diperoleh dalam kegiatan belajar mengajar. |
| 3 | Kimia | 89 | Peserta didik memiliki pengetahuan yang baik dalam penguasaan mata pelajaran secara teori. | 90 | Peserta didik dapat menerapkan dengan sangat baik pengetahuan yang diperoleh dalam kegiatan belajar mengajar. |
| 4 | Matematika | 87,5 | Peserta didik memiliki pengetahuan yang baik dalam penguasaan mata pelajaran secara teori. | 85 | Peserta didik dapat menerapkan dengan sangat baik pengetahuan yang diperoleh dalam kegiatan belajar mengajar. |
| 5 | Sistem Digital | 89 | Peserta didik memiliki pengetahuan yang baik dalam penguasaan mata pelajaran secara teori. | 87,5 | Peserta didik dapat menerapkan dengan sangat baik pengetahuan yang diperoleh dalam kegiatan belajar mengajar. |

Copyright © 2020 || SMK MUTU Kemagi

Gambar 4. 241 Antarmuka Detail Rapor**55. Lihat Data Pengajuan Prakerin**

Pada Gambar 4. 242 merupakan implementasi antarmuka untuk fungsi lihat daftar data pengajuan prakerin, yang memungkinkan aktor (siswa) dapat melihat daftar data pengajuan prakerin yang tersimpan.

| No | NISN | Nama | Tempat Prakerin | Alamat | Deskripsi | Status | Aksi |
|----|------|-------------------------------|------------------------|---|---|-------------------|------|
| 1 | 7812 | Muhammad Taftiyen Ilham Akbar | PT Ajinomoto Indonesia | Jl. Raya Mlirip No.310, Gedong, Mlirip, Kec. Jetis, Mojokerto, Jawa Timur 61395 | Mulai bulan Desember 2020 berakhir pada tanggal 12 Februari 2021. | Menunggu Validasi | |

Perhatian! pengajuan hanya diperkenankan bagi siswa yang sudah mengirimkan laporan ke BKK

Copyright © 2020 || SMK MUTU Kemagi

Gambar 4. 242 Antarmuka Data Pengajuan Prakerin**56. Tambah Pengajuan Prakerin**

Pada Gambar 4. 243 merupakan implementasi antarmuka untuk fungsi tambah data pengajuan prakerin, yang memungkinkan aktor (siswa) dapat melakukan penambahan data pengajuan prakerin.

The screenshot shows a web-based application interface for 'SIAKAD SMK MUTU'. On the left is a sidebar with icons for Dashboard, Data Diri, Kelas Satu Iri, Rapor, Prakerin, Kalendar Akademik, and Logout. The main content area is titled 'Form Pengajuan Prakerin'. It contains several input fields: 'NISN' (with placeholder 'Masukkan NISN anda'), 'Nama Lengkap' (placeholder 'Masukkan nama lengkap anda'), 'Tempat Prakerin' (placeholder 'Masukkan industri tempat anda melaksanakan prakerin'), 'Alamat Prakerin' (placeholder 'Masukkan alamat industri tempat anda melaksanakan prakerin'), and 'Deskripsi Prakerin' (placeholder 'Deskripsikan secara singkat pelaksanaan prakerin anda'). A blue 'Simpan' button is located at the bottom right of the form.

Gambar 4. 243 Antarmuka Pengajuan Penuntasan Prakerin

57. Ubah Data Pengajuan Prakerin

Pada Gambar 4. 244 merupakan implementasi antarmuka untuk fungsi ubah data pengajuan prakerin, yang memungkinkan aktor (siswa) dapat melakukan perubahan terhadap data pengajuan prakerin.

This screenshot shows the 'Ubah Data Pengajuan Prakerin' (Change Prakerin Application Data) page. The sidebar on the left is identical to the previous screenshot. The main form has the following data: 'NISN' field contains '7812'; 'Nama Lengkap' field contains 'Muhammad Taftyan Ilham Akbar'; 'Tempat Prakerin' field contains 'PT Ajinomoto Indonesia'; 'Alamat Prakerin' field contains 'Jl. Raya Mlimp No 110, Gedong, Mlimp, Kec. Jetis, Mojokerto, Jawa Timur 61352'; and 'Deskripsi Prakerin' field contains 'Mulai bulan Desember 2020 berakhir pada tanggal 12 Februari 2021.' A blue 'Perbarui' (Update) button is at the bottom.

Gambar 4. 244 Antarmuka Ubah Data Pengajuan Prakerin

4.4 Pengujian

Pada tahap ini akan dilakukan serangkaian proses atau kegiatan yang berhubungan dengan pengujian terhadap sistem. Pengujian sistem dilakukan untuk memastikan pemenuhan kebutuhan yang telah dispesifikasikan sebelumnya, teknik yang dilakukan pada tahap pengujian adalah pengujian validasi menggunakan *black-box testing* dan pengujian penerimaan pengguna menggunakan *user acceptance test (UAT)*.

4.4.1 Pengujian Validasi

Pengujian validasi dilakukan untuk memastikan pemenuhan kebutuhan sistem secara teknis, yaitu dengan melakukan uji coba terhadap fitur-fitur yang ada pada sistem yang selanjutnya dicocokan hasil keluaran dari sistem apakah sesuai dengan hasil yang diharapkan. Berikut merupakan rincian kasus uji terhadap masing-masing fitur yang terdapat pada sistem informasi.

1. Login Sistem

Pengujian proses *login* sistem secara rinci dijelaskan pada Tabel 4.107, prosedur meliputi nomor kasus uji, nama kasus uji, objek pengujian, prosedur pengujian, hasil yang didapatkan, hasil yang diharapkan.

Tabel 4. 107 Rincian Pengujian Login Sistem

| | |
|-----------------------|--|
| Nomor Kasus Uji | Validasi_01 |
| Nama Kasus Uji | <i>Login Sistem Informasi</i> |
| Objek pengujian | SIAKAD-F-01 |
| Prosedur pengujian | <ol style="list-style-type: none">1) Penguji mengakses halaman sistem informasi melalui peramban web.2) Penguji memasukkan username dan password.3) Penguji mengklik tombol masuk. |
| Hasil yang diharapkan | <ol style="list-style-type: none">1) Sistem akan mengarahkan penguji pada halaman <i>dashboard</i> sesuai dengan level aksesnya apabila data yang dimasukkan valid.2) Sistem akan menampilkan keterangan data yang dimasukkan salah apabila ada kekeliruan data <i>username</i> atau <i>password</i>.3) Sistem akan mengarahkan penguji pada halaman login apabila penguji mencoba mengakses halaman <i>dashboard</i> sebelum login. |
| Hasil yang didapatkan | <ol style="list-style-type: none">1) Sistem berhasil mengarahkan penguji pada halaman <i>dashboard</i> sesuai dengan level aksesnya saat data yang dimasukkan valid.2) Sistem berhasil menampilkan keterangan data yang dimasukkan salah saat ada kekeliruan data <i>username</i> atau <i>password</i>. |

| | |
|--|---|
| | 3) Sistem berhasil mengarahkan penguji pada halaman <i>login</i> saat penguji mencoba mengakses halaman <i>dashboard</i> sebelum <i>login</i> . |
|--|---|

2. Logout Sistem

Pengujian proses *logout* sistem secara rinci dijelaskan pada Tabel 4. 108, prosedur meliputi nomor kasus uji, nama kasus uji, objek pengujian, prosedur pengujian, hasil yang didapatkan, hasil yang diharapkan.

Tabel 4. 108 Rincian Pengujian Logout Sistem

| | |
|-----------------------|---|
| Nomor Kasus Uji | Validasi_02 |
| Nama Kasus Uji | <i>Logout Sistem Informasi</i> |
| Objek pengujian | SIAKAD-F-02 |
| Prosedur pengujian | Penguji mengklik tombol <i>logout</i> yang ada pada bagian <i>sidebar</i> sistem informasi. |
| Hasil yang diharapkan | Sistem akan menghapus informasi <i>login</i> yang tersimpan, dan mengarahkan penguji pada halaman <i>login</i> . |
| Hasil yang didapatkan | Sistem berhasil menghapus informasi <i>login</i> yang tersimpan, lalu mengarahkan penguji pada halaman <i>login</i> . |

3. Melihat Kalender Akademik

Pengujian proses melihat kalender akademik secara rinci dijelaskan pada Tabel 4. 109, prosedur meliputi nomor kasus uji, nama kasus uji, objek pengujian, prosedur pengujian, hasil yang didapatkan, hasil yang diharapkan.

Tabel 4. 109 Rincian Pengujian Lihat Kalender Akademik

| | |
|-----------------------|--|
| Nomor Kasus Uji | Validasi_03 |
| Nama Kasus Uji | Melihat Kalender Akademik |
| Objek pengujian | SIAKAD-F-03 |
| Prosedur pengujian | Penguji mengklik tombol lihat kalender akademik yang ada pada bagian <i>sidebar</i> sistem informasi. |
| Hasil yang diharapkan | Sistem akan mengarahkan penguji pada halaman kalender akademik dan menampilkan data kalender akademik. |

| | |
|-----------------------|---|
| Hasil yang didapatkan | Sistem berhasil mengarahkan pengujji pada halaman kalender akademik dan menampilkan data kalender akademik. |
|-----------------------|---|

4. Mengelola Data Jurusan

Pengujian proses mengelola data jurusan secara rinci dijelaskan pada Tabel 4. 110, prosedur meliputi nomor kasus uji, nama kasus uji, objek pengujian, prosedur pengujian, hasil yang didapatkan, hasil yang diharapkan.

Tabel 4. 110 Rincian Pengujian Mengelola Data Jurusan

| | |
|-----------------------|--|
| Nomor Kasus Uji | Validasi_04 |
| Nama Kasus Uji | Mengelola Data Jurusan |
| Objek pengujian | SIAKAD-F-04 |
| Prosedur pengujian | <ol style="list-style-type: none">1) Melihat Daftar Jurusan<ul style="list-style-type: none">• Pengujji mengklik menu jurusan yang ada pada bagian <i>sidebar</i> sistem informasi.• Pengujji melihat daftar data jurusan yang ada saat ini.2) Menambah Data Jurusan<ul style="list-style-type: none">• Pengujji mengklik tombol tambah data jurusan yang ada diatas daftar jurusan.• Pengujji mengisi data jurusan dan mengklik tombol simpan.3) Mengubah Data Jurusan<ul style="list-style-type: none">• Pengujji mengklik ikon pensil yang ada pada daftar jurusan.• Pengujji mengisi data terbaru dan mengklik tombol simpan.4) Menghapus Data Jurusan<ul style="list-style-type: none">• Pengujji mengklik ikon tong sampah yang ada pada daftar jurusan.• Pengujji mengklik pilihan konfirmasi yang muncul. |
| Hasil yang diharapkan | <ol style="list-style-type: none">1) Melihat Daftar Jurusan |

| | |
|-----------------------|--|
| | <ul style="list-style-type: none"> • Sistem dapat menampilkan data jurusan yang sesuai. <p>2) Menambah Data Jurusan</p> <ul style="list-style-type: none"> • Sistem dapat menampilkan halaman <i>form</i> tambah data jurusan. • Sistem dapat menyimpan data jurusan yang telah dimasukkan oleh pengujji. <p>3) Mengubah Data Jurusan</p> <ul style="list-style-type: none"> • Sistem dapat menampilkan halaman <i>form</i> ubah data jurusan. • Sistem dapat mengubah data jurusan sesuai dengan masukkan dari pengujji. <p>4) Menghapus Data Jurusan</p> <ul style="list-style-type: none"> • Sistem dapat melakukan penghapusan data jurusan yang dipilih. |
| Hasil yang didapatkan | <p>1) Melihat Daftar Jurusan</p> <ul style="list-style-type: none"> • Sistem berhasil menampilkan data jurusan yang sesuai. <p>2) Menambah Data Jurusan</p> <ul style="list-style-type: none"> • Sistem berhasil menampilkan halaman <i>form</i> tambah data jurusan. • Sistem berhasil menyimpan data jurusan yang telah dimasukkan oleh pengujji. <p>3) Mengubah Data Jurusan</p> <ul style="list-style-type: none"> • Sistem berhasil menampilkan halaman <i>form</i> ubah data jurusan. • Sistem berhasil mengubah data jurusan sesuai dengan masukkan dari pengujji. <p>4) Menghapus Data Jurusan</p> <ul style="list-style-type: none"> • Sistem berhasil melakukan penghapusan data jurusan yang dipilih. |

5. Mengelola Data Kelas

Pengujian proses mengelola data kelas secara rinci dijelaskan pada Tabel 4. 111, prosedur meliputi nomor kasus uji, nama kasus uji, objek pengujian, prosedur pengujian, hasil yang didapatkan, hasil yang diharapkan.

Tabel 4. 111 Rincian Pengujian Mengelola Data Kelas

| | |
|-----------------------|--|
| Nomor Kasus Uji | Validasi_05 |
| Nama Kasus Uji | Mengelola Data Kelas |
| Objek pengujian | SIAKAD-F-05 |
| Prosedur pengujian | <ol style="list-style-type: none">1) Melihat Daftar Kelas<ul style="list-style-type: none">• Penguji mengklik menu kelas yang ada pada bagian <i>sidebar</i> sistem informasi.• Penguji melihat daftar data kelas yang ada saat ini.2) Menambah Data Kelas<ul style="list-style-type: none">• Penguji mengklik tombol tambah data kelas yang ada diatas daftar kelas.• Penguji mengisi data kelas dan mengklik tombol simpan.3) Melihat Detail Kelas<ul style="list-style-type: none">• Penguji mengklik ikon info yang ada pada daftar kelas.• Penguji melihat detail informasi kelas.4) Mengubah Data Kelas<ul style="list-style-type: none">• Penguji mengklik ikon pensil yang ada pada daftar kelas.• Penguji mengisi data terbaru dan mengklik tombol simpan.5) Menghapus Data Jurusan<ul style="list-style-type: none">• Penguji mengklik ikon tong sampah yang ada pada daftar kelas.• Penguji mengklik pilihan konfirmasi yang muncul. |
| Hasil yang diharapkan | <ol style="list-style-type: none">1) Melihat Daftar Kelas |

| | |
|-----------------------|---|
| | <ul style="list-style-type: none"> • Sistem dapat menampilkan data kelas yang sesuai. <p>2) Melihat Detail Kelas</p> <ul style="list-style-type: none"> • Sistem dapat menampilkan detail informasi kelas yang sesuai. <p>3) Menambah Data Kelas</p> <ul style="list-style-type: none"> • Sistem dapat menampilkan halaman <i>form</i> tambah data kelas. • Sistem dapat menyimpan data kelas yang telah dimasukkan oleh pengujji. <p>4) Mengubah Data Kelas</p> <ul style="list-style-type: none"> • Sistem dapat menampilkan halaman <i>form</i> ubah data kelas. • Sistem dapat mengubah data kelas sesuai dengan masukkan dari pengujji. <p>5) Menghapus Data Kelas</p> <ul style="list-style-type: none"> • Sistem dapat melakukan penghapusan data kelas yang dipilih. |
| Hasil yang didapatkan | <p>1) Melihat Daftar Kelas</p> <ul style="list-style-type: none"> • Sistem berhasil menampilkan data kelas yang sesuai. <p>2) Melihat Detail Kelas</p> <ul style="list-style-type: none"> • Sistem berhasil menampilkan detail informasi kelas yang sesuai. <p>3) Menambah Data Kelas</p> <ul style="list-style-type: none"> • Sistem berhasil menampilkan halaman <i>form</i> tambah data kelas. • Sistem berhasil menyimpan data kelas yang telah dimasukkan oleh pengujji. <p>4) Mengubah Data Kelas</p> <ul style="list-style-type: none"> • Sistem berhasil menampilkan halaman <i>form</i> ubah data kelas. • Sistem berhasil mengubah data kelas sesuai dengan masukkan dari pengujji. <p>5) Menghapus Data Kelas</p> |

| | | | | |
|--|--|--|--|--|
| | | | | • Sistem berhasil melakukan penghapusan data kelas yang dipilih. |
|--|--|--|--|--|

6. Mengelola Data Mata Pelajaran

Pengujian proses mengelola data mata pelajaran secara rinci dijelaskan pada Tabel 4. 112, prosedur meliputi nomor kasus uji, nama kasus uji, objek pengujian, prosedur pengujian, hasil yang didapatkan, hasil yang diharapkan.

Tabel 4. 112 Rincian Pengujian Mengelola Data Mata Pelajaran

| | |
|--------------------|--|
| Nomor Kasus Uji | Validasi_06 |
| Nama Kasus Uji | Mengelola Data Mata Pelajaran |
| Objek pengujian | SIAKAD-F-06 |
| Prosedur pengujian | <ol style="list-style-type: none">1) Melihat Daftar Mata Pelajaran<ul style="list-style-type: none">• Penguji mengklik menu mata pelajaran yang ada pada bagian sidebar sistem informasi.• Penguji melihat daftar data mata pelajaran yang ada saat ini.2) Melakukan Filter Jadwal Pelajaran<ul style="list-style-type: none">• Penguji memilih jurusan yang ingin dilihat daftar mata pelajaran yang ada.• Penguji mengklik tombol filter.3) Menambah Data Mata Pelajaran<ul style="list-style-type: none">• Penguji mengklik tombol tambah data mata pelajaran yang ada diatas daftar mata pelajaran.• Penguji mengisi data mata pelajaran dan mengklik tombol simpan.4) Mengubah Data Mata Pelajaran<ul style="list-style-type: none">• Penguji mengklik ikon pensil yang ada pada daftar mata pelajaran.• Penguji mengisi data terbaru dan mengklik tombol simpan.5) Menghapus Data Mata Pelajaran<ul style="list-style-type: none">• Penguji mengklik ikon tong sampah yang ada pada daftar mata pelajaran. |

| | |
|-----------------------|--|
| | <ul style="list-style-type: none"> Penguji mengklik pilihan konfirmasi yang muncul. |
| Hasil yang diharapkan | <ol style="list-style-type: none"> 1) Melihat Daftar Mata Pelajaran <ul style="list-style-type: none"> Sistem dapat menampilkan daftar data mata pelajaran yang sesuai. 2) Filter Jadwal Pelajaran <ul style="list-style-type: none"> Sistem dapat menampilkan daftar mata pelajaran sesuai jurusan yang dipilih. 3) Menambah Data Mata Pelajaran <ul style="list-style-type: none"> Sistem dapat menampilkan halaman form tambah data mata pelajaran. Sistem dapat menyimpan data mata pelajaran yang telah dimasukkan oleh penguji. 4) Mengubah Data Mata Pelajaran <ul style="list-style-type: none"> Sistem dapat menampilkan halaman form ubah data mata pelajaran. Sistem dapat mengubah data mata pelajaran sesuai dengan masukan dari penguji. 5) Menghapus Data Mata Pelajaran <ul style="list-style-type: none"> Sistem dapat melakukan penghapusan data mata pelajaran yang dipilih. |
| Hasil yang didapatkan | <ol style="list-style-type: none"> 1) Melihat Daftar Mata Pelajaran <ul style="list-style-type: none"> Sistem berhasil menampilkan data mata pelajaran yang sesuai. 2) Filter Data Mata Pelajaran <ul style="list-style-type: none"> Sistem berhasil menampilkan data mata pelajaran sesuai jurusan yang dipilih. 3) Menambah Data Mata Pelajaran <ul style="list-style-type: none"> Sistem berhasil menampilkan halaman form tambah data mata pelajaran. Sistem berhasil menyimpan data mata pelajaran yang telah dimasukkan oleh penguji. |

| | |
|--|---|
| | <p>4) Mengubah Data Mata Pelajaran</p> <ul style="list-style-type: none"> • Sistem berhasil menampilkan halaman <i>form</i> ubah data mata pelajaran. • Sistem berhasil mengubah data mata pelajaran sesuai dengan masukkan dari penguji. <p>5) Menghapus Data Mata Pelajaran</p> <ul style="list-style-type: none"> • Sistem berhasil melakukan penghapusan data mata pelajaran yang dipilih. |
|--|---|

7. Mengelola Data Guru

Pengujian proses mengelola data guru secara rinci dijelaskan pada Tabel 4. 113, prosedur meliputi nomor kasus uji, nama kasus uji, objek pengujian, prosedur pengujian, hasil yang didapatkan, hasil yang diharapkan.

Tabel 4. 113 Rincian Pengujian Mengelola Data Guru

| | |
|--------------------|--|
| Nomor Kasus Uji | Validasi_07 |
| Nama Kasus Uji | Mengelola Data Guru |
| Objek pengujian | SIAKAD-F-07 |
| Prosedur pengujian | <p>1) Melihat Daftar Guru</p> <ul style="list-style-type: none"> • Penguji mengklik menu guru yang ada pada bagian <i>sidebar</i> sistem informasi. • Penguji melihat daftar data guru yang ada saat ini. <p>2) Menambah Data Guru</p> <ul style="list-style-type: none"> • Penguji mengklik tombol tambah data guru yang ada diatas daftar guru. • Penguji mengisi data guru dan mengklik tombol simpan. <p>3) Melihat Detail Guru</p> <ul style="list-style-type: none"> • Penguji mengklik ikon info yang ada pada daftar guru. • Penguji melihat detail informasi guru. <p>4) Mengubah Data Guru</p> |

| | |
|-----------------------|---|
| | <ul style="list-style-type: none"> Penguji mengklik ikon pensil yang ada pada daftar guru. Penguji mengisi data terbaru dan mengklik tombol simpan. <p>5) Menghapus Data Guru</p> <ul style="list-style-type: none"> Penguji mengklik ikon tong sampah yang ada pada daftar guru. Penguji mengklik pilihan konfirmasi yang muncul. |
| Hasil yang diharapkan | <p>1) Melihat Daftar Guru</p> <ul style="list-style-type: none"> Sistem dapat menampilkan daftar data guru yang sesuai. <p>2) Melihat Detail Guru</p> <ul style="list-style-type: none"> Sistem dapat menampilkan detail informasi guru yang sesuai. <p>3) Menambah Data Guru</p> <ul style="list-style-type: none"> Sistem dapat menampilkan halaman form tambah data guru. Sistem dapat menyimpan data guru yang telah dimasukkan oleh penguji. <p>4) Mengubah Data Guru</p> <ul style="list-style-type: none"> Sistem dapat menampilkan halaman form ubah data guru. Sistem dapat mengubah data guru sesuai dengan masukkan dari penguji. <p>5) Menghapus Data Guru</p> <ul style="list-style-type: none"> Sistem dapat melakukan penghapusan data guru yang dipilih. |
| Hasil yang didapatkan | <p>1) Melihat Daftar Guru</p> <ul style="list-style-type: none"> Sistem berhasil menampilkan daftar data guru yang sesuai. <p>2) Melihat Detail Guru</p> <ul style="list-style-type: none"> Sistem berhasil menampilkan detail informasi guru yang sesuai. <p>3) Menambah Data Guru</p> |

| | |
|--|--|
| | <ul style="list-style-type: none"> Sistem berhasil menampilkan halaman <i>form</i> tambah data guru. Sistem berhasil menyimpan data guru yang telah dimasukkan oleh pengujii. <p>4) Mengubah Data Guru</p> <ul style="list-style-type: none"> Sistem berhasil menampilkan halaman <i>form</i> ubah data guru. Sistem berhasil mengubah data guru sesuai dengan masukkan dari pengujii. <p>5) Menghapus Data Guru</p> <ul style="list-style-type: none"> Sistem berhasil melakukan penghapusan data guru yang dipilih. |
|--|--|

8. Mengelola Data Siswa

Pengujian proses mengelola data siswa secara rinci dijelaskan pada Tabel 4. 114, prosedur meliputi nomor kasus uji, nama kasus uji, objek pengujian, prosedur pengujian, hasil yang didapatkan, hasil yang diharapkan.

Tabel 4. 114 Rincian Pengujian Mengelola Data Siswa

| | |
|--------------------|---|
| Nomor Kasus Uji | Validasi_08 |
| Nama Kasus Uji | Mengelola Data Siswa |
| Objek pengujian | SIAKAD-F-08 |
| Prosedur pengujian | <p>1) Melihat Daftar Siswa</p> <ul style="list-style-type: none"> Penguji mengklik menu siswa yang ada pada bagian <i>sidebar</i> sistem informasi. Penguji melihat daftar data siswa yang ada saat ini. <p>2) Mencari Data Siswa</p> <ul style="list-style-type: none"> Penguji mengisi data NISN pada kolom yang disediakan. Penguji mengklik tombol cari. <p>3) Melihat Detail Siswa</p> <ul style="list-style-type: none"> Penguji mengklik ikon info yang ada pada daftar siswa. Penguji melihat detail informasi siswa. |

| | |
|-----------------------|--|
| | <p>4) Menambah Data Siswa</p> <ul style="list-style-type: none">• Penguji mengklik tombol tambah data siswa yang ada diatas daftar siswa.• Penguji mengisi data siswa dan mengklik tombol simpan. <p>5) Mengubah Data Siswa</p> <ul style="list-style-type: none">• Penguji mengklik ikon pensil yang ada pada daftar siswa.• Penguji mengisi data terbaru dan mengklik tombol simpan. <p>6) Menghapus Data Siswa</p> <ul style="list-style-type: none">• Penguji mengklik ikon tong sampah yang ada pada daftar siswa.• Penguji mengklik pilihan konfirmasi yang muncul. |
| Hasil yang diharapkan | <p>1) Melihat Daftar Siswa</p> <ul style="list-style-type: none">• Sistem dapat menampilkan daftar data siswa yang sesuai. <p>2) Mencari Data Siswa</p> <ul style="list-style-type: none">• Sistem dapat menampilkan data siswa sesuai dengan NISN yang dimasukkan. <p>3) Melihat Detail Siswa</p> <ul style="list-style-type: none">• Sistem dapat menampilkan detail informasi siswa yang sesuai. <p>4) Menambah Data Siswa</p> <ul style="list-style-type: none">• Sistem dapat menampilkan halaman form tambah data siswa.• Sistem dapat menyimpan data siswa yang telah dimasukkan oleh penguji. <p>5) Mengubah Data Siswa</p> <ul style="list-style-type: none">• Sistem dapat menampilkan halaman form ubah data siswa.• Sistem dapat mengubah data siswa sesuai dengan masukkan dari penguji. <p>6) Menghapus Data Siswa</p> |

| | |
|-----------------------|--|
| | <ul style="list-style-type: none"> • Sistem dapat melakukan penghapusan data siswa yang dipilih. |
| Hasil yang didapatkan | <ol style="list-style-type: none"> 1) Melihat Daftar Siswa <ul style="list-style-type: none"> • Sistem berhasil menampilkan daftar data siswa yang sesuai. 2) Mencari Data Siswa <ul style="list-style-type: none"> • Sistem berhasil menampilkan data siswa sesuai dengan NISN yang dimasukkan. 3) Melihat Detail Siswa <ul style="list-style-type: none"> • Sistem berhasil menampilkan detail informasi siswa yang sesuai. 4) Menambah Data Siswa <ul style="list-style-type: none"> • Sistem berhasil menampilkan halaman form tambah data siswa. • Sistem berhasil menyimpan data siswa yang telah dimasukkan oleh pengujji. 5) Mengubah Data Siswa <ul style="list-style-type: none"> • Sistem berhasil menampilkan halaman form ubah data siswa. • Sistem berhasil mengubah data siswa sesuai dengan masukan dari pengujji. 6) Menghapus Data Siswa <ul style="list-style-type: none"> • Sistem berhasil melakukan penghapusan data siswa yang dipilih. |

9. Mengelola Data Jadwal Pelajaran

Pengujian proses mengelola data jadwal pelajaran secara rinci dijelaskan pada Tabel 4. 115, prosedur meliputi nomor kasus uji, nama kasus uji, objek pengujian, prosedur pengujian, hasil yang didapatkan, hasil yang diharapkan.

Tabel 4. 115 Rincian Pengujian Mengelola Data Jadwal Pelajaran

| | |
|-----------------|---------------------------------|
| Nomor Kasus Uji | Validasi_09 |
| Nama Kasus Uji | Mengelola Data Jadwal Pelajaran |
| Objek pengujian | SIAKAD-F-09 |

| | |
|------------------------------|--|
| Prosedur pengujian | <ol style="list-style-type: none">1) Melihat Daftar Jadwal Pelajaran<ul style="list-style-type: none">• Penguji mengklik menu jadwal pelajaran yang ada pada bagian sidebar sistem informasi.2) Menambah Data Jadwal Pelajaran<ul style="list-style-type: none">• Penguji mengklik tombol tambah data jadwal pelajaran yang ada diatas daftar jadwal pelajaran.• Penguji mengisi data jadwal pelajaran dan mengklik tombol simpan.3) Mengubah Data Jadwal Pelajaran<ul style="list-style-type: none">• Penguji mengklik ikon pensil yang ada pada daftar jadwal pelajaran.• Penguji mengisi data terbaru dan mengklik tombol simpan.4) Menghapus Data Jadwal Pelajaran<ul style="list-style-type: none">• Penguji mengklik ikon tong sampah yang ada pada daftar jadwal pelajaran.• Penguji mengklik pilihan konfirmasi yang muncul. |
| Hasil yang diharapkan | <ol style="list-style-type: none">1) Melihat Daftar Jadwal Pelajaran<ul style="list-style-type: none">• Sistem dapat menampilkan data jadwal pelajaran yang sesuai.2) Menambah Data Jadwal Pelajaran<ul style="list-style-type: none">• Sistem dapat menampilkan halaman form tambah data jadwal pelajaran.• Sistem dapat menyimpan data jadwal pelajaran yang telah dimasukkan oleh penguji.3) Mengubah Data Jadwal Pelajaran<ul style="list-style-type: none">• Sistem dapat menampilkan halaman form ubah data jadwal pelajaran.• Sistem dapat mengubah data jadwal pelajaran sesuai dengan masukan dari penguji. |

| | |
|-----------------------|--|
| | <p>4) Menghapus Data Jadwal Pelajaran</p> <ul style="list-style-type: none"> • Sistem dapat melakukan penghapusan data jadwal pelajaran yang dipilih. |
| Hasil yang didapatkan | <p>1) Melihat Daftar Jadwal Pelajaran</p> <ul style="list-style-type: none"> • Sistem berhasil menampilkan data jadwal pelajaran yang sesuai. <p>2) Menambah Data Jadwal Pelajaran</p> <ul style="list-style-type: none"> • Sistem berhasil menampilkan halaman <i>form</i> tambah data jadwal pelajaran. <p>3) Mengubah Data Jadwal Pelajaran</p> <ul style="list-style-type: none"> • Sistem berhasil menampilkan halaman <i>form</i> ubah data jadwal pelajaran. <p>4) Menghapus Data Jadwal Pelajaran</p> <ul style="list-style-type: none"> • Sistem berhasil melakukan penghapusan data jadwal pelajaran yang dipilih. |

10. Mengelola Data Kegiatan Akademik

Pengujian proses mengelola data kegiatan akademik secara rinci dijelaskan pada Tabel 4. 116, prosedur meliputi nomor kasus uji, nama kasus uji, objek pengujian, prosedur pengujian, hasil yang didapatkan, hasil yang diharapkan.

Tabel 4. 116 Rincian Pengujian Mengelola Data Kegiatan Akademik

| | |
|--------------------|---|
| Nomor Kasus Uji | Validasi_10 |
| Nama Kasus Uji | Mengelola Data Kegiatan Akademik |
| Objek pengujian | SIAKAD-F-10 |
| Prosedur pengujian | <p>1) Melihat Daftar Kegiatan Akademik</p> <ul style="list-style-type: none"> • Penguji mengklik menu kegiatan akademik yang ada pada bagian sidebar sistem informasi. |

| | |
|-----------------------|--|
| | <ul style="list-style-type: none"> Penguji melihat daftar data kegiatan akademik yang ada saat ini. <p>2) Menambah Data Kegiatan Akademik</p> <ul style="list-style-type: none"> Penguji mengklik tombol tambah data kegiatan akademik yang ada diatas daftar kegiatan akademik. Penguji mengisi data kegiatan akademik dan mengklik tombol simpan. <p>3) Mengubah Data Kegiatan Akademik</p> <ul style="list-style-type: none"> Penguji mengklik ikon pensil yang ada pada daftar kegiatan akademik. Penguji mengisi data terbaru dan mengklik tombol simpan. <p>4) Menghapus Data Kegiatan Akademik</p> <ul style="list-style-type: none"> Penguji mengklik ikon tong sampah yang ada pada daftar kegiatan akademik. Penguji mengklik pilihan konfirmasi yang muncul. |
| Hasil yang diharapkan | <p>1) Melihat Daftar Kegiatan Akademik</p> <ul style="list-style-type: none"> Sistem dapat menampilkan data kegiatan akademik yang sesuai. <p>2) Menambah Data Kegiatan Akademik</p> <ul style="list-style-type: none"> Sistem dapat menampilkan halaman form tambah data kegiatan akademik. Sistem dapat menyimpan data kegiatan akademik yang telah dimasukkan oleh penguji. <p>3) Mengubah Data Kegiatan Akademik</p> <ul style="list-style-type: none"> Sistem dapat menampilkan halaman form ubah data kegiatan akademik. Sistem dapat mengubah data kegiatan akademik sesuai dengan masukkan dari penguji. <p>4) Menghapus Data Kegiatan Akademik</p> |

| | |
|-----------------------|--|
| | <ul style="list-style-type: none"> Sistem dapat melakukan penghapusan data kegiatan akademik yang dipilih. |
| Hasil yang didapatkan | <ol style="list-style-type: none"> 1) Melihat Daftar Kegiatan Akademik <ul style="list-style-type: none"> Sistem berhasil menampilkan data kegiatan akademik yang sesuai. 2) Menambah Data Kegiatan Akademik <ul style="list-style-type: none"> Sistem berhasil menampilkan halaman <i>form</i> tambah data kegiatan akademik. Sistem berhasil menyimpan data kegiatan akademik yang telah dimasukkan oleh pengujji. 3) Mengubah Data Kegiatan Akademik <ul style="list-style-type: none"> Sistem berhasil menampilkan halaman <i>form</i> ubah data kegiatan akademik. Sistem berhasil mengubah data kegiatan akademik sesuai dengan masukan dari pengujji. 4) Menghapus Data Jurusan <ul style="list-style-type: none"> Sistem berhasil melakukan penghapusan data kegiatan akademik yang dipilih. |

11. Melakukan Validasi Pelaksanaan Prakerin

Pengujian proses melakukan validasi pelaksanaan prakerin secara rinci dijelaskan pada Tabel 4. 117, prosedur meliputi nomor kasus uji, nama kasus uji, objek pengujian, prosedur pengujian, hasil yang didapatkan, hasil yang diharapkan.

Tabel 4. 117 Rincian Pengujian Validasi Pelaksanaan Prakerin

| | |
|--------------------|---|
| Nomor Kasus Uji | Validasi_11 |
| Nama Kasus Uji | Melakukan Validasi Pelaksanaan Prakerin |
| Objek pengujian | SIAKAD-F-11 |
| Prosedur pengujian | <ol style="list-style-type: none"> 1) Pengujji mengklik tombol verifikasi prakerin yang ada pada bagian <i>sidebar</i> sistem informasi. 2) Pengujji memeriksa data yang ada dan memilih tombol ACC atau Tolak. |

| | |
|-----------------------|---|
| Hasil yang diharapkan | Sistem akan mengubah status pengajuan sesuai dengan tombol yang dipilih oleh penguji. |
| Hasil yang didapatkan | Sistem berhasil mengubah status pengajuan sesuai dengan tombol yang dipilih oleh penguji. |

12. Mengelola Data Mitra Industri

Pengujian proses mengelola mitra industri secara rinci dijelaskan pada Tabel 4. 118, prosedur meliputi nomor kasus uji, nama kasus uji, objek pengujian, prosedur pengujian, hasil yang didapatkan, hasil yang diharapkan.

Tabel 4. 118 Rincian Pengujian Mengelola Data Mitra Industri

| | |
|--------------------|---|
| Nomor Kasus Uji | Validasi_12 |
| Nama Kasus Uji | Mengelola Data Mitra Industri |
| Objek pengujian | SIAKAD-F-12 |
| Prosedur pengujian | <p>1) Melihat Daftar Mitra Industri</p> <ul style="list-style-type: none"> Penguji mengklik menu mitra industri yang ada pada bagian <i>sidebar</i> sistem informasi. Penguji melihat daftar data mitra industri yang ada saat ini. <p>2) Menambah Data Mitra Industri</p> <ul style="list-style-type: none"> Penguji mengklik tombol tambah data mitra industri yang ada diatas daftar mitra industri. Penguji mengisi data mitra industri dan mengklik tombol simpan. <p>3) Mengubah Data Mitra Industri</p> <ul style="list-style-type: none"> Penguji mengklik ikon pensil yang ada pada daftar mitra industri. Penguji mengisi data terbaru dan mengklik tombol simpan. <p>4) Menghapus Data Mitra Industri</p> <ul style="list-style-type: none"> Penguji mengklik ikon tong sampah yang ada pada daftar mitra industri. |

| | | |
|-----------------------|--|--|
| | | <ul style="list-style-type: none">• Pengujii mengklik pilihan konfirmasi yang muncul. |
| Hasil yang diharapkan | | <ol style="list-style-type: none">1) Melihat Daftar Mitra Industri<ul style="list-style-type: none">• Sistem dapat menampilkan data mitra industri yang sesuai.2) Menambah Data Mitra Industri<ul style="list-style-type: none">• Sistem dapat menampilkan halaman <i>form</i> tambah data mitra industri.• Sistem dapat menyimpan data mitra industri yang telah dimasukkan oleh pengujii.3) Mengubah Data Mitra Industri<ul style="list-style-type: none">• Sistem dapat menampilkan halaman <i>form</i> ubah data mitra industri.• Sistem dapat mengubah data mitra industri sesuai dengan masukkan dari pengujii.4) Menghapus Data Mitra Industri<ul style="list-style-type: none">• Sistem dapat melakukan penghapusan data mitra industri yang dipilih. |
| Hasil yang didapatkan | | <ol style="list-style-type: none">1) Melihat Daftar Mitra Industri<ul style="list-style-type: none">• Sistem berhasil menampilkan data mitra industri yang sesuai.2) Menambah Data Mitra Industri<ul style="list-style-type: none">• Sistem berhasil menampilkan halaman <i>form</i> tambah data mitra industri.• Sistem berhasil menyimpan data mitra industri yang telah dimasukkan oleh pengujii.3) Mengubah Data Mitra Industri<ul style="list-style-type: none">• Sistem berhasil menampilkan halaman <i>form</i> ubah data mitra industri.• Sistem berhasil mengubah data mitra industri sesuai dengan masukkan dari pengujii.4) Menghapus Data Mitra Industri |

| | | |
|--|--|---|
| | | <ul style="list-style-type: none"> Sistem berhasil melakukan penghapusan data mitra industri yang dipilih. |
|--|--|---|

13. Melihat Pelaksanaan Prakerin

Pengujian proses melihat pelaksanaan prakerin secara rinci dijelaskan pada Tabel 4. 119, prosedur meliputi nomor kasus uji, nama kasus uji, objek pengujian, prosedur pengujian, hasil yang didapatkan, hasil yang diharapkan.

Tabel 4. 119 Rincian Pengujian Melihat Pelaksanaan Prakerin

| | |
|-----------------------|---|
| Nomor Kasus Uji | Validasi_13 |
| Nama Kasus Uji | Melihat Pelaksanaan Prakerin |
| Objek pengujian | SIAKAD-F-13 |
| Prosedur pengujian | <ol style="list-style-type: none"> Penguji mengklik tombol pelaksanaan prakerin yang ada pada bagian sidebar sistem informasi. Penguji melihat daftar pelaksanaan prakerin yang telah diverifikasi. |
| Hasil yang diharapkan | Sistem dapat menampilkan daftar pengajuan prakerin siswa yang telah diverifikasi. |
| Hasil yang didapatkan | Sistem berhasil menampilkan daftar pengajuan prakerin siswa yang telah diverifikasi. |

14. Melakukan Validasi Kalender Akademik

Pengujian proses melakukan validasi kalender akademik secara rinci dijelaskan pada Tabel 4. 120, prosedur meliputi nomor kasus uji, nama kasus uji, objek pengujian, prosedur pengujian, hasil yang didapatkan, hasil yang diharapkan.

Tabel 4. 120 Rincian Pengujian Validasi Kalender Akademik

| | |
|--------------------|---|
| Nomor Kasus Uji | Validasi_14 |
| Nama Kasus Uji | Melakukan Validasi Kalender Akademik |
| Objek pengujian | SIAKAD-F-14 |
| Prosedur pengujian | <ol style="list-style-type: none"> Penguji mengklik tombol verifikasi kalender akademik yang ada pada bagian sidebar sistem informasi. |

| | |
|-----------------------|---|
| | 2) Penguji memeriksa data yang ada dan memilih tombol ACC atau Tolak. |
| Hasil yang diharapkan | Sistem dapat mengubah status pengajuan sesuai dengan tombol yang dipilih oleh penguji. |
| Hasil yang didapatkan | Sistem berhasil mengubah status pengajuan sesuai dengan tombol yang dipilih oleh penguji. |

15. Melakukan Validasi Penetapan Wali Kelas

Pengujian proses melakukan validasi penetapan wali kelas secara rinci dijelaskan pada Tabel 4. 121, prosedur meliputi nomor kasus uji, nama kasus uji, objek pengujian, prosedur pengujian, hasil yang didapatkan, hasil yang diharapkan.

Tabel 4. 121 Rincian Pengujian Validasi Penetapan Kelas

| | |
|-----------------------|---|
| Nomor Kasus Uji | Validasi_15 |
| Nama Kasus Uji | Melakukan Validasi Penetapan Wali Kelas |
| Objek pengujian | SIAKAD-F-15 |
| Prosedur pengujian | <ol style="list-style-type: none"> 1) Penguji mengklik tombol verifikasi wali kelas yang ada pada bagian <i>sidebar</i> sistem informasi. 2) Penguji memeriksa data yang ada dan memilih tombol ACC atau Tolak. |
| Hasil yang diharapkan | Sistem akan mengubah status pengajuan sesuai dengan tombol yang dipilih oleh penguji. |
| Hasil yang didapatkan | Sistem berhasil mengubah status pengajuan sesuai dengan tombol yang dipilih oleh penguji. |

16. Memperbarui Data Diri

Pengujian proses memperbarui data diri secara rinci dijelaskan pada Tabel 4. 122, prosedur meliputi nomor kasus uji, nama kasus uji, objek pengujian, prosedur pengujian, hasil yang didapatkan, hasil yang diharapkan.

Tabel 4. 122 Rincian Pengujian Memperbarui Data Diri

| | |
|-----------------------|--|
| Nomor Kasus Uji | Validasi_16 |
| Nama Kasus Uji | Memperbarui Data Diri |
| Objek pengujian | SIAKAD-F-16 |
| Prosedur pengujian | 1) Penguji mengklik tombol data diri yang ada pada bagian sidebar sistem informasi. 2) Penguji mengisi data terbaru pada field yang tersedia. 3) Penguji mengklik tombol perbarui. |
| Hasil yang diharapkan | Sistem dapat mengubah status pengajuan sesuai dengan data yang dimasukkan oleh penguji. |
| Hasil yang didapatkan | Sistem berhasil mengubah status pengajuan sesuai dengan data yang dimasukkan oleh penguji. |

17. Melihat Kelas Yang Diwakili

Pengujian proses melihat kelas yang diwakili secara rinci dijelaskan pada Tabel 4. 123, prosedur meliputi nomor kasus uji, nama kasus uji, objek pengujian, prosedur pengujian, hasil yang didapatkan, hasil yang diharapkan.

Tabel 4. 123 Rincian Pengujian Melihat Kelas Yang Diwakili

| | |
|-----------------------|---|
| Nomor Kasus Uji | Validasi_17 |
| Nama Kasus Uji | Melihat Kelas Yang Diwakili |
| Objek pengujian | SIAKAD-F-17 |
| Prosedur pengujian | Penguji mengklik menu pelaksanaan prakerin yang ada pada bagian sidebar sistem informasi. |
| Hasil yang diharapkan | Sistem dapat menampilkan detail kelas yang diwakili. |
| Hasil yang didapatkan | Sistem berhasil menampilkan detail kelas yang diwakili. |

| | |
|---|---|
| 18. Mengelola Nilai Rapor Siswa | Pengujian proses mengelola nilai rapor siswa secara rinci dijelaskan pada Tabel 4. 124, prosedur meliputi nomor kasus uji, nama kasus uji, objek pengujian, prosedur pengujian, hasil yang didapatkan, hasil yang diharapkan. |
| Tabel 4. 124 Rincian Pengujian Mengelola Nilai Rapor Siswa | |
| Nomor Kasus Uji | Validasi_18 |
| Nama Kasus Uji | Mengelola Nilai Rapor Siswa |
| Objek pengujian | SIAKAD-F-18 |
| Prosedur pengujian | <p>1) Melihat Daftar Nilai</p> <ul style="list-style-type: none"> Penguji mengklik menu nilai siswa yang ada pada bagian sidebar sistem informasi. Penguji melihat daftar data nilai yang ada saat ini. <p>2) Menambah Data Nilai</p> <ul style="list-style-type: none"> Penguji mengklik tombol tambah data nilai yang ada diatas daftar nilai. Penguji mengisi data nilai dan mengklik tombol simpan. <p>3) Mengubah Data Nilai</p> <ul style="list-style-type: none"> Penguji mengklik ikon pensil yang ada pada daftar nilai. Penguji mengisi data terbaru dan mengklik tombol simpan. <p>4) Menghapus Data Nilai</p> <ul style="list-style-type: none"> Penguji mengklik ikon tong sampah yang ada pada daftar nilai. Penguji mengklik pilihan konfirmasi yang muncul. |
| Hasil yang diharapkan | <p>1) Melihat Daftar Nilai</p> <ul style="list-style-type: none"> Sistem dapat menampilkan data nilai yang sesuai. <p>2) Menambah Data Nilai</p> <ul style="list-style-type: none"> Sistem dapat menampilkan halaman form tambah data nilai. |

| | |
|-----------------------|--|
| | <ul style="list-style-type: none"> Sistem dapat menyimpan data nilai yang telah dimasukkan oleh pengujii. <p>3) Mengubah Data Nilai</p> <ul style="list-style-type: none"> Sistem dapat menampilkan halaman <i>form</i> ubah data nilai. Sistem dapat mengubah data nilai sesuai dengan masukkan dari nilai. <p>4) Menghapus Data Nilai</p> <ul style="list-style-type: none"> Sistem dapat melakukan penghapusan data nilai yang dipilih. |
| Hasil yang didapatkan | <p>1) Melihat Daftar Nilai</p> <ul style="list-style-type: none"> Sistem berhasil menampilkan data nilai yang sesuai. <p>2) Menambah Data Nilai</p> <ul style="list-style-type: none"> Sistem berhasil menampilkan halaman <i>form</i> tambah data nilai. Sistem berhasil menyimpan data nilai yang telah dimasukkan oleh pengujii. <p>3) Mengubah Data Nilai</p> <ul style="list-style-type: none"> Sistem berhasil menampilkan halaman <i>form</i> ubah data nilai. Sistem berhasil mengubah data nilai sesuai dengan masukkan dari pengujii. <p>4) Menghapus Data Nilai</p> <ul style="list-style-type: none"> Sistem berhasil melakukan penghapusan data nilai yang dipilih. |

19. Melihat Rapor

Pengujian proses melihat detail kelas secara rinci dijelaskan pada Tabel 4. 125, prosedur meliputi nomor kasus uji, nama kasus uji, objek pengujian, prosedur pengujian, hasil yang didapatkan, hasil yang diharapkan.

Tabel 4. 125 Rincian Pengujian Melihat Rapor

| | |
|-----------------|---------------|
| Nomor Kasus Uji | Validasi_19 |
| Nama Kasus Uji | Melihat Rapor |

| | |
|-----------------------|--|
| Objek pengujian | SIAKAD-F-19 |
| Prosedur pengujian | <p>1) Penguji mengklik menu rapor yang ada pada bagian <i>sidebar</i> sistem informasi.</p> <p>2) Penguji memilih daftar rapor yang ingin dilihat detailnya.</p> |
| Hasil yang diharapkan | <p>1) Sistem dapat menampilkan daftar rapor yang tersimpan.</p> <p>2) Sistem dapat menampilkan detail rapor yang dipilih.</p> |
| Hasil yang didapatkan | <p>1) Sistem dapat menampilkan daftar rapor yang tersimpan.</p> <p>2) Sistem dapat menampilkan detail rapor yang dipilih.</p> |

20. Mengelola Pengajuan Prakerin

Pengujian proses mengelola pengajuan prakerin secara rinci dijelaskan pada Tabel 4. 126, prosedur meliputi nomor kasus uji, nama kasus uji, objek pengujian, prosedur pengujian, hasil yang didapatkan, hasil yang diharapkan.

Tabel 4. 126 Rincian Pengujian Mengelola Pengajuan Prakerin

| | |
|--------------------|--|
| Nomor Kasus Uji | Validasi_20 |
| Nama Kasus Uji | Mengelola Pengajuan Prakerin |
| Objek pengujian | SIAKAD-F-20 |
| Prosedur pengujian | <p>1) Melihat Daftar Pengajuan Prakerin</p> <ul style="list-style-type: none"> • Penguji mengklik menu prakerin yang ada pada bagian <i>sidebar</i> sistem informasi. • Penguji melihat daftar data pengajuan prakerin yang ada saat ini. <p>2) Mengajukan Penuntasan Pelaksanaan Prakerin</p> <ul style="list-style-type: none"> • Penguji mengklik tombol buat pengajuan yang ada diatas daftar pengajuan prakerin. • Penguji mengisi data pengajuan dan mengklik tombol simpan. |

| | |
|-----------------------|---|
| | <p>3) Mengubah Data Pengajuan Penuntasan Prakerin</p> <ul style="list-style-type: none">• Penguji mengklik ikon pensil yang ada pada daftar pengajuan.• Penguji mengisi data terbaru dan mengklik tombol simpan. |
| Hasil yang diharapkan | <p>1) Melihat Daftar Pengajuan Prakerin</p> <ul style="list-style-type: none">• Sistem dapat menampilkan daftar data pengajuan yang sesuai. <p>2) Mengajukan Penuntasan Prakerin</p> <ul style="list-style-type: none">• Sistem dapat menampilkan halaman form pengajuan penuntasan prakerin.• Sistem dapat menyimpan data pengajuan yang telah dimasukkan oleh penguji. <p>3) Mengubah Data Pengajuan Penuntasan Prakerin</p> <ul style="list-style-type: none">• Sistem dapat menampilkan halaman form ubah data pengajuan.• Sistem dapat mengubah data pengajuan sesuai dengan masukan dari penguji. |
| Hasil yang didapatkan | <p>1) Melihat Daftar Pengajuan Prakerin</p> <ul style="list-style-type: none">• Sistem berhasil menampilkan daftar data pengajuan yang sesuai. <p>2) Mengajukan Penuntasan Prakerin</p> <ul style="list-style-type: none">• Sistem berhasil menampilkan halaman form pengajuan penuntasan prakerin.• Sistem berhasil menyimpan data pengajuan yang telah dimasukkan oleh penguji. <p>3) Mengubah Data Pengajuan Penuntasan Prakerin</p> <ul style="list-style-type: none">• Sistem berhasil menampilkan halaman form ubah data pengajuan. |

| | | | | | | | |
|--|--|--|--|--|--|--|--|
| | | | | | | | <ul style="list-style-type: none"> Sistem berhasil mengubah data pengajuan sesuai dengan masukkan dari pengujian. |
|--|--|--|--|--|--|--|--|

Hasil pengujian validasi pada masing-masing kasus uji diatas menyatakan bahwa sistem informasi telah berjalan sebagaimana mestinya dan memberikan keluaran yang diharapkan, hasil pengujian validasi diperoleh sebesar 100% sehingga sistem informasi akademik SMK Muhammadiyah 1 Kemlagi dinyatakan valid untuk disebarluaskan pada pengguna.

4.4.2 Pengujian Penerimaan Pengguna

Pelaksanaan pengujian penerimaan pengguna dilakukan dalam jangka waktu yang berbeda-beda, dikarenakan kebijakan sekolah dari rumah pada saat penelitian dilakukan. Pengujian dilakukan melalui pengamatan secara langsung, serta pemberian kuesioner setelah pengguna mencoba fitur-fitur yang ada pada aplikasi. Jumlah responden terdiri dari 10 orang yang terdiri dari 1 (satu) orang admin sistem, 1 (satu) orang admin bursa kerja khusu (BKK), 1 (satu) orang waka kurikulum, 2 (dua) orang guru dan 5 (lima) orang siswa. Detail hasil pengujian penerimaan pengguna dapat dilihat pada Lampiran C.

4.4.2.1 Hasil Pengujian Penerimaan Pengguna

Hasil pengujian terhadap sepuluh orang responden diperoleh ringkasan sebagaimana tertera pada Tabel 4. 127.

Tabel 4. 127 Hasil Uji Penerimaan Pengguna

| No | Identitas Responden | | SKOR | | | | Total Skor | Skor Maks. | Penerimaan Pengguna (%) |
|----|---------------------|----------------|------|-------|-----|----|------------|------------|-------------------------|
| | Nama | Jabatan | PU | PEO U | ATU | BI | | | |
| 1 | R1 | Waka Kurikulum | 16 | 12 | 12 | 16 | 56 | 70 | 80,0 |
| 2 | R2 | Kepala BKK | 17 | 13 | 13 | 20 | 63 | 70 | 90,0 |
| 3 | R3 | Tata Usaha | 16 | 12 | 12 | 15 | 59 | 70 | 84,2 |
| 4 | R4 | Guru | 16 | 13 | 14 | 16 | 59 | 70 | 84,2 |
| 5 | R5 | Guru | 19 | 15 | 14 | 16 | 64 | 70 | 91,4 |
| 6 | R6 | Siswa | 15 | 12 | 11 | 16 | 54 | 70 | 77,1 |

| No | Identitas Responden | | SKOR | | | | Total Skor | Skor Maks. | Penerimaan Pengguna (%) |
|---|---------------------|---------|------|-------|------|------|---|------------|-------------------------|
| | Nama | Jabatan | PU | PEO U | ATU | BI | | | |
| 7 | R7 | Siswa | 16 | 11 | 10 | 15 | 52 | 70 | 74,2 |
| 8 | R8 | Siswa | 16 | 11 | 10 | 16 | 53 | 70 | 75,7 |
| 9 | R9 | Siswa | 18 | 11 | 12 | 15 | 56 | 70 | 80,0 |
| 10 | R10 | Siswa | 17 | 12 | 15 | 15 | 59 | 70 | 84,2 |
| Total Skor | | | 166 | 119 | 119 | 180 | Rata – rata penerimaan pengguna(%) | | |
| Skor Maksimal | | | 200 | 150 | 150 | 200 | | | |
| Penerimaan Pengguna(%) (Total Skor/Skor Max) x 100% | | | 83,0 | 79,3 | 79,3 | 90,0 | | | |

Pengujian terhadap empat domain utama yang terdiri atas empat belas pertanyaan kuesioner terhadap sepuluh responden. Didapati hasil sebagai berikut, pada domain PU (*Perceived Usefulness*) didapati nilai sebesar 83,0% yang merepresentasikan “sangat dapat diterima” Pada domain PEOU (*Perceived Ease of Use*) didapati nilai sebesar 79,3% yang merepresentasikan “dapat diterima” pada domain ATU (*Attitude Towards Use*) didapati nilai sebesar 79,3% yang merepresentasikan “dapat diterima” dan pada domain BI (*Behavioural Intention*) didapati nilai sebesar 90% yang merepresentasikan “sangat dapat diterima” Hasil rata-rata penerimaan pengguna pada empat domain adalah 82,9% Yang merepresentasikan sangat dapat diterima.



BAB 5 PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dari penelitian yang telah dilakukan, dapat ditarik keimpulan sebagai berikut:

1. Sekolah Menengah Kejuruan 1 Muhammadiyah Kemlagi memiliki masalah dalam pelaksanaan kegiatan akademik sekolah yang perlu dioptimalisasi menggunakan sistem informasi akademik. Masalah tersebut berkenaan dengan pengolahan dan sinkronisasi data-data akademik. Pengolahan data akademik dengan penerapan teknologi informasi merupakan salah satu implementasi dari visi dan misi yang diusung oleh SMK Muhammadiyah 1 Kemlagi.
2. Pada pengembangan sistem informasi akademik (SIAKAD) pada SMK Muhammadiyah 1 Kemlagi diperlukan proses analisis kebutuhan yang baik sehingga kebutuhan akan sistem dapat teridentifikasi dengan baik. Pada proses elisitasi kebutuhan diperlukan penggalian secara mendalam akan kebutuhan sistem melalui wawancara dengan *stakeholder* terkait. Selain menggali data melalui *stakeholder*, analisis terhadap dokumentasi pelaksanaan kurikulum sekolah juga bisa dijadikan acuan dalam pemodelan proses bisnis yang akan diadaptasi oleh sistem informasi.
3. Penggunaan *waterfall model* dalam pengembangan sistem informasi akademik SMK Muhammadiyah 1 Kemlagi merupakan langkah yang tepat, karena kebutuhan fungsionalitas sistem dapat teridentifikasi secara mendalam pada saat elisitasi kebutuhan, serta kebutuhan yang telah teridentifikasi dipastikan tidak mengalami perubahan pada saat pengembangan sistem.
4. Pengujian yang dilakukan meliputi pengujian *black-box* dan pengujian UAT, pada pengujian *black-box* diperoleh validasi sistem telah 100% terpenuhi dan telah berjalan sesuai dengan yang diharapkan. Hasil pengujian UAT dengan melibatkan admin TU, admin BKK, waka kurikulum, 5 (lima) siswa dan 2 (dua) guru menunjukkan hasil penerimaan terhadap sistem sebesar 82,9%. Sehingga dapat diperoleh kesimpulan bahwa sistem informasi akademik SMK Muhammadiyah 1 Kemlagi sangat dapat diterima oleh pengguna sistem informasi.

5.2 Saran

Saran untuk penelitian selanjutnya yang memiliki relevansi dengan penelitian ini adalah sebagai berikut:

1. Dapat dilakukan peningkatan terhadap proses bisnis pengelolaan rapor siswa dan jadwal pelajaran, diharapkan proses bisnis tersebut dapat dilakukan dengan lebih cepat dengan mengurangi proses-proses yang berulang.
2. Dapat dilakukan peningkatan tampilan dengan menambahkan lebih banyak foto dokumentasi atau video yang berhubungan dengan SMK Muhammadiyah 1 Kemlagi.
3. Dapat ditambahkan beberapa fitur baru yang mencakup kegiatan yang berhubungan dengan kesiswaan, seperti penerimaan siswa baru (PPDB), pembayaran administrasi (SPP), kegiatan ekstrakurikuler sekolah, dan lain-lain.



DAFTAR PUSTAKA

- Abdullah, D., 2017. *Merancang Aplikasi Perpustakaan Menggunakan SDLC*. Sefa Bumi Persada: Lhokseumawe.
- Akbar, M. A., Tolle, H. & Raya, T. J., 2021. Pengembangan Aplikasi Sistem Bimbingan Penasehat Akademik (PA) Jurusan Sistem Informasi Filkom UB Berbasis Mobile Native Android. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, pp. 26-33.
- Akil, I., 2016. *Rekayasa Perangkat Lunak Dengan Model Unified Process Studi Kasus : Sistem Informasi Journal*. [Online] Available at: <https://www.neliti.com/id/publications/227534/rekayasa-perangkat-lunak-dengan-model-unified-process-studi-sistem-informa>
- Alam, G. A., Aknuranda, I. & Rachmadi, A., 2019. *Pemodelan dan Evaluasi Proses Bisnis Menggunakan Business Process Model and Notation (BPMN) (Studi Kasus: Percetakan Mabes Printing)*. [Online] Available at: <https://i-ptik.ub.ac.id/index.php/i-ptik/article/view/4172/1932> [Diakses 1 April 2021].
- Arda, F. R., Arwan, A. & Pradana, F., 2021. Pengembangan Sistem Manajemen Fasilitas Laboratorium (Studi Kasus: Laboratorium di Jurusan Biologi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Brawijaya). *Jurnal Pengembangan Teknologi Infomasi dan Ilmu Komputer*, pp. 154-163.
- Azis, M. F., 2005. *Object Oriented Programming dengan PHP5*. Jakarta: Elex Media Komputindo.
- Bootstrap Core Team, 2021. *Bootstrap Documentation*. [Online] Available at: <https://getbootstrap.com/docs/5.0/about/overview/> [Diakses 31 Januari 2021].
- Brugger, N. & Milligan, I., 2018. *The SAGE Handbook of Web History*. Thousand Oaks: SAGE.
- Codeigniter Foundation, 2021. *Codeigniter Documentation*. [Online] Available at: https://www.codeigniter.com/user_guide/concepts/mvc.html
- Davis, M. E. & Phillips, J. A., 2006. *Learning PHP & MySQL*. Sebastopol: O'Reilly Media.
- Fehily, C., 2008. *SQL*. 3rd penyunt. Berkeley: Peachpit Press.
- Google Inc, 2021. *Google Chrome*. [Online] Available at: <https://www.google.com/chrome/> [Diakses 31 Januari 2021].
- Ismanto, Hidayah, F. & Kristinanti, 2020. *Pemodelan Proses Bisnis Menggunakan Business Process Modelling Notation (BPMN) (Studi Kasus Unit Penelitian dan Pengabdian Kepada Masyarakat (P2KM) Akademi Komunitas Negeri Putra Sang Fajar Blitar)*. [Online]

- Available at: <http://www.jurnal.unublitar.ac.id/index.php/brilliant> [Diakses 1 April 2021].
- Jaya, E. A., Wedyawati, V. & Ridwan, M., 2018. *Perancangan Sistem Informasi Akademik pada Sekolah Menengah Kejuruan Negeri (SMKN) I Lintau Buo Menggunakan Php Mysql*. [Online] Available at: http://ojs.sttind.ac.id/sttind_ojs/index.php/Sain/article/view/119 [Diakses 8 Maret 2021].
- Junirianto, E., 2018. *Pemrograman WEB Dengan Framework Laravel*. Ponorogo: WADE Group.
- Kristanto, H., 1994. *Konsep & Perancangan Database*. Yogyakarta: Penerbit Andi.
- Limaye, M. G., 2009. *Software Testing*. s.l.:Tata McGraw-Hill Education.
- Mardiana, D. L., Aknurandha, I. & Mursityo, Y. T., 2018. *Analisis dan Perancangan Sistem Informasi Pelaporan Pendataan Keluarga Berencana Kabupaten Jombang Pada Dinas Pengendalian Penduduk dan Keluarga Berencana Kabupaten Jombang*. [Online] Available at: <http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/1595/811>
- Miles, R. & Hamilton, K., 2006. *Learning UML 2.0*. Sebastopol: O'Reilly Media.
- Mozilla Corporation, 2021. *Firefox Web Browser*. [Online] Available at: <https://www.mozilla.org/id/firefox/new/> [Diakses 31 Januari 2021].
- Muslihudin, M. & Oktafianto, 2016. *Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML*. Yogyakarta: Penerbit Andi.
- Nixon, R., 2018. *Learning PHP, MySQL & JavaScript, 5th Edition*. 5th penyunt. Sebastopol: O'Reilly Media.
- Nuraini, C. K., Mohd, C. K. & Shahbodin, F., 2016. *Personalized Learning Environment: Alpha Testing, Beta Testing & User Acceptance Test*. [Online] Available at: <http://www.sciencedirect.com/science/article/pii/S1877042815037982> [Diakses 2 Februari 2021].
- Nurhayati, L. & Setiadi, D., 2017. *Pemodelan Proses Bisnis (Studi Kasus PD. Simpati Sumedang)*. [Online] Available at: <https://ejournal.stmik-sumedang.ac.id/index.php/infomans/article/view/39> [Diakses 2 Februari 2021].
- Otaduy, I. & Diaz, O., 2017. *User acceptance testing for Agile-developed web-based applications: Empowering customers through wikis and mind maps*. [Online] Available at: <http://www.sciencedirect.com/science/article/pii/S016412121730002X> [Diakses 2 Februari 2021].

- Pargiyani, A., Priyambadha, B. & Arwan, A., 2019. Pengembangan Sistem Informasi Akademik Pada Sekolah Dasar Islam Terpadu Insantama (SDIT) Malang Berbasis Website. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, III(Vol. 3, No. 2, Februari 2019, hlm. 2121-2127), pp. 2121-2127.
- Pargiyani, A., Priyambadha, B. & Arwan, A., 2019. Pengembangan Sistem Informasi Akademik Pada Sekolah Dasar Islam Terpadu Insantama (SDIT) Malang Berbasis Website. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, pp. 2121-2127.
- Patma, T. S., Maskan, M. & Utaminingsih, A., 2018. *Sistem Informasi Manajemen Guna Mendukung Keputusan*. Malang: UPT Percetakan dan Penerbitan Polinema.
- Patsun, 2015. Paradigma Administrasi Dalam Pengelolaan Sekolah. *Cendekia: Jurnal Studi Keislaman*, pp. 171-181.
- Pratama, I. G. Y., Wicaksono, S. A. & Saputra, M. C., 2018. *Pengembangan Sistem Informasi Manajemen Perpustakaan Pada Ruang Baca Fakultas Ilmu Komputer Universitas Brawijaya*. [Online] Available at: <http://i-ptiik.ub.ac.id/index.php/i-ptiik/article/view/3687/1465>
- Pressman, R. S. & Maxim, B. R., 2015. *Software Engineering A Practitioner's Approach*. New York: Mc Graw Hill Education.
- Rahman, M. & Juliyanto, 2019. *Sistem Informasi Akademik SMA Nurul Falah Berbasis WEB*. [Online] Available at: <https://jitech.i-tech.ac.id/index.php/jitech/article/view/141/73> [Diakses 8 Maret 2021].
- Rahmatya, M. D. & Faris, M., 2018. *Pengembangan Sistem Informasi Akademik Berbasis WEB Pada SMA Pasundan 2 Bandung*. [Online] Available at: <https://ojs.unikom.ac.id/index.php/jamika/article/view/642> [Diakses 8 Maret 2021].
- Rahmawati, D., Rokhmawati, R. I. & Perdanakusuma, A. R., 2017. *Analisis dan Pemodelan Proses Bisnis Bidang Pelayanan Perizinan Menggunakan Business Process Model and Notation (BPMN)(Studi Pada Dinas Penanaman Modal dan Pelayanan Terpadu Satu Pintu Pemerintah Kota Malang)*. [Online] Available at: <https://i-ptiik.ub.ac.id/index.php/i-ptiik/article/view/469/194> [Diakses 1 April 2021].
- Roebuck, K., 2011. *Systems Development Life Cycle (SDLC): High-impact Strategies – What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors*. La Vergne: Lightning Source.
- Saputra, R. H., Baba, J. A. & Siregar, G. Y. K. S., 2018. *Penilaian Kinerja Dosen Menggunakan Modifikasi Skala Likert dengan Metode Simple Additive Weighting*. [Online] Available at: <http://jurnal.ulb.ac.id/index.php/explore/article/view/1029> [Diakses 2 Februari 2021].

- Satia, Z. P., 2015. *Sistem Informasi Sekolah Pengalaman Sekolah Sukma Bangsa*. Jakarta: Pustaka Alvabet.
- Setyawati, E., Sarwani, Wijoyo, H. & Soeharmoko, N., 2020. *Relational Database Management Systems (RDBMS)*. Banyumas: Pena Persada.
- Simarmata, J., 2010. *Rekayasa Perangkat Lunak*. Yogyakarta: Penerbit Andi.
- SMK Muhammadiyah 1 Kemlagi, 2016. *Dokumen Pelaksanaan Kegiatan Akademik*. Mojokerto, Humas SMK MUTU.
- Solichin, A., 2016. *Pemrograman WEB dengan PHP MySql*. Jakarta: Penerbit Budi Luhur.
- Sommerville, I., 2011. *Software Engineering*. Boston: Pearson.
- Subari, A., Tadeus, D. Y., Winarno, H. & Yuwono, T., 2018. *Gema Teknologi*. [Online] Available at: https://ejurnal.undip.ac.id/index.php/gema_teknologi/article/view/19147 [Diakses 31 Januari 2021].
- Sucipto, Indriati, R. & Hariawan, F. B., 2017. *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*. [Online] Available at: <http://jurnal.stkipgritulungagung.ac.id/index.php/jipi/article/view/367/220> [Diakses 31 Januari 2021].
- Sudirman, A. et al., 2020. *Sistem Informasi Manajemen*. Medan: Yayasan Kita Menulis.
- Sutabri, T., 2012. *Konsep Sistem Informasi*. Yogyakarta: Penerbit Andi.
- Sutabri, T., 2012. *Konsep Sistem Informasi*. Yogyakarta: Penerbit Andi.
- Wadi, H., 2020. *Panduan Praktis Codeigniter Untuk Pemula : Langkah demi langkah memahami dan memulai Codeigniter untuk membuat web dinamis*. Jakarta: TR Publisher.
- Wahana Komputer, 2000. *Desain Web dengan Microsoft FrontPage 97*. Yogyakarta: Penerbit Andi.
- Wahana Komputer, 2010. *Panduan Belajar MySQL Database Server*. Jakarta: Media Kita.
- Waluyo, A. & Munawar, A., 2017. *Jurnal Sisfokom*. [Online] Available at: <http://jurnal.atmaluhur.ac.id/index.php/sisfokom/article/view/44> [Diakses 21 Februari 2021].
- Wardani, E. S., Putranto, H. & Wibawa, A. P., 2017. *Sistem Informasi Di SMK dan Upaya Peningkatan Kinerjanya*. [Online] Available at: <https://media.neliti.com/media/publications/224395-sistem-informasi-di-smk-dan-upaya-pening-c2c623b5.pdf> [Diakses 21 Februari 2021].
- Whitten, J. L. & Bentley, L. D., 2007. *Systems Analysis and Design Methods*. New York: McGraw-Hill Irwin.

Lampiran A Hasil Wawancara

LAMPIRAN

LEMBAR HASIL WAWANCARA

Topik : Penggalian Latar Belakang dan Kebutuhan Pengembangan Sistem Informasi

Hari/Tanggal : Rabu 24 Februari 2021

Tempat : SMK Muhammadiyah 1 Kemlagi

Pewawancara : Muhammad Taftiyah Ilham Akbar (175150601111001)

Narasumber : Bpk. Nur Awalludin, S.T. (Kepala BKK)

A.1 Penggalian Latar Belakang

1. Bagaimana pelaksanaan kegiatan akademik pada SMK Muhammadiyah 1 Kemlagi?

Jawaban :

Pelaksanaan kegiatan akademik di SMK Mutu hampir sama dengan sekolah menengah kejuruan pada umumnya, seperti mewajibkan praktik kerja industri, melakukan evaluasi di tengah semester dan akhir semester.

2. Dalam pelaksanaan kegiatan akademik sekolah apakah SMK Muhammadiyah 1 Kemlagi menggunakan bantuan sistem informasi?

Jawaban:

Saat ini masih belum menggunakan sistem informasi. Namun sudah memiliki web profil sekolah.

3. Apakah ada proses manajemen sekolah yang perlu segera menerapkan sistem informasi?

Jawaban:

Saat ini masih yang menjadi prioritas adalah pencatatan data-data yang berhubungan dengan akademik.

4. Bagaimana proses pencatatan data-data akademik dilakukan saat ini?

Jawaban:

Dilakukan dengan menggunakan program Microsoft Excell yang diperbarui berkala secara manual.

5. Mengapa proses akademik perlu segera menerapkan sistem informasi?

Jawaban:

- Supaya data dapat lebih valid.
- Update data lebih mudah.
- Pelaporan data lebih mudah.

LEMBAR HASIL WAWANCARA

- Seiring bertambahnya alumni memerlukan pencatatan yang baik dan relevan.
 - Mengurangi pekerjaan yang tidak perlu.
6. Apa saja kendala yang dialami oleh stakeholder dalam pelaksanaan kegiatan akademik yang saat ini perlu menerapkan sistem informasi?

Jawaban:

Pembaruan berkala dilakukan secara manual sehingga sering terjadi inkonsistensi/data tidak sinkron antara data yang dilaporkan dengan data di lapangan.

7. Hasil apa yang diharapkan dalam penerapan sistem informasi akademik sekolah?

Jawaban:

Terwujudnya sistem pencatatan yang konsisten, mudahnya pencarian data, pelaporan yang lebih mudah, dan pencatatan yang lebih mudah.

8. Apakah saat ini di lingkungan sekolah memiliki jaringan internet dan perangkat komputer yang memadai (dapat mengakses peramban web)?

Jawaban:

Memadai

A.2 Penggalian Kebutuhan Sistem Informasi

1. Bagaimana proses bisnis yang diharapkan dalam penerapan sistem informasi akademik sekolah?

Jawaban:

Diharapkan sistem informasi dapat mempercepat proses bisnis dengan akses yang lebih mudah dan dapat dipahami dengan cepat oleh para pemangku kepentingan.

2. Dalam sistem informasi yang akan dikembangkan, siapa saja *stakeholder* yang melakukan interaksi secara langsung dengan sistem?

Jawaban:

Sasaran pengguna sistem informasi adalah sebagai berikut :

- Admin Sistem
- Admin BKK
- Waka Kurikulum
- Guru
- Siswa

3. Bagaimanakah penjabaran dari tugas dan wewenang dari masing stakeholder tersebut dalam penggunaan sistem informasi akademik SMK Muhammadiyah 1 Kemlagi?

LEMBAR HASIL WAWANCARA

Jawaban:

Masing-masing pengguna punya akses fitur yang berbeda-beda sesuai dengan batasan yang diterapkan oleh sekolah, teknisnya adalah sebagai berikut :

A. Admin Sistem:

- Mengelola data jurusan.
- Mengelola data kelas.
- Mengelola data guru.
- Mengelola data siswa.
- Mengelola jadwal pelajaran.
- Mengelola kalender akademik.

B. Admin BKK:

- Mengelola data lowongan pekerjaan.
- Memvalidasi pelaksanaan prakerin.
- Melihat laporan pelaksanaan prakerin.
- Mengelola data alumni.

C. Waka Kurikulum

- Memvalidasi kalender akademik.
- Memvalidasi penentuan wali kelas.
- Melihat data guru.
- Melihat data alumni.

D. Guru

- Memperbarui data pribadi.
- Memasukkan data hasil evaluasi siswa.
- Melihat informasi siswa perkelas.
- Melihat seluruh jadwal pelajaran.

E. Siswa :

- Memperbarui data diri.
- Mengajukan pelaksanaan prakerin.
- Melihat jadwal pelajaran.
- Melihat hasil evaluasi tengah semester dan evaluasi akhir semester.
- Melihat Kalender Akademik.

LEMBAR HASIL WAWANCARA

4. Informasi-informasi apa saja yang perlu diproses dalam sistem informasi?

Jawaban:

Informasi-informasi umum yang diperlukan untuk pengarsipan dan pencatatan, lebih detailnya adalah sebagai berikut :

A. Siswa

- NISN
- Nama Lengkap
- TTL
- Jurusan
- Alamat
- Agama
- Anak ke
- Status Akademik
 - Aktif/Lulus/Pindah/Dikeluarkan
- Nomor Telepon
- Informasi Kelas
 - Kelas X
 - Kelas XI
 - Kelas XII
- Status Prakerin
 - Sudah Prakerin/Belum Prakerin/Sedang Prakerin
- Informasi Orang tua/Wali
 - Nama Ayah
 - Nama Ibu
 - Pekerjaan Ayah
 - Pekerjaan Ibu
 - Nomor Telepon Ayah
 - Nomor Telepon Ibu
 - Alamat Orang Tua
- Sekolah Sebelumnya
 - Nama Sekolah

LEMBAR HASIL WAWANCARA

- Diterima di kelas
- Tanggal Penerimaan

B. Guru

- NBM
- Nama Lengkap
- Jenis Kelamin
- Agama
- Alamat
- TTL
- Pendidikan Terakhir
- Jabatan
- Masa Mengabdi
- Status

C. Prakerin

- Tempat Prakerin
- Deskripsi Tempat Prakerin
- Tanggal Mulai Prakerin
- Tanggal Akhir Prakerin

D. BKK

- Nama Pekerjaan
- Deskripsi Pekerjaan
- Batas Pendaftaran
- Narahubung

5. Dalam pemrosesan informasi, data apa saja yang perlu dilakukan proses validasi?

Jawaban:

- Pengajuan pelaksanaan prakerin.
- Pengajuan status alumni.
- Kalender Akademik.
- Penempatan guru sebagai wali kelas.

LEMBAR HASIL WAWANCARA

6. Bagaimana proses pengaksesan sistem akademik sekolah SMK Muhammadiyah 1 Kemlagi?

Jawaban:

Dapat diakses online melalui web browser. Pengguna perlu login terlebih dahulu pada sistem.

7. Apakah terdapat kebutuhan lain yang diharapkan diluar kebutuhan utama sistem (kebutuhan non fungsional)?

Jawaban:

Dapat diakses melalui perangkat seluler dengan baik. Tampilan sistem informasi tidak terlalu rumit.

Mengetahui

Waka Kurikulum



Mengetahui

Kepala Bursa Kerja Khusus



Lampiran B Kuesioner UAT**LEMBAR UJI PENERIMAAN PENGGUNA (UAT)****KUESIONER PENGUJIAN PENERIMAAN PENGGUNA SISTEM INFORMASI AKADEMIK SMK
MUHAMMADIYAH 1 KEMLAGI**

Nama Lengkap :

Jabatan :

A. Petunjuk Pengisian

1. Mohon ketersediaan Bapak/Ibu/Saudara untuk mengisi butir-butir pertanyaan pada kuesioner dibawah sesuai dengan persepsi Bapak/Ibu/Saudara.
2. Bapak/Ibu/Saudara dapat memberikan *checkbox* () pada kolom jawaban yang ada pada tabel dibawah. Jawaban terdiri dari 5 (lima) alternatif jawaban, keterangan dari masing-masing alternatif jawaban adalah sebagai berikut:
 - SS = Sangat Setuju (5)
 - S = Setuju (4)
 - N = Netral (3)
 - TS = Tidak Setuju (2)
 - STS = Sangat Tidak Setuju (1)

B. Kuesioner Pengujian

Berikut merupakan butir-butir pertanyaan mengenai persepsi penerimaan pengguna terhadap sistem informasi akademik SMK Muhammadiyah 1 Kemlagi.

1. Persepsi kebergunaan sistem informasi (*Perceived of Usefulness*).

| No | Pernyataan | Pilihan Jawaban | | | | |
|----|---|-----------------|---|---|----|-----|
| | | SS | S | N | TS | STS |
| 1 | Penggunaan sistem informasi akademik SMK Muhammadiyah 1 Kemlagi mempercepat pekerjaan saya. | | | | | |
| 2 | Penggunaan sistem informasi akademik SMK Muhammadiyah 1 Kemlagi membuat pekerjaan yang ada lebih efektif dan efisien. | | | | | |

LEMBAR UJI PENERIMAAN PENGGUNA (UAT)

| No | Pernyataan | Pilihan Jawaban | | | | |
|----|---|-----------------|---|---|----|-----|
| | | SS | S | N | TS | STS |
| 3 | Penggunaan sistem informasi akademik SMK Muhammadiyah 1 Kemlagi membuat saya menjadi lebih produktif. | | | | | |
| 4 | Sistem informasi akademik SMK Muhammadiyah 1 Kemlagi memiliki manfaat untuk saya. | | | | | |

2. Persepsi kemudahan penggunaan sistem informasi (*Perceive Ease of Use*).

| No | Pernyataan | Pilihan Jawaban | | | | |
|----|---|-----------------|---|---|----|-----|
| | | SS | S | N | TS | STS |
| 1 | Sistem informasi akademik SMK Muhammadiyah 1 Kemlagi sangat mudah untuk saya pelajari. | | | | | |
| 2 | Saya dapat cepat beradaptasi dengan sistem informasi akademik SMK Muhammadiyah 1 Kemlagi. | | | | | |
| 3 | Saya dapat menjadi ahli dan terampil dalam penggunaan sistem informasi akademik SMK Muhammadiyah 1 Kemlagi. | | | | | |

3. Sikap/Pandangan terhadap penggunaan sistem informasi (*Attitude Toward Using*).

| No | Pernyataan | Pilihan Jawaban | | | | |
|----|--|-----------------|---|---|----|-----|
| | | SS | S | N | TS | STS |
| 1 | Saya menyukai sistem informasi akademik SMK Muhammadiyah 1 Kemlagi. | | | | | |
| 2 | Penerapan sistem informasi akademik di SMK Muhammadiyah 1 Kemlagi adalah ide yang bagus. | | | | | |

LEMBAR UJI PENERIMAAN PENGGUNA (UAT)

| No | Pernyataan | Pilihan Jawaban | | | | |
|----|--|-----------------|---|---|----|-----|
| | | SS | S | N | TS | STS |
| 3 | Melakukan pekerjaan dengan sistem informasi akademik SMK Muhammadiyah 1 Kemlagi akan lebih menyenangkan. | | | | | |

4. Minat terhadap penggunaan sistem informasi (*Behavioural Intention*).

| No | Pernyataan | Pilihan Jawaban | | | | |
|----|---|-----------------|---|---|----|-----|
| | | SS | S | N | TS | STS |
| 1 | Jika ada pilihan lain, saya lebih menyukai penggunaan sistem informasi akademik SMK Muhammadiyah 1 Kemlagi dalam menjalankan tugas atau pekerjaan saya. | | | | | |
| 2 | Jika ada pilihan, saya lebih meminati sistem informasi akademik SMK Muhammadiyah 1 Kemlagi dalam menjalankan tugas atau pekerjaan saya. | | | | | |
| 3 | Jika ada pilihan lain, saya akan memilih untuk tetap menggunakan sistem informasi akademik SMK Muhammadiyah 1 Kemlagi. | | | | | |
| 4 | Jika ada pilihan lain, saya akan merekomendasikan penggunaan sistem informasi akademik SMK Muhammadiyah 1 Kemlagi ini pada orang lain. | | | | | |

C. Saran dan Masukkan



Lampiran C Rekapitulasi Hasil Uji Penerimaan Pengguna

| No | Identitas Responden | PU | SKOR JAWABAN | | | | | | | | | | Total Skor Maksimal | Skor Maksimal (%) | Penerimaan Pengguna (%) | | | |
|----|---------------------------|--|-------------------------------|------------------------|------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|---|-------------------------|------|----|------|
| | | | P | P | P | P | P | P | P | P | P | Bi | | | | | | |
| 1 | Suwandi S.T., M.Pd | Waka Kunkulum | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 56 | 70 | 80,0 | | |
| 2 | Nur Awaludin, S.T | Ketua BKK | 4 | 5 | 4 | 4 | 5 | 4 | 4 | 4 | 5 | 3 | 4 | 4 | 4 | 56 | 70 | 90,0 |
| 3 | Azzahro Wachida, S.I.Pust | Tata Usaha | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 59 | 70 | 84,2 |
| 4 | Tibyanu A.R, S.Sos | Guru | 4 | 4 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 4 | 4 | 4 | 4 | 59 | 70 | 84,2 |
| 5 | Ayu Azuririn, M.Si | Guru | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 4 | 4 | 4 | 4 | 4 | 4 | 64 | 70 | 91,4 |
| 6 | Dia Puspita | Siswi | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 3 | 4 | 4 | 4 | 4 | 4 | 54 | 70 | 77,1 |
| 7 | Anggraini Dwip | Siswi | 4 | 4 | 4 | 4 | 4 | 4 | 3 | 3 | 4 | 4 | 3 | 4 | 4 | 52 | 70 | 74,2 |
| 8 | Muhammad Bugi | Siswi | 4 | 4 | 4 | 4 | 4 | 4 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 53 | 70 | 75,7 |
| 9 | Anggun Dea P | Siswi | 5 | 5 | 4 | 4 | 4 | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 3 | 56 | 70 | 80,0 |
| 10 | Dela Yosinta Ashari | Siswi | 3 | 4 | 5 | 4 | 5 | 3 | 5 | 5 | 4 | 4 | 4 | 4 | 3 | 59 | 70 | 84,2 |
| | | Total Skor | 166 | 119 | 119 | 180 | 180 | 200 | 200 | 200 | 200 | 200 | 200 | Rata-rata Penerimaan Pengguna(%) | | | | |
| | | Penerimaan Pengguna(%) (Total Skor/Skor Maksimal) x 100% | 83,0 | 79,3 | 79,3 | 90,0 | 90,0 | 100 | 82,9 | | | |
| | | Representasi Penerimaan Pengguna | Sangat dapat diterima. | Dapat diterima. | Dapat diterima. | Sangat dapat diterima. | | | | |



Gambar 5. 1 Penjelasan Sistem Informasi Pada Waka Kurikulum



Gambar 5. 2 Pengujian Sistem Informasi Oleh Admin BKK



Gambar 5. 3 Pengisian Kuesioner Oleh Siswa