

**PENGEMBANGAN SISTEM INFORMASI MANAJEMEN
AKADEMIK SEKOLAH MENENGAH ATAS
(STUDI KASUS PADA SMA SEJAHTERA 1 DEPOK)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Ammar Burhanuddin Sayuti

NIM: 155150401111063



PROGRAM STUDI SISTEM INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019

PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI MANAJEMEN
AKADEMIK SEKOLAH MENENGAH ATAS
(STUDI KASUS PADA SMA SEJAHTERA 1 DEPOK)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

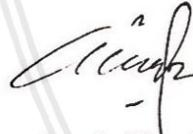
Disusun Oleh :
Ammar Burhanuddin Sayuti
NIM: 155150401111063

Skripsi ini telah diuji dan dinyatakan lulus pada
17 Juni 2019

Telah diperiksa dan disetujui oleh:

Pembimbing I

Pembimbing II



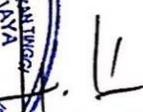
Dr.Eng. Ahmad Afif Supianto, S.Si., M.Kom.
NIK: 2012018206231001



Welly Purnomo, S.T., M.Kom.
NIK: 2017088101171001

Mengetahui
Ketua Jurusan Sistem Informasi




Herman Tolle, S.T., M.T.
NIP: 197408232000121001



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, 17 Juni 2019



Ammar Burhanuddin Sayuti

NIM: 155150401111063

PRAKATA

Puji syukur kehadirat Allah SWT atas segala berkat dan rahmat yang telah diberikan sehingga penulis dapat menyelesaikan skripsi ini yang berjudul “Pengembangan Sistem Informasi Manajemen Akademik Sekolah Menengah Atas (Studi Kasus Pada SMA Sejahtera 1 Depok)”.

Penyusunan dan penulisan skripsi ini tidak terlepas dari bantuan, bimbingan, dukungan, serta motivasi dari berbagai pihak. Namun, penulis dapat melewati hal tersebut karena anugerah Allah SWT, semangat dan dukungan dari beberapa pihak. Oleh karena itu, dalam kesempatan ini penulis dengan senang hati menyampaikan rasa terimakasih kepada :

1. Bapak Dr.Eng. Ahmad Afif Supianto, S.Si, M.Kom. selaku dosen pembimbing satu dan juga Bapak Welly Purnomo, S.T., M.Kom. selaku dosen pembimbing dua yang telah banyak memberikan ilmu, saran, motivasi dan do’a untuk penyelesaian skripsi ini.
2. Bapak Dr. Eng., Herman Tolle, S.T., M.T. selaku Ketua Jurusan Sistem Informasi Universitas Brawijaya.
3. Yusi Tyroni Mesityo, S.Kom.,M.AB. selaku Ketua Program Studi Sistem Informasi Brawijaya.
4. Umi Sosiawati dan Alm. Abi Aditya Burhanuddin sebagai orang tua penulis yang senantiasa memberikan do’a dan dukungan baik dalam bentuk moril maupun materil kepada penulis sehingga skripsi ini dapat terselesaikan. Serta sebagai motivasi kepada penulis untuk menyelesaikan skripsi.
5. Bapak Siswanto selaku guru BPTIK SMA Sejahtera 1 Depok yang telah memberikan izin kepada penulis untuk melakukan penelitian skripsi pada SMA Sejahtera 1 Depok. Serta selalu membantu dan mensupport penulis dalam menyelesaikan skripsi. Serta membantu memberikan masukan-masukan terhadap pengembangan sistem informasi ini.
6. Achmad Firmansyah, Bhanio Gemilang Putakaz, Chaerul Irfan Nugraha, Fikri Rizky Adinata, Irvan Dwiantono Kartomiharjo, Muhammad Arroofi Arga Kusumah, Nofry Arman, dan Rivalno Al Fath Ismubandono sebagai sahabat Lulusan Whatsapp yang selalu membantu penulis, serta memberikan semangat, motivasi, canda tawa, dan dukungan kepada penulis untuk menyelesaikan penelitian skripsi.
7. Faris Bagaskoro dan Maulida Sabrina selaku sahabat yang senantiasa menemani penulis di hari-hari yang penuh kebahagiaan dan kesedihan serta memberikan nasehat, saran dan dukungannya kepada penulis.
8. Bachtiar Eko Cahyo, Willy Aditya Nugraha, dan Widhi Arya sebagai sahabat seperbimbingan yang selalu membantu dan mendukung penulis, serta memberikan semangat, motivasi, canda tawa, dan dukungan kepada penulis untuk menyelesaikan penelitian skripsi.
9. Teman sepermainan berduabelas yang selalu memberikan semangat dan motivasi penulis dalam menyelesaikan penelitian skripsi.

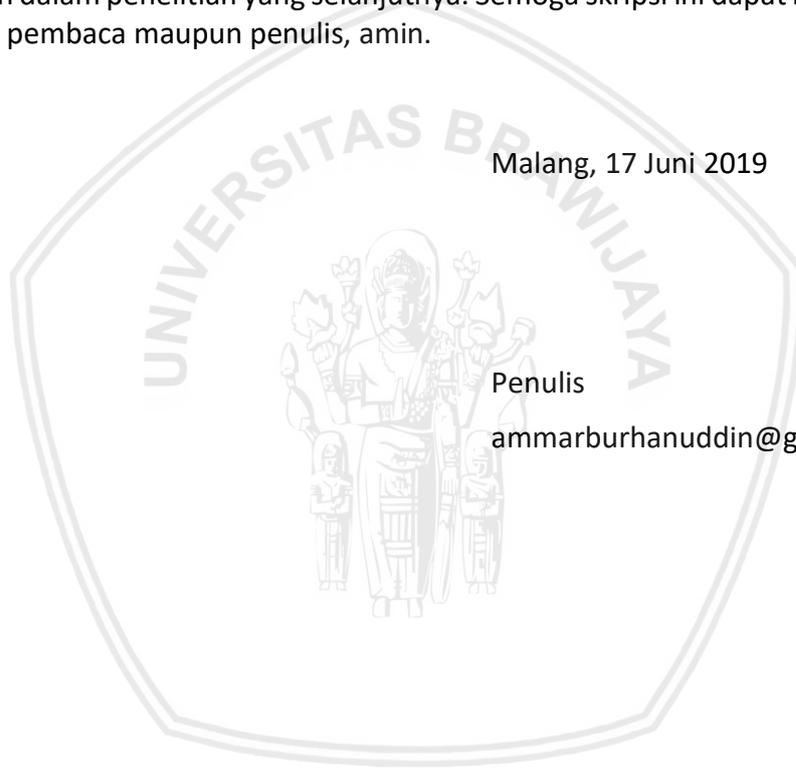
10. Teman-teman TEDxUniversitasBrawijaya yang telah menjadi sarana bagi penulis untuk berkembang dan memberikan kesempatan untuk aktif berorganisasi dan mendapatkan pengalaman yang berharga.
11. Teman-teman Keluarga Besar Mahasiswa Sistem Informasi yang tidak dapat penulis sebutkan Namanya satu per satu, yang telah menjadi teman yang baik bagi penulis, dan senantiasa memberikan dukungan moril dan motivasi dalam menyelesaikan penelitian skripsi ini, sehingga penulis dapat menyelesaikan skripsi ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan baik format laporan maupun isinya. Oleh karena itu, penulis mengharapkan saran dan kritik yang membangun dari para pembaca guna perbaikan dalam penelitian yang selanjutnya. Semoga skripsi ini dapat bermanfaat baik bagi pembaca maupun penulis, amin.

Malang, 17 Juni 2019

Penulis

ammarburhanuddin@gmail.com



ABSTRAK

Ammar Burhanuddin Sayuti, Pengembangan Sistem Informasi Manajemen Akademik Sekolah Menengah Atas (Studi Kasus Pada SMA Sejahtera 1 Depok)

Pembimbing: Dr.Eng. Ahmad Afif Supianto, S.Si, M.Kom dan Welly Purnomo, S.T., M.Kom

Sekolah Menengah Atas Sejahtera 1 Depok merupakan salah satu instansi pendidikan di kota Depok di bawah naungan Yayasan keluarga Sejahtera Bandung. Tugas pokok SMA Sejahtera 1 adalah menjalankan proses pembelajaran, evaluasi hasil belajar peserta didik, dan meluluskan peserta didik yang berkualitas memenuhi standar yang dipersyaratkan. Berdasarkan hasil wawancara dengan Bapak Siswanto selaku Guru BPTIK (Bimbingan dan Penyuluhan TIK) mengatakan bahwa terdapat kendala yang dihadapi, proses Pendaftaran Peserta Didik Baru masih dilakukan secara manual. Pada proses evaluasi siswa, distribusi penilaian untuk rapor siswa sangat kompleks, jadi harus melalui berbagai macam tahap. Sehingga berdasarkan kendala yang sudah dijelaskan sebelumnya diperlukan suatu sistem informasi yang dapat membantu menangani masalah tersebut. Pengembangan sistem informasi manajemen akademik sekolah dilakukan dengan menggunakan *Waterfall model* dan diimplementasikan dengan aplikasi berbasis web. Tahapan yang dilakukan yaitu analisis kebutuhan, perancangan, implementasi, dan pengujian. Dari hasil pengujian *validation testing* sistem berdasarkan pada kebutuhan fungsional didapatkan hasil valid dan pada pengujian *User Acceptance Testing* yang diberikan kepada pihak guru BPTIK sebagai admin (nilai persentase UAT sebesar 90%), pihak guru selaku penilai siswa (nilai persentase UAT sebesar 90%), dan pihak calon siswa sebagai pendaftar (nilai persentase UAT sebesar 85%).

Kata kunci: sistem informasi manajemen akademik sekolah, *Waterfall Model*

ABSTRACT

Ammar Burhanuddin Sayuti, *Development of Senior High School Academic Management Information System (Case Study Sejahtera 1 Depok Senior High School)*

Supervisor: Dr.Eng. Ahmad Afif Supianto, S.Si, M.Kom dan Welly Purnomo, S.T., M.Kom

Sejahtera 1 Depok Senior High School is one of the educational institutions in Depok city under the auspices of the family sejahtera bandung foundation. Main tasks Sejahtera 1 Senior High School is to run the learning process, evaluate the learning outcomes, and pass quality students who meet the required standarts. Based on interview with mr. Siswanto as BPTIK Teacher said that some problem encountered, the process of registration new students is still manually. In the process evaluate student, the distribution of scores for student report is very complex, so have to go throught several s phase. Therefore the need for information systems that can help deal with the problem. Development of school academic management information systems using waterfall model and implemented on website. Which is conducted phase is requirement analysis, design, implementation, and testing. After the result of the User Acceptance Test given to the BPTIK teacher as a admin (the percentage of UAT by 90%), the teacher as student assessor (the percentage of UAT by 90%), and prospective student as registrant (the percentage of UAT by 85%).

Keywords: school academic management information systems, waterfall model

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN	xvi
BAB 1 PENDAHULUAN.....	Error! Bookmark not defined.
1.1 Latar Belakang.....	Error! Bookmark not defined.
1.2 Rumusan Masalah.....	Error! Bookmark not defined.
1.3 Tujuan	Error! Bookmark not defined.
1.4 Manfaat.....	Error! Bookmark not defined.
1.5 Batasan Masalah.....	Error! Bookmark not defined.
1.6 Sistematika Pembahasan.....	Error! Bookmark not defined.
BAB 1 PENDAHULUAN.....	Error! Bookmark not defined.
BAB 2 LANDASAN KEPUSTAKAAN.....	Error! Bookmark not defined.
BAB 3 METODOLOGI PENELITIAN	Error! Bookmark not defined.
BAB 4 ANALISIS KEBUTUHAN	Error! Bookmark not defined.
BAB 5 PERANCANGAN	Error! Bookmark not defined.
BAB 6 IMPLEMENTASI	Error! Bookmark not defined.
BAB 7 PENGUJIAN	Error! Bookmark not defined.
BAB 8 PENUTUP.....	Error! Bookmark not defined.
BAB 2 LANDASAN KEPUSTAKAAN	Error! Bookmark not defined.
2.1 Kajian Pustaka	Error! Bookmark not defined.
2.2 Sekolah Menengah Atas Sejahtera 1 Depok	Error! Bookmark not defined.
2.3 Sistem Informasi	Error! Bookmark not defined.
2.3.1 Sistem Informasi Manajemen	Error! Bookmark not defined.



2.4	Rekayasa Perangkat Lunak	Error! Bookmark not defined.
2.5	<i>Software Development Life Cycle</i>	Error! Bookmark not defined.
2.6	<i>Waterfall Model</i>	Error! Bookmark not defined.
2.6.1	<i>Communication (Project Initiation & Requirements Gathering)</i>	Error! Bookmark not defined.
2.6.2	<i>Planning (Estimating, Scheduling, Tracking)</i>	Error! Bookmark not defined.
2.6.3	<i>Modeling (Analysis & Design)</i>	Error! Bookmark not defined.
2.6.4	<i>Construction (Code & Test)</i>	Error! Bookmark not defined.
2.6.5	<i>Deployment (Delivery, Support, Feedback)</i> .	Error! Bookmark not defined.
2.7	<i>Requirement Gathering</i>	Error! Bookmark not defined.
2.8	Proses Bisnis.....	Error! Bookmark not defined.
2.8.2	<i>Business Process Model and Notation (BPMN)</i>	Error! Bookmark not defined.
2.9	Perancangan Sistem.....	Error! Bookmark not defined.
2.10	UML (<i>Unified Modelling Language</i>).....	Error! Bookmark not defined.
2.10.1	<i>Use Case Diagram</i>	Error! Bookmark not defined.
2.10.2	<i>Activity Diagram</i>	Error! Bookmark not defined.
2.10.3	<i>Sequence Diagram</i>	Error! Bookmark not defined.
2.10.1	<i>Class Diagram</i>	Error! Bookmark not defined.
2.10.2	<i>Physical Data Model</i>	Error! Bookmark not defined.
2.11	Implementasi Sistem	Error! Bookmark not defined.
2.11.1	<i>Object Oriented Programming (OOP)</i>	Error! Bookmark not defined.
2.11.2	Codeigniter	Error! Bookmark not defined.
2.11.3	Aplikasi berbasis Web	Error! Bookmark not defined.
2.12	Pengujian Sistem.....	Error! Bookmark not defined.
2.12.1	<i>Black-Box Testing</i>	Error! Bookmark not defined.
2.12.2	<i>User Acceptance Testing (UAT)</i>	Error! Bookmark not defined.
BAB 3	METODOLOGI	Error! Bookmark not defined.
3.1	<i>Communication (Requirements Gathering)</i>	Error! Bookmark not defined.
3.2	<i>Planning (Estimating, Scheduling, Tracking)</i>	Error! Bookmark not defined.
3.3	<i>Modeling (Analysis & Design)</i>	Error! Bookmark not defined.

3.4 Construction (Code & Test)	Error! Bookmark not defined.
BAB 4 ANALISIS KEBUTUHAN	Error! Bookmark not defined.
4.1 Aturan Penomoran	Error! Bookmark not defined.
4.2 Pemodelan Proses Bisnis Saat Ini.....	Error! Bookmark not defined.
4.3 Analisis Proses Bisnis Yang Sedang Berjalan (<i>as-is</i>)	Error! Bookmark not defined.
4.4 Pemodelan Proses Bisnis Usulan	Error! Bookmark not defined.
4.5 Identifikasi Stakeholder	Error! Bookmark not defined.
4.6 Identifikasi Fitur	Error! Bookmark not defined.
4.7 Kebutuhan Fungsional	Error! Bookmark not defined.
4.8 Kebutuhan nonfungsional	Error! Bookmark not defined.
4.9 Use Case Diagram	Error! Bookmark not defined.
4.10 Use Case Scenario.....	Error! Bookmark not defined.
BAB 5 perancangan sistem.....	Error! Bookmark not defined.
5.1 Activity Diagram	Error! Bookmark not defined.
5.1.1 Mengisi Formulir Pendaftaran Peserta Didik Baru / PPDB	Error! Bookmark not defined.
5.1.2 Melakukan Verifikasi Pendaftaran	Error! Bookmark not defined.
5.1.3 Memasukkan Detail Nilai	Error! Bookmark not defined.
5.2 Sequence Diagram	Error! Bookmark not defined.
5.2.1 Mengisi Formulir Pendaftaran Peserta Didik Baru / PPDB	Error! Bookmark not defined.
5.2.2 Melakukan Verifikasi Pendaftaran	Error! Bookmark not defined.
5.2.3 Memasukkan Detail Nilai	Error! Bookmark not defined.
5.3 Class Diagram	Error! Bookmark not defined.
5.4 Physical Data Model	Error! Bookmark not defined.
5.5 Perancangan Antarmuka Pengguna.....	Error! Bookmark not defined.
5.5.1 Antarmuka Pengguna Autentikasi	Error! Bookmark not defined.
5.5.2 Antarmuka Pengguna Mengisi Formulir Pendaftaran Peserta Didik Baru	Error! Bookmark not defined.
5.5.3 Antarmuka Pengguna Melakukan Verifikasi Pendaftaran	Error! Bookmark not defined.

5.5.4 Antarmuka Pengguna Memasukkan Detail Nilai	Error! Bookmark not defined.
BAB 6 IMPLEMENTASI	Error! Bookmark not defined.
6.1 Spesifikasi Sistem	Error! Bookmark not defined.
6.1.1 Spesifikasi Perangkat Keras	Error! Bookmark not defined.
6.1.2 Spesifikasi Perangkat Lunak	Error! Bookmark not defined.
6.2 Struktur Artefak Sistem	Error! Bookmark not defined.
6.3 Implementasi Algoritme	Error! Bookmark not defined.
6.3.1 Mengisi Formulir Pendaftaran Peserta Didik Baru	Error! Bookmark not defined.
6.3.2 Melakukan Verifikasi Pendaftaran	Error! Bookmark not defined.
6.3.3 Memasukkan Detail Nilai	Error! Bookmark not defined.
6.3.4 Melihat Detail Nilai	Error! Bookmark not defined.
6.4 Implementasi Antarmuka Pengguna	Error! Bookmark not defined.
6.4.1 Antarmuka Pengguna Autentikasi	Error! Bookmark not defined.
6.4.2 Antarmuka Pengguna Mengisi Formulir Pendaftaran Peserta Didik Baru	Error! Bookmark not defined.
6.4.3 Antarmuka Pengguna Melakukan Verifikasi Pendaftaran	Error! Bookmark not defined.
6.4.4 Antarmuka Pengguna Memasukkan Detail Nilai	Error! Bookmark not defined.
BAB 7 PENGUJIAN	Error! Bookmark not defined.
7.1 <i>Validation Testing</i>	Error! Bookmark not defined.
7.1.1 <i>Validation Testing</i> Mengisi Formulir Pendaftaran Peserta Didik Baru	Error! Bookmark not defined.
7.1.2 <i>Validation Testing</i> Memasukkan Detail Nilai	Error! Bookmark not defined.
7.1.3 Analisis Hasil Pengujian Sistem	Error! Bookmark not defined.
7.2 <i>Compability Testing</i>	Error! Bookmark not defined.
7.3 <i>User Acceptance Testing (UAT)</i>	Error! Bookmark not defined.
7.3.1 Perancangan Pengujian UAT	Error! Bookmark not defined.
7.3.2 Hasil Pengujian UAT	Error! Bookmark not defined.
7.3.3 Analisis Hasil Pengujian UAT	Error! Bookmark not defined.
BAB 8 Penutup	Error! Bookmark not defined.

8.1 Kesimpulan.....**Error! Bookmark not defined.**

8.2 Saran**Error! Bookmark not defined.**

DAFTAR REFERENSI**Error! Bookmark not defined.**

LAMPIRAN A HASIL WAWANCARA.....**Error! Bookmark not defined.**

LAMPIRAN B HASIL PENGUJIAN *USER ACCEPTANCE TESTING* (UAT).....**Error! Bookmark not defined.**

LAMPIRAN C ALUR PENDAFTARAN PESERTA DIDIK BARU**Error! Bookmark not defined.**

LAMPIRAN D TEMPLATE RAPORT**Error! Bookmark not defined.**



DAFTAR TABEL

Tabel 2.1 Rangkuman Kajian Pustaka	Error! Bookmark not defined.
Tabel 2.2 Notasi <i>BPMN</i>	Error! Bookmark not defined.
Tabel 2.3 Simbol – Simbol dalam <i>Use Case Diagram</i>	Error! Bookmark not defined.
Tabel 2.4 Simbol – simbol dalam <i>Activity Diagram</i> ..	Error! Bookmark not defined.
Tabel 2.5 Simbol – Simbol dalam <i>Sequence Diagram</i>	Error! Bookmark not defined.
Tabel 2.6 Simbol – Simbol dalam <i>Class Diagram</i>	Error! Bookmark not defined.
Tabel 2.7 Rincian Bobot Nilai	Error! Bookmark not defined.
Tabel 2.8 Rincian Bobot Nilai	Error! Bookmark not defined.
Tabel 4.1 Realisasi Aturan Penomoran	Error! Bookmark not defined.
Tabel 4.2 Analisis Proses Bisnis Pendaftaran Peserta Didik Baru	Error! Bookmark not defined.
Tabel 4.3 Analisis Proses Bisnis Mengevaluasi Siswa	Error! Bookmark not defined.
Tabel 4.4 Kodifikasi Aktivitas Proses Bisnis	Error! Bookmark not defined.
Tabel 4.5 Identifikasi Stakeholder	Error! Bookmark not defined.
Tabel 4.6 Identifikasi Fitur	Error! Bookmark not defined.
Tabel 4.7 Kebutuhan Fungsional	Error! Bookmark not defined.
Tabel 4.8 Kebutuhan Non-Fungsional	Error! Bookmark not defined.
Tabel 4.9 Identifikasi <i>Use Case</i>	Error! Bookmark not defined.
Tabel 4.10 <i>Use Case Scenario</i> Mengisi Formulir Pendaftaran Peserta Didik Baru	Error! Bookmark not defined.
Tabel 4.11 <i>Use Case Scenario</i> Melakukan Verifikasi Pendaftaran	Error! Bookmark not defined.
Tabel 4.12 <i>Use Case Scenario</i> Memasukkan Detail Nilai	Error! Bookmark not defined.
Tabel 4.13 <i>Use Case Scenario</i> Memasukkan Detail Nilai Sikap	Error! Bookmark not defined.
Tabel 4.14 <i>Use Case Scenario</i> Melihat Detail Nilai....	Error! Bookmark not defined.
Tabel 4.15 <i>Use Case Scenario</i> Autentikasi	Error! Bookmark not defined.
Tabel 4.16 <i>Use Case Scenario</i> Keluar	Error! Bookmark not defined.
Tabel 6.1 Spesifikasi Perangkat Keras	Error! Bookmark not defined.
Tabel 6.2 Spesifikasi Perangkat Lunak	Error! Bookmark not defined.

Tabel 6.3 Implementasi Algoritma Mengisi Formulir Pendaftaran Didik Baru **Error! Bookmark not defined.**

Tabel 6.4 Implementasi Algoritme Melakukan Verifikasi Pendaftaran**Error! Bookmark not defined.**

Tabel 6.5 Implementasi Algoritme Memasukkan Detail Nilai**Error! Bookmark not defined.**

Tabel 6.6 Implementasi Algoritme Melihat Detail Nilai**Error! Bookmark not defined.**

Tabel 7.1 Skenario *Validation Testing* Mengisi Formulir Pendaftaran Peserta Didik Baru**Error! Bookmark not defined.**

Tabel 7.2 Skenario *Validation Testing* Mengisi Formulir Pendaftaran Peserta Didik Baru**Error! Bookmark not defined.**

Tabel 7.3 Skenario *Validation Testing* Mengisi Formulir Pendaftaran Peserta Didik Baru**Error! Bookmark not defined.**

Tabel 7.4 Skenario *Validation Testing* Memasukkan Detail Nilai**Error! Bookmark not defined.**

Tabel 7.5 Skenario *Validation Testing* Memasukkan Detail Nilai**Error! Bookmark not defined.**

Tabel 7.6 Skenario *Validation Testing* Memasukkan Detail Nilai Yang Sudah Melewati Batas Waktu Input Nilai**Error! Bookmark not defined.**

Tabel 7.7 Perancangan pengujian UAT untuk aktor admin**Error! Bookmark not defined.**

Tabel 7.8 Perancangan pengujian UAT untuk aktor guru**Error! Bookmark not defined.**

Tabel 7.9 Perancangan pengujian UAT untuk aktor calon siswa**Error! Bookmark not defined.**

Tabel 7.10 Perancangan *Acceptance Test Plan (ATP)***Error! Bookmark not defined.**

Tabel 7.11 Hasil pengujian UAT terhadap aktor admin**Error! Bookmark not defined.**

Tabel 7.12 Rekapitulasi hasil UAT terhadap aktor admin**Error! Bookmark not defined.**

Tabel 7.13 Hasil pengujian UAT terhadap aktor guru**Error! Bookmark not defined.**

Tabel 7.14 Rekapitulasi hasil UAT terhadap aktor admin**Error! Bookmark not defined.**

Tabel 7.15 Hasil pengujian UAT terhadap aktor calon siswa**Error! Bookmark not defined.**

Tabel 7.16 Rekapitulasi hasil UAT terhadap aktor calon siswa **Error! Bookmark not defined.**



DAFTAR GAMBAR

Gambar 2.1 Sistem Informasi.....	Error! Bookmark not defined.
Gambar 2.2 Klasifikasi Tipe Sistem Informasi	Error! Bookmark not defined.
Gambar 2.3 <i>Waterfall Model</i>	Error! Bookmark not defined.
Gambar 2.4 Contoh penggambaran diagram <i>Use Case</i>	Error! Bookmark not defined.
Gambar 2.5 Contoh penggambaran diagram <i>Activity</i>	Error! Bookmark not defined.
Gambar 2.6 Contoh penggambaran diagram <i>Sequence</i>	Error! Bookmark not defined.
Gambar 2.7 Contoh penggambaran diagram <i>Class</i> ..	Error! Bookmark not defined.
Gambar 2.8 Contoh <i>physical data model</i>	Error! Bookmark not defined.
Gambar 3.1 Alur Penelitian	Error! Bookmark not defined.
Gambar 4.1 Proses Bisnis Pendaftaran Peserta Didik Baru Saat Ini	Error! Bookmark not defined.
Gambar 4.2 Proses Bisnis Mengevaluasi Siswa Saat Ini	Error! Bookmark not defined.
Gambar 4.3 Proses Bisnis Pendaftaran Peserta Didik Baru Usulan	Error! Bookmark not defined.
Gambar 4.4 Proses Bisnis Mengevaluasi Siswa Usulan	Error! Bookmark not defined.
Gambar 4.5 <i>Use Case Diagram</i>	Error! Bookmark not defined.
Gambar 5.1 <i>Activity Diagram</i> Mengisi Formulir Pendaftaran Peserta Didik Baru	Error! Bookmark not defined.
Gambar 5.2 <i>Activity Diagram</i> Melakukan Verifikasi Pendaftaran	Error! Bookmark not defined.
Gambar 5.3 <i>Activity Diagram</i> Memasukkan Detail Nilai	Error! Bookmark not defined.
Gambar 5.4 <i>Sequence Diagram</i> Mengisi Formulir Pendaftaran Peserta Didik Baru	Error! Bookmark not defined.
Gambar 5.5 <i>Sequence Diagram</i> Melakukan Verifikasi Pendaftaran.....	Error! Bookmark not defined.
Gambar 5.6 <i>Sequence Diagram</i> Memasukkan Detail Nilai	Error! Bookmark not defined.
Gambar 5.7 <i>Class Diagram Controller</i>	Error! Bookmark not defined.

Gambar 5.8 *Class Diagram Model***Error! Bookmark not defined.**

Gambar 5.9 *Pyhsical Data Model*.....**Error! Bookmark not defined.**

Gambar 5.10 *Pyhsical Data Model Normalisasi*.....**Error! Bookmark not defined.**

Gambar 5.11 Antarmuka Pengguna autentikasi**Error! Bookmark not defined.**

Gambar 5.12 Antarmuka Pengguna Mengisi Pendaftaran Peserta Didik Baru**Error! Bookmark not defined.**

Gambar 5.13 Antarmuka Pengguna Melakukan Verifikasi Pendaftaran**Error! Bookmark not defined.**

Gambar 5.14 Antarmuka Pengguna Memasukkan Detail Nilai**Error! Bookmark not defined.**

Gambar 6.1 Implementasi *Class Diagram* untuk *View***Error! Bookmark not defined.**

Gambar 6.2 Struktur Artefak Sistem.....**Error! Bookmark not defined.**

Gambar 6.3 Antarmuka Pengguna Autentikasi.....**Error! Bookmark not defined.**

Gambar 6.4 Antarmuka Pengguna Mengisi Formulir Pendaftaran Peserta Didik Baru**Error! Bookmark not defined.**

Gambar 6.5 Antarmuka Pengguna Melakukan Verifikasi Pendaftaran**Error! Bookmark not defined.**

Gambar 6.6 Antarmuka Pengguna Memasukkan Detail Nilai**Error! Bookmark not defined.**

Gambar 7.1 Kategori *Compablity* Pada Aplikasi SortSite**Error! Bookmark not defined.**

Gambar 7.2 Hasil *Compability Testing***Error! Bookmark not defined.**

DAFTAR LAMPIRAN

LAMPIRAN A HASIL WAWANCARA.....**Error! Bookmark not defined.**

LAMPIRAN B HASIL PENGUJIAN *USER ACCEPTANCE TESTING* (UAT).....**Error! Bookmark not defined.**

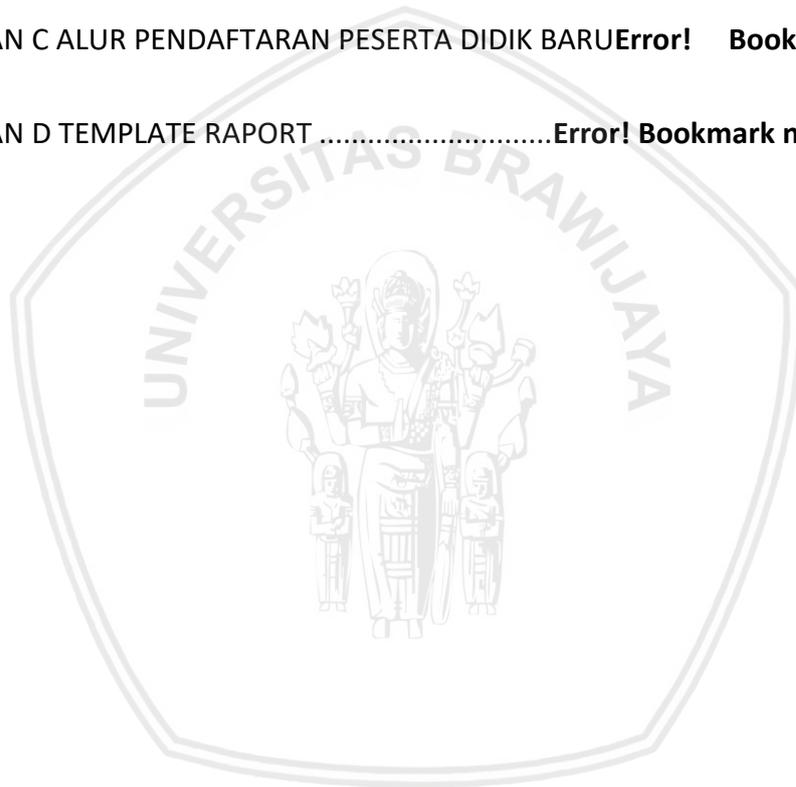
B.1 Pengujian UAT terhadap aktor admin**Error! Bookmark not defined.**

B.2 Pengujian UAT terhadap aktor guru**Error! Bookmark not defined.**

B.3 Pengujian UAT terhadap aktor calon siswa**Error! Bookmark not defined.**

LAMPIRAN C ALUR PENDAFTARAN PESERTA DIDIK BARU**Error! Bookmark not defined.**

LAMPIRAN D TEMPLATE RAPORT**Error! Bookmark not defined.**





BAB 1 PENDAHULUAN

1.1 Latar Belakang

Teknologi Informasi telah memberikan peran penting sebagai media informasi dan komunikasi dalam bidang pendidikan. Salah satu peran tersebut adalah pengelolaan akademik sekolah. Akademik adalah suatu bidang yang mempelajari tentang kurikulum atau pembelajaran dalam fungsinya untuk meningkatkan pengetahuan dalam segi pendidikan / pembelajaran dalam fungsinya untuk meningkatkan pengetahuan dalam segi pendidikan / pembelajaran yang dapat dikelola oleh suatu sekolah atau Lembaga pendidikan (Liatmaja, 2013).

Sekolah Menengah Atas Sejahtera 1 Depok merupakan salah satu instansi pendidikan di Kota Depok di bawah naungan Yayasan Keluarga Sejahtera Bandung, yang selalu berusaha untuk meningkatkan mutu baik dalam hal prestasi maupun pelayanan. Tugas pokok SMA Sejahtera 1 adalah menjalankan proses pembelajaran, evaluasi hasil belajar peserta didik, dan meluluskan peserta didik yang berkualitas memenuhi standar yang dipersyaratkan. Tugas pokok dan fungsi sekolah menurut Sagala (2010:75) diperlukan langkah – langkah sebagai berikut: (1) menyusun rencana dan pelaksanaan program sekolah didukung anggaran yang tersedia dan yang mungkin dapat disediakan;(2) mengkoordinasikan dan menyalurkan sumber daya sekolah terhadap program sekolah; (3) melaksanakan manajemen sekolah yang efektif dan efisien; (4) melaksanakan pengawasan (*supervise*) dan pembimbingan; (5) melakukan evaluasi pencapaian target; (6) menyusun laporan sekolah; dan (7) mempertanggungjawabkan penyelenggaraan sekolah secara periodik. Fungsi dan tugas utama sekolah menunjukkan bahwa perlu membangun suatu sistem yang dapat meningkatkan efektifitas kerja akademik. Untuk itu SMA Sejahtera 1 memerlukan fasilitas yang mendukung guna meningkatkan efektifitas kerja akademik dimana data akademik yang masih terpisah.

Pengelolaan akademik dapat diimplementasikan dengan menerapkan Sistem Informasi Akademik Siswa (SIAS). Sistem Informasi Akademik Siswa adalah sistem yang memberikan layanan informasi yang berupa data dalam hal yang berhubungan dengan data akademik (Setiyawan, 2013). Dengan menggunakan sistem yang terkomputerisasi, maka semua data dapat tersimpan dengan rapi, terstruktur, pengolahan data atau informasi dapat dilakukan secara cepat, tepat dan akurat.

Namun pengelolaan akademik pada SMA Sejahtera 1 Depok, belum dimanfaatkan dengan baik. Berdasarkan hasil wawancara dengan Bapak Siswanto selaku Guru BPTIK (Bimbingan dan Penyuluhan TIK) mengatakan bahwa terdapat kendala yang dihadapi, proses Pendaftaran Peserta Didik Baru masih dilakukan secara manual, yaitu dengan mengambil formulir untuk diisi, setelah itu diserahkan kembali pada pihak panitia PPDB (Pendaftaran Peserta Didik Baru), kemudian calon siswa mengikuti test / seleksi CBT (*Computer Based Test*). Berdasarkan hasil data penerimaan siswa baru pada tahun ajaran 2018-2019

menunjukkan bahwa jumlah total pendaftar 781 calon siswa yang terdiri dari 400 diterima dan 381 tidak diterima. Pada proses evaluasi siswa, distribusi penilaian untuk rapor siswa sangat kompleks, jadi harus melalui berbagai macam tahap, seperti wali kelas mengentri nilai siswa dan diberikan ke guru BPTIK, selanjutnya guru BPTIK merekap hasil nilai tersebut, Kemudian diberikan kembali ke wali kelas. Sehingga berdasarkan data tersebut diperlukannya suatu sistem informasi yang dapat membantu menangani masalah tersebut.

Dari permasalahan di atas dapat diidentifikasi bahwa terdapat dua akar utama penyebab permasalahan yang dialami adalah: Pertama, pendaftaran peserta didik baru masih dilakukan secara *manual*, sehingga menyebabkan permasalahan dalam mengelola data pendaftaran peserta didik baru. Kedua, proses evaluasi siswa, distribusi penilaian untuk rapor siswa sangat kompleks, jadi harus melalui berbagai macam tahap, seperti wali kelas mengentri nilai siswa dan diberikan ke guru BPTIK, selanjutnya guru BPTIK merekap hasil nilai tersebut, Kemudian diberikan kembali ke wali kelas. Pihak SMA Sejahtera 1 Depok membutuhkan pengembangan sistem informasi yang terstruktur, dan mengakomodir kebutuhan. Maka penulis memilih metode *waterfall* sebagai metode pengembangan sistem informasi.

Dalam pengembangan suatu sistem informasi, hal yang pertama dilakukan adalah pendefinisian masalah, berlanjut ke analisis kebutuhan. Mengacu pada standar IEEE nomor: *IEEE-STD-1220-1998*, kebutuhan adalah sebuah pernyataan yang mengidentifikasi produk atau proses, baik secara operasional maupun fungsional. Setelah memastikan kebutuhan telah dianalisis dengan benar, tahap selanjutnya yaitu melakukan perancangan sistem. Tahap perancangan merupakan tahap yang penting dalam pengembangan suatu sistem karena kualitas hasil perancangan dapat mempengaruhi kualitas sistem. Perancangan yang mudah dalam pemeliharaan adalah perancangan dengan pendekatan *Object Oriented* (Sommerville, 2011). Setelah melalui tahap ini, hasil rancangan diwujudkan dalam bentuk sistem yang berfungsi secara utuh, menggunakan bahasa pemrograman. Bahasa pemrograman memiliki 2 orientasi yaitu terstruktur dan *Object Oriented*. Untuk sistem yang dirancang dengan pendekatan *Object Oriented* menggunakan *Object Oriented Programming*, dan diimplementasikan pada aplikasi berbasis *web* untuk kemudahan aksesnya. Terakhir, pengujian digunakan untuk memastikan sistem berjalan sesuai rancangan dan memenuhi semua kebutuhan yang telah di definisikan. Beberapa metode pengujian yang dapat dilakukan yakni *Black Box – Validation Testing* untuk memastikan sistem berjalan sesuai dengan scenario (Homes, 2012), serta *User Acceptance Testing* untuk mengetahui penerimaan sistem terhadap pengguna.

Berdasarkan uraian pernyataan tersebut, peneliti tertarik untuk melakukan penelitian tentang pengembangan sistem informasi manajemen akademik sekolah, yang nantinya akan diterapkan di SMA Sejahtera 1 Depok. Judul penelitian yang diambil oleh peneliti yakni “Pengembangan Sistem Informasi Manajemen Akademik Sekolah Menengah Atas (Studi Kasus SMA Sejahtera 1 Depok)”. Melalui penelitian ini, diharapkan pengembangan dapat

diimplementasikan guna mempermudah proses kegiatan belajar mengajar dan evaluasi studi siswa, serta tercukupinya kebutuhan masyarakat umum akan layanan pendidikan yang aktual dan informatif.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijabarkan, rumusan masalah pada penelitian ini sebagai berikut:

- a. Bagaimana hasil pemodelan proses bisnis dari Sistem Informasi Manajemen Akademik Sekolah Menengah Atas Sejahtera 1 Depok?
- b. Bagaimana hasil analisis kebutuhan Sistem Informasi Manajemen Akademik Sekolah Menengah Atas Sejahtera 1 Depok?
- c. Bagaimana hasil perancangan dan implementasi Sistem Informasi Manajemen Akademik Sekolah Menengah Atas Sejahtera 1 Depok?
- d. Bagaimana hasil pengujian Sistem Informasi Manajemen Akademik Sekolah Menengah Atas Sejahtera 1 Depok terkait fungsionalitas, non fungsionalitas dan tingkat penerimaan calon siswa dalam pendaftaran peserta didik baru, admin dalam memverifikasi pendaftaran calon siswa, dan guru dalam mengevaluasi siswa?

1.3 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini ialah:

- a. Memodelkan proses bisnis pendaftaran peserta didik baru dan mengevaluasi siswa Sekolah Menengah Atas Sejahtera 1 Depok.
- b. Menganalisis dan menyusun spesifikasi kebutuhan Sistem Informasi Manajemen Akademik Sekolah Menengah Atas Sejahtera 1 Depok.
- c. Merancang dan mengimplementasikan rancangan Sistem Informasi Manajemen Akademik Sekolah Menengah Atas Sejahtera 1 Depok.
- d. Menguji terkait kebutuhan fungsionalitas, non fungsionalitas dan tingkat penerimaan admin, calon siswa dan guru terhadap Sistem Informasi Manajemen Akademik Sekolah Menengah Atas Sejahtera 1 Depok.

1.4 Manfaat

Manfaat yang diharapkan didapat dari penelitian adalah sebagai berikut:

1. Bagi SMA Sejahtera 1 Depok:

- a. Membantu mewujudkan Sistem Informasi Akademik Sekolah Menengah Atas.
- b. Memperbaiki dan memudahkan pengolahan data akademik di SMA Sejahtera 1 Depok.
- c. Membantu mewujudkan Visi Misi SMA Sejahtera 1 Depok.

- d. Menjadikan SMA yang unggul dibidang *Information and Communication Tecnology*

2. Bagi Calon Siswa dan Guru sebagai pengguna sistem:

- a. Mempermudah proses pendaftaran dan verifikasi pendaftaran siswa.
- b. Mempermudah proses penyampaian nilai (rapor) dengan menggunakan Sistem Informasi Akademik Sekolah Menengah Atas Sejahtera 1 Depok.

1.5 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

- a. Penelitian ini berfokus pada Pendaftaran Peserta Didik Baru dan evaluasi siswa pada Sekolah Menengah Atas Sejahtera 1 Depok.
- b. Penelitian ini mencakup pada fase *Communication, Planning, Modeling, dan Construction*. Akan tetapi, tidak sampai *Deployment* pada metode *waterfall*.

1.6 Sistematika Pembahasan

Sistematika bahasan dalam penyusunan skripsi ini ialah sebagai berikut:

BAB 1 PENDAHULUAN

Bab ini berisi penjelasan tentang latar belakang masalah yang ada, rumusan masalah, tujuan dari penelitian, manfaat yang diharapkan didapat, batasan masalah serta sistematika penulisan yang digunakan.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini menjelaskan tentang kajian pustaka dan dasar teori yang mendasari Pengembangan Sistem Informasi Akademik Sekolah Menengah Atas (Studi Kasus Pada SMA Sejahtera 1 Depok).

BAB 3 METODOLOGI PENELITIAN

Pada bab ini, menjelaskan tentang langkah – langkah yang di tempuh atau metode penelitian yang digunakan dalam melakukan penelitian ini. Metode yang digunakan memuat pengambilan data, pengembangan sistem, pengolahan data serta solusi yang akan diberikan dari hasil penelitian ini.

BAB 4 ANALISIS KEBUTUHAN

Pada bab ini menjelaskan pengidentifikasian kebutuhan apa saja baik kebutuhan fungsional maupun kebutuhan nonfungsional untuk sistem yang akan dibangun berdasarkan metodologi penelitian.

BAB 5 PERANCANGAN

Bab ini menjelaskan bagaimana memodelkan kebutuhan ke dalam rancangan berupa diagram perancangan sistem sesuai dengan metode yang digunakan. Diagram yang dijelaskan dalam bab perancangan di antaranya adalah *activity diagram*, *sequence diagram*, *class diagram*, *physical data model*, dan perancangan antarmuka pengguna.

BAB 6 IMPLEMENTASI

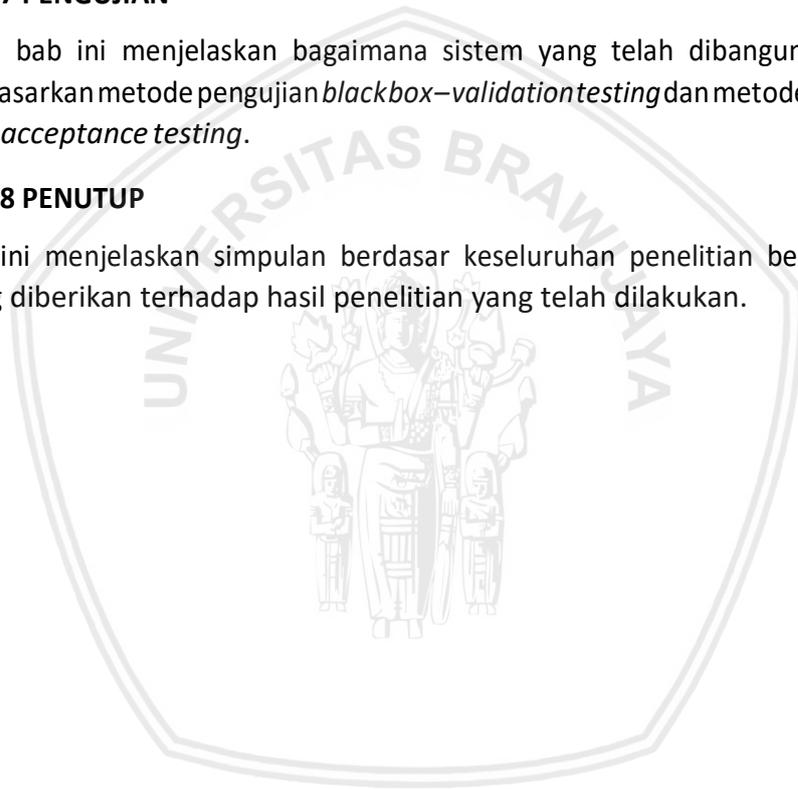
Bab ini menjelaskan mengenai spesifikasi lingkungan implementasi, beberapa hasil implementasi algoritme, dan implementasi antarmuka pengguna.

BAB 7 PENGUJIAN

Pada bab ini menjelaskan bagaimana sistem yang telah dibangun akan diuji berdasarkan metode pengujian *blackbox-validation testing* dan metode pengujian *user acceptance testing*.

BAB 8 PENUTUP

Bab ini menjelaskan simpulan berdasar keseluruhan penelitian beserta saran yang diberikan terhadap hasil penelitian yang telah dilakukan.



BAB II LANDASAN KEPUSTAKAAN

Bab ini membahas tentang dasar teori dan kajian pustaka yang akan digunakan peneliti untuk mendukung proses penelitian. Kajian pustaka berisi penelitian terkait yang sebelumnya telah dilakukan, digunakan sebagai acuan pada penelitian ini, dan dasar teori akan dijadikan pedoman untuk melakukan penelitian.

2.1 Kajian Pustaka

Isi dari kajian pustaka berupa beberapa penelitian terkait yang telah dilakukan sebelumnya. Penelitian terkait pertama yakni berjudul “Analisis dan Perancangan Sistem Informasi Sekolah Berbasis Web” membahas mengenai penggunaan sistem informasi untuk menangani permasalahan dalam bidang akademik (Prasetyo, 2015). Beberapa masalah yang melatarbelakangi penelitian tersebut ialah pembelajaran di kelas belum memaksimalkan fungsi teknologi informasi khususnya internet, sistem informasi yang sudah ada masih sebatas web statis dan kesulitan dalam melakukan proses *update* informasi, pengolahan informasi masih belum menggunakan sistem informasi terpadu yang dinamis, dan penyebaran informasi yang kurang efektif karena masih menggunakan surat menyurat untuk komunikasi dengan siswa maupun orang tua. Dalam penelitian tersebut, sistem informasi dibangun dengan menggunakan model model *ADDIE (Analysis-Design-Develop-Implement-Evaluate)* memodelkan proses menggunakan *Unified Modelling Language (UML)*, menggunakan MySQL DBMS untuk pengelolaan data, dan menggunakan PHP & Javascript sebagai bahasa pemrograman web. Keterkaitan penelitian tersebut dengan penelitian yang dilakukan penulis ialah dalam topik yang sama, yakni topik sistem informasi manajemen akademik.

Penelitian terkait kedua memiliki judul “Pengembangan Sistem Informasi Akademik Berbasis Web Sebagai Sistem Pengolahan Nilai Siswa di SMK Negeri 1 Kudus” membahas mengenai Pengembangan Sistem Informasi Akademik Berbasis Web Sebagai Sistem Pengolahan Nilai Siswa di SMK Negeri 1 Kudus (Suryandani, F., Basori & Maryono, D., 2017). Masalah yang melatarbelakangi penelitian tersebut yaitu proses penilaian evaluasi belajar siswa yang masih terbilang konvensional, jadi ketika nilai itu dibutuhkan dalam pengelolaan rapor atau penyampaian nilai kepada siswa kurang efisien, hal tersebut dirasa kurang baik dari segi penggunaan waktu dan tenaga. Penelitian tersebut menggunakan model pengembangan *waterfall*, menggunakan *Unified Modelling Language (UML)* sebagai bahasa pemodelan sistem berbasis *object oriented*. Keterkaitan penelitian tersebut adalah menggunakan orientasi yang sama dalam perancangan sistemnya, yaitu menggunakan *object oriented design*.

Berdasarkan beberapa penelitian yang terkait, maka peneliti tertarik untuk melakukan penelitian tentang pengembangan sistem informasi manajemen akademik sekolah menengah atas menggunakan metode *Waterfall*, menggunakan *Unified Modelling Language* sebagai bahasa pemodelan sistemnya, dan mengimplementasikan sistem dalam bentuk aplikasi berbasis *web*.

Tabel 2.1 Rangkuman Kajian Pustaka

No.	Nama Peneliti, Nama Jurnal, dan Tahun	Tujuan	Metode Penelitian / Pengembangan	Hasil dan Kesimpulan
1.	<p>Sapto Prasetyo</p> <p><i>Analisis dan Perancangan Sistem Informasi Sekolah Berbasis Web.2015</i></p> <p>(Diambil dari <i>eprints@UNY Lumbung Pustaka Universitas Negeri Yogyakarta (UNY)</i>)</p>	<p>Penelitian ini mengenai penggunaan sistem informasi untuk menangani permasalahan dalam bidang akademik</p>	<p>model <i>ADDIE (Analysis-Design-Develop-Implement-Evaluate)</i> memodelkan proses menggunakan <i>Unified Modelling Language (UML)</i>, menggunakan MySQL DBMS untuk pengelolaan data, dan menggunakan PHP & Javascript sebagai bahasa pemrograman web.</p>	<p>Pengembangan sistem informasi sekolah dengan model <i>ADDIE</i>. Dalam tahapan tersebut dijabarkan, 1. Analisis kebutuhan sistem, 2. Desain sistem dan perangkat lunak, 3. Pengembangan tampilan dan fungsionalitas, 4. Implementasi dan integrasi perangkat lunak, dan 5. Pengujian kelayakan perangkat lunak.</p>

Tabel 0.1 Rangkuman kajian pustaka (lanjutan)

No.	Nama Peneliti, Nama Jurnal, dan Tahun	Tujuan	Metode Penelitian	Hasil dan Kesimpulan
2.	<p>Farid Suryandani, Basori, dan Dwi Maryono</p> <p><i>Pengembangan Sistem Informasi Akademik Berbasis Web Sebagai Sistem Pengolahan Nilai Siswa di SMK Negeri 1 Kudus. 2017</i></p> <p>(Diambil dari <i>Jurnal Ilmiah Pendidikan Teknik Kejuruan (JIPTEK)</i>)</p>	<p>Penelitian ini mengenai Pengembangan Sistem Informasi Akademik Berbasis Web Sebagai Sistem Pengolahan Nilai Siswa di SMK Negeri 1 Kudus</p>	<p>model pengembangan <i>waterfall</i>, menggunakan <i>Unified Modelling Language (UML)</i> sebagai bahasa pemodelan sistem berbasis object oriented.</p>	<p>Sistem yang dikembangkan berbasis <i>web</i>, sehingga mudah untuk diakses menggunakan <i>browser</i>. Memudahkan dalam pengelolaan nilai. Akan tetapi, belum adanya integrasi dengan portal sekolah yang dapat memberikan layanan yang terintegrasi.</p>
3.	<p>Yupayao Alsammary, Chaiyuth Siristuhi, and Surat Duangchartthom. 2016</p> <p><i>Development of Educational Management System in Small Primary School</i></p> <p>(Diambil dari <i>International Education Studies</i>)</p>	<p>Penelitian ini mengenai Metode Pengembangan <i>Educational Management System</i> pada Sekolah Dasar.</p>	<p>Pengumpulan data dengan kuisisioner, wawancara, dan observasi.</p>	<p>Terdapat 3 aspek yaitu; 1. <i>Input</i> terdiri dari admin, guru, orang tua, kurikulum, keuangan, media dan teknologi, sumber belajar, dan murid. 2. <i>process</i>, terdiri dari administrasi akademik, keuangan, pribadi, dan umum. 3. <i>Output</i>, terdiri dari Kualitas Manajemen pendidikan sekolah.</p>

2.1 Sekolah Menengah Atas Sejahtera 1 Depok

Sekolah Menengah Atas merupakan jenjang pendidikan menengah yang mengutamakan penyiapan siswa untuk melanjutkan pendidikan yang lebih tinggi dengan pengkhususan. (Depdiknas, 2004: 112). Perwujudan pengkhususan tersebut berupa diselenggarakannya penjurusan di mulai di kelas XI (sebelas), yakni, penjurusan pada Ilmu Pengetahuan Alam (IPA), Ilmu Pengetahuan Sosial (IPS), dan Bahasa yang mulai diterapkan pada siswa SMA kelas XI tahun pelajaran 2005/2006.

Fungsi dari Pendidikan menengah adalah mengembangkan nilai-nilai dan sikap rasa keindahan dan harmoni, pengetahuan, kemampuan, dan keterampilan sebagai persiapan untuk melanjutkan ke pendidikan tinggi dan/atau untuk hidup di masyarakat dalam rangka mencapai tujuan pendidikan nasional. Sedangkan tujuan pendidikan menengah adalah untuk meningkatkan keimanan dan ketakwaan, hidup sehat, memperluas pengetahuan dan seni, memiliki keahlian dan keterampilan, menjadi anggota masyarakat yang bertanggung jawab, serta mempersiapkan peserta didik untuk mengikuti pendidikan lebih lanjut.

Sekolah Menengah Atas Sejahtera 1 Depok merupakan instansi pendidikan yang terletak di Jl. Anyelir Raya no. 68, Kecamatan Pancoran Mas, Kota Depok, dan dimiliki oleh Yayasan Keluarga Sejahtera Bandung. Sekolah ini didirikan pada tahun 1981, dan juga telah mendapat Akreditasi dari Badan Akreditasi Sekolah (BAS) dengan nilai A, Terakreditasi A No. 02.00/444/BAP-SM/X/2009. SMA Sejahtera 1 Depok memiliki visi, yaitu:

“Unggul dalam disiplin, cakap dan berakhlak mulia, berprestasi dengan memberdayakan potensi untuk menjadi sekolah yang berkualitas nasional.”

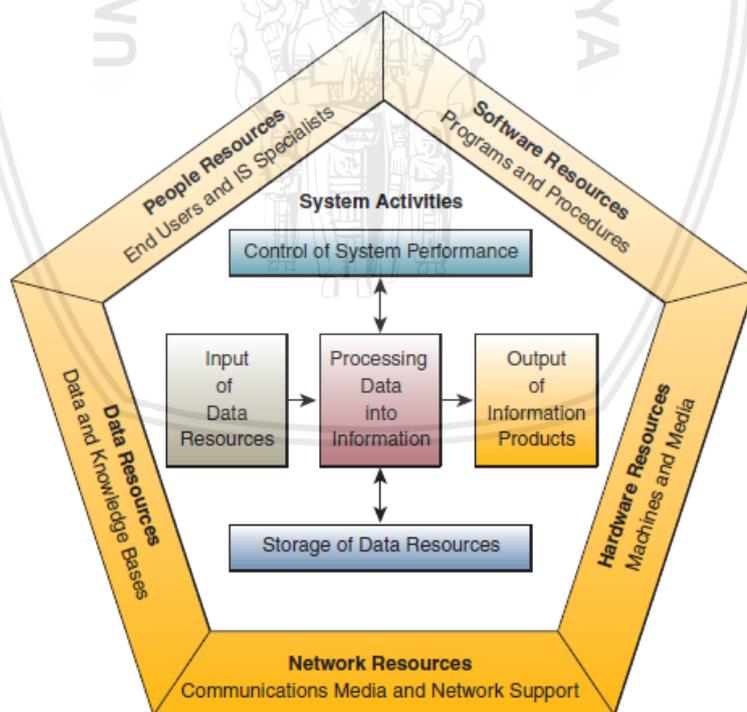
Serta memiliki Misi:

1. Menciptakan nuansa sekolah yang disiplin dan menumbuh kembangkan gairah belajar siswa dalam suasana yang demokratis.
2. Melaksanakan proses pembelajaran dengan berorientasi kecakapan hidup dengan muatan iptek dan imtaq.
3. Menumbuhkembangkan potensi siswa secara tepat.
4. Memanajemen sekolah secara professional.
5. Menanamkan sikap dan prilaku serta budi pekerti yang luhur.
6. Mengembangkan sarana dan jaringan teknologi informasi dan komunikasi untuk peningkatan mutu pembelajaran dan pelayanan administrasi.
7. Mengembangkan perpustakaan yang representatif menuju elektronik *library* sebagai sumber belajar.
8. Mengembangkan bahan ajar setiap mata pelajaran.

2.2 Sistem Informasi

Kata sistem didefinisikan sebagai seperangkat komponen yang saling terkait, memiliki batasan yang jelas, dan bekerja sama untuk mencapai tujuan. Sedangkan sistem informasi ialah kombinasi dari manusia, perangkat keras, perangkat lunak, sumber data, jaringan komunikasi, kebijakan dan prosedur yang mengelola dan menyebarkan informasi dalam suatu organisasi. Sistem informasi digunakan oleh manusia untuk berkomunikasi satu sama lain menggunakan hardware sebagai perangkat fisik, software untuk pemrosesan informasi dan prosedur, jaringan sebagai saluran komunikasi dan data yang tersimpan sebagai sumber data (O'Brien & Marakas, 2013).

Sistem informasi merupakan suatu sistem yang menerima data sebagai input, memproses data sehingga menghasilkan output berupa informasi yang bernilai bagi pengguna, yang melibatkan berbagai komponen untuk mencapai tujuan. Gambar 2.1 merupakan model sistem informasi yang mengungkapkan kerangka konseptual mendasar untuk komponen dan aktivitas utama dalam sistem informasi. Semua sistem informasi melibatkan manusia, *hardware*, *software*, data, dan jaringan untuk melakukan input, proses, output, penyimpanan dan kontrol dengan tujuan mengubah sumber data menjadi sebuah informasi (O'Brien & Marakas, 2013).



Gambar 2.1 Sistem Informasi

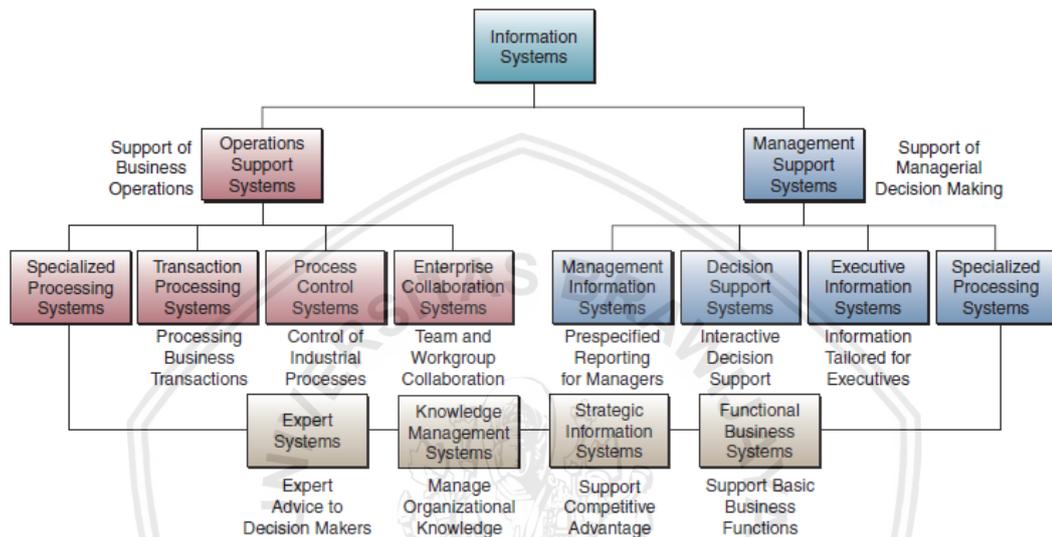
Sumber: O'Brien & Marakas (2013)

Penerapan sistem informasi dapat meningkatkan efisiensi suatu bisnis karena dengan adanya sistem informasi, informasi dapat diakses secara cepat, tepat, akurat, dan dapat dimanfaatkan untuk mencapai tujuan bisnis. Selain itu,

kemudahan akses informasi juga dapat digunakan untuk membantu pengambilan keputusan.

2.2.1 Sistem Informasi Manajemen

Sistem Informasi Manajemen merupakan salah satu klasifikasi dari sistem informasi yang memberikan informasi berupa laporan dan *display* kepada para manajer dan profesional bisnis, dan merupakan bentuk sistem informasi yang paling umum dalam sebuah organisasi (O'Brien & Marakas, 2010).



Gambar 2.2 Klasifikasi Tipe Sistem Informasi

Sumber: O'Brien & Marakas (2013)

Gambar 2.2 mengklasifikasikan konseptual model dari sistem informasi berdasarkan tujuannya. Sistem informasi memiliki dua klasifikasi yaitu sebagai *operations support system* dan *management support system*. *Management support systems* berfokus pada penyediaan informasi dan dukungan untuk mengambil keputusan yang efektif oleh manajer dan profesional bisnis lainnya. Contohnya, manajer bagian penjualan dapat menggunakan komputer yang terhubung dengan internet untuk mengakses laporan hasil penjualan produk secara *real-time* pada *web browser* dan dapat melakukan analisis penjualan harian untuk kebutuhan evaluasi (O'Brien & Marakas, 2013).

2.3 Rekayasa Perangkat Lunak

Perangkat lunak (*Software*) merupakan program komputer beserta dokumentasi yang menyertainya. Rekayasa perangkat lunak adalah disiplin ilmu yang berfokus pada segala aspek produksi perangkat lunak dari awal spesifikasi sistem hingga sistem jadi dan siap untuk digunakan. Rekayasa perangkat lunak merupakan bagian dari rekayasa sistem. Segala aspek dalam produksi perangkat lunak bukan hanya berfokus pada teknik proses pengembangan perangkat lunak, namun juga termasuk aktivitas seperti manajemen proyek dan *tools*



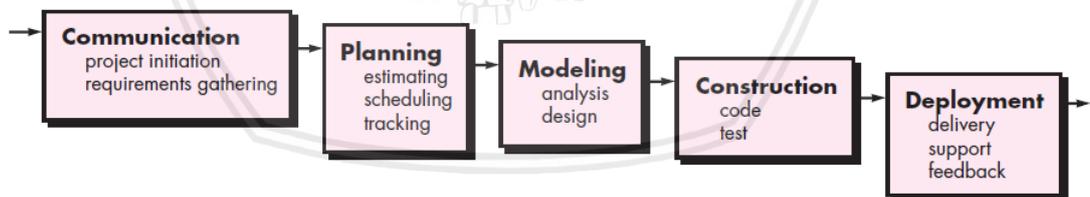
pengembangannya, metode, dan teori yang mendukung pengembangan perangkat lunak (Sommerville, 2011).

2.4 Software Development Life Cycle

Software Development Life Cycle (SDLC) adalah metode pengembangan sistem tradisional yang terstruktur dan yang digunakan sebagian besar organisasi saat ini. SDLC adalah kerangka kerja (*framework*) yang terstruktur yang berisi proses-proses sekuensial dimana sistem informasi dikembangkan (Turban, Rainer & Potter, 2010). SDLC menyediakan urutan kegiatan dalam merancang perangkat lunak yang dapat memungkinkan perusahaan untuk mengatur jadwal kerja secara efisien untuk membangun perangkat lunak dengan kerangka waktu dan anggaran tertentu (Mishra & Dubey, 2013). Dalam SDLC terdapat beberapa teknik pengembangan sistem yang umum digunakan, dalam penelitian ini akan menggunakan teknik *Waterfall Model*.

2.5 Waterfall Model

Waterfall Model adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Nama model ini sebenarnya adalah "*Linear Sequential Model*". Model ini sering disebut juga dengan "*classic life cycle*" atau metode waterfall. Model ini termasuk ke dalam model *generic* pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE). model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan (Pressman, 2010).



Gambar 2.3 Waterfall Model

Sumber: Pressman (2010)

2.5.1 Communication (Project Initiation & Requirements Gathering)

Sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan *customer* demi memahami dan mencapai tujuan yang ingin dicapai. Hasil dari komunikasi tersebut adalah inisialisasi proyek, seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data – data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi *software*. Pengumpulan data – data tambahan bisa juga diambil dari jurnal, artikel, dan internet.

2.5.2 Planning (Estimating, Scheduling, Tracking)

Tahap berikutnya adalah tahapan perencanaan yang menjelaskan tentang estimasi tugas – tugas teknis yang akan dilakukan, resiko – resiko yang dapat terjadi, sumber daya yang diperlukan dalam membuat sistem, produk kerja yang akan dihasilkan, penjadwalan kerja yang akan dilaksanakan, dan *tracking* proses pengerjaan sistem

2.5.3 Modeling (Analysis & Design)

Tahapan ini adalah tahap perancangan dan permodelan arsitektur sistem yang berfokus pada perancangan struktur data, arsitektur *software*, tampilan *interface*, dan algoritma program. Tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan.

2.5.4 Construction (Code & Test)

Tahapan *Construction* ini merupakan proses penerjemahan bentuk desain menjadi kode atau bentuk / bahasa yang dapat dibaca oleh mesin. Setelah pengkodean selesai, dilakukan pengujian terhadap sistem dan juga kode yang sudah dibuat. Tujuannya untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki.

2.5.5 Deployment (Delivery, Support, Feedback)

Tahapan *Deployment* merupakan tahapan implementasi *software* ke *customer*, pemeliharaan *software* secara berkala, perbaikan *software*, evaluasi *software*, pengembangan *software* berdasarkan *feedback* yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya, dan formal rilis yang diberikan ke *customer*, sehingga dapat dioperasikan secara utuh, dan dapat diluncurkan ke publik (Pressman, 2015:17).

Keuntungan menggunakan metode *waterfall* adalah prosesnya lebih terstruktur, hal ini membuat kualitas *software* baik dan tetap terjaga. Dari sisi *user* juga lebih menguntungkan, karena dapat merencanakan dan menyiapkan kebutuhan data dan proses yang diperlukan sejak awal. Penjadwalan juga menjadi lebih menentu, karena jadwal setiap proses dapat ditentukan secara pasti. Sehingga dapat dilihat jelas target penyelesaian pengembangan program. Dengan adanya urutan yang pasti, dapat dilihat pula perkembangan untuk setiap tahap secara pasti. Dari sisi lain, model ini merupakan jenis model yang bersifat dokumen lengkap sehingga proses pemeliharaan dapat dilakukan dengan mudah.

Kelemahan menggunakan metode *waterfall* adalah bersifat kaku, sehingga sulit melakukan perubahan di tengah proses. Jika terdapat kekurangan proses/prosedur dari tahap sebelumnya, maka tahapan pengembangan harus dilakukan mulai dari awal lagi. Hal ini akan memakan waktu yang lebih lama. Karena jika proses sebelumnya belum selesai sampai akhir, maka proses selanjutnya juga tidak dapat berjalan. Oleh karena itu, jika terdapat kekurangan dalam permintaan *user* maka proses pengembangan harus dimulai kembali dari

awal. Karena itu, dapat dikatakan proses pengembangan *software* dengan metode *waterfall* bersifat lambat.

Kelemahan lainnya menggunakan metode *waterfall* adalah membutuhkan daftar kebutuhan yang lengkap sejak awal. Tetapi, biasanya jarang sekali *customer* yang dapat memenuhi itu. Untuk menghindari pengulangan tahap dari awal, *user* harus memberikan seluruh prosedur, data, dan laporan yang diinginkan mulai dari tahap awal pengembangan. Tetapi pada banyak kondisi, *user* sering melakukan permintaan di tahap pertengahan pengembangan sistem. Dengan metode ini, maka *development* harus dilakukan mulai lagi dari tahap awal. Karena *development* disesuaikan dengan desain hasil *user* pada saat tahap pengembangan awal. Di sisi lain, *user* tidak dapat mencoba sistem sebelum sistem benar-benar selesai. Selain itu, kinerja personil menjadi kurang optimal karena terdapat proses menunggu suatu tahap selesai terlebih dahulu. Oleh karena itu, seringkali diperlukan personil yang "*multi-skilled*" sehingga minimal dapat membantu pengerjaan untuk tahapan berikutnya. (Pressman, 2015:42-43)

2.6 Requirement Gathering

Requirement Gathering atau Analisis kebutuhan sistem merupakan proses untuk membangun layanan atau produk yang dibutuhkan oleh pelanggan atau pengguna beserta batasan-batasan yang diberlakukan ketika produk atau layanan tersebut digunakan dan beroperasi, serta ketika produk atau layanan tersebut dibangun atau dikembangkan. Kebutuhan itu sendiri mengacu pada standar *IEEE* nomor: *IEEE-STD-1220-1998* menyatakan bahwa kebutuhan adalah sebuah pernyataan yang mengidentifikasi produk atau proses, baik secara operasional maupun fungsional, atau diperlukan batasan atau karakteristik dari desain, yang tidak ambigu, dapat diuji, atau terukur, dan agar produk dan prosesnya dapat diterima oleh konsumen atau sesuai dengan petunjuk penjaminan mutu secara internal. Kebutuhan sistem tersebut terdiri dari beberapa *stakeholder* yaitu, Admin, Calon Siswa, Siswa, dan Guru.

Kebutuhan pengguna secara umum terbagi menjadi 2 kategori utama, yaitu kebutuhan fungsional dan kebutuhan nonfungsional. Kebutuhan fungsional merepresentasikan apa yang sistem lakukan, sedangkan kebutuhan nonfungsional merepresentasikan batasan apa saja yang harus dipenuhi oleh sistem dalam hal batasan kualitas dan batasan penggunaan (*usability*).

Menurut Pressman (2010), kebutuhan memiliki 3 tingkatan yakni:

1. Kebutuhan normal: yaitu kebutuhan yang harus dipenuhi dan dinyatakan secara eksplisit oleh pengguna/*stakeholder* seperti fungsionalitas sistem dan performa sistem.
2. Kebutuhan yang diharapkan (*expected*): yaitu kebutuhan yang tidak dinyatakan secara eksplisit namun menentukan kepuasan pelanggan, seperti: kemudahan interaksi dengan sistem, akurasi dan kebenaran proses.

3. Kebutuhan yang mengejutkan (*exciting*): yaitu kebutuhan yang melebihi kebutuhan normal untuk dapat lebih memuaskan pelanggan, seperti fungsionalitas tambahan sistem lainnya yang lebih memuaskan pelanggan ketika menggunakan sistem dalam menyelesaikan masalah, mencapai tujuan pengguna, dengan lebih efisien.

Hasil dari proses analisis kebutuhan sistem diantaranya sebagai bentuk kesepakatan antara pengembang dengan pengguna akan kebutuhan yang harus dipenuhi, menyediakan dasar yang akurat bagi proses perancangan sistem, menjadi referensi bagi proses validasi untuk memastikan bahwa seluruh kebutuhan telah dispesifikasikan dengan benar, lengkap dan konsisten.

2.7 Proses Bisnis

Proses bisnis merupakan sekumpulan aktivitas yang memerlukan satu atau lebih jenis *input* dan membuat *output* yang bernilai bagi pelanggan. Proses bisnis terdiri dari sekumpulan aktivitas yang dilakukan secara koordinasi untuk mewujudkan tujuan bisnis. Setiap proses bisnis diberlakukan oleh satu organisasi, tapi memungkinkan berinteraksi dengan proses bisnis yang dilakukan oleh organisasi lain (Weske, 2007).

Berikut ini adalah jenis aktivitas dalam proses bisnis (Weske, 2007):

1. Aktivitas *manual* adalah aktivitas yang tidak didukung oleh sistem informasi. Contoh dari aktivitas manual adalah mengirimkan *parcel* kepada rekan bisnis.
2. Aktivitas interaksi pengguna adalah aktivitas yang dilakukan oleh karyawan menggunakan sistem informasi. Aktivitas ini perlu didukung antarmuka sistem yang baik agar pekerjaan yang dilakukan dapat lebih efektif. Contoh dari aktivitas interaksi pengguna adalah memasukkan data asuransi menggunakan sistem.
3. Aktivitas sistem adalah aktivitas yang tidak melibatkan peran pengguna. Aktivitas sepenuhnya dilakukan oleh sistem. Contoh dari aktivitas sistem adalah mengecek saldo rekening bank.

Proses bisnis direpresentasikan dengan model proses bisnis. Model proses bisnis diekspresikan dalam sebuah diagram proses bisnis. Diagram proses bisnis merupakan diagram yang merepresentasikan aktivitas-aktivitas dan tugas-tugas dari suatu proses dan hubungan antar keduanya. Diagram proses bisnis digambarkan menggunakan notasi grafik tertentu. Contoh notasi yang dapat digunakan untuk memodelkan proses bisnis adalah UML, *Activity Diagram* dan BPMN (*Business Process Model and Notation*) (Weske, 2007).

2.7.1.1 Analisis Pemodelan Proses Bisnis *As-Is*

Analisis dan pemodelan proses bisnis *as-is* merupakan kegiatan untuk mengumpulkan dan memodelkan proses bisnis yang actual pada organisasi. Hal ini dilakukan untuk menyajikan informasi mengenai situasi dan proses bisnis actual perusahaan atau organisasi, mengetahui kekurangan dari proses bisnis dan

memunculkan peningkatan dalam proses bisnis yang sudah ada (Becker, et al., 2013)

Proses bisnis *as-is* dapat diidentifikasi dengan melakukan wawancara dengan pegawai yang berpartisipasi dalam eksekusi proses bisnis. Selain itu proses bisnis *as-is* dapat diidentifikasi dengan melakukan observasi terhadap pegawai yang sedang mengerjakan tugas-tugasnya dalam organisasi atau perusahaan (Przybylek, 2011).

2.7.1.2 Analisis Pemodelan Proses Bisnis *To-Be*

Analisis dan pemodelan proses bisnis *to-be* merupakan kegiatan untuk mengidentifikasi dan memodelkan proses bisnis yang sudah mengalami peningkatan (*improvement*) berdasarkan analisis terhadap proses bisnis *as-is*. Penerapan proses bisnis *to-be* dapat dilakukan dalam jangka waktu menengah atau jangka panjang (Becker, et al., 2013).

Analisis dan pemodelan proses bisnis *to-be* dilakukan dengan melibatkan pemangku kepentingan (*stakeholder*) untuk dapat memperoleh kemungkinan peningkatan proses bisnis pada organisasi atau perusahaan. Peningkatan proses bisnis mungkin dilakukan dengan menerapkan otomasi pada beberapa proses bisnis. Namun, otomasi tidak menjamin peningkatan yang signifikan, sehingga perlu melibatkan pemangku kepentingan (*stakeholder*) dalam kegiatan analisis dan pemodelan proses bisnis *to-be* (Przybylek, 2011)

2.7.2 *Business Process Model and Notation* (BPMN)

Business Process Model and Notation (BPMN) merupakan notasi grafis yang digunakan untuk memodelkan suatu proses bisnis. Tujuan utama dari BPMN ialah untuk memberikan notasi yang mudah dipahami oleh semua *business users*, mulai dari analis bisnis yang membuat konsep awal dari proses - proses, pengembang teknis yang bertanggung jawab dalam implementasi teknologi yang akan melakukan proses - proses tersebut, dan pelaku bisnis yang akan mengelola dan memantau proses - proses tersebut (Weske, 2007).

BPMN memiliki empat kategori elemen dasar yaitu (Weske, 2007):

1. *Flow object*.
Flow object adalah elemen utama yang menentukan perilaku proses bisnis. *Flow Object* terdiri dari *events*, aktivitas, *gateways*.
2. *Artefacts*.
Artefacts digunakan untuk menampilkan informasi tambahan tentang proses bisnis. *Artefacts* terdiri dari *data object*, *groups*, dan *text annotations*. Setiap *artefacts* dapat dikaitkan dengan *flow elements*.
3. *Connecting object*.
Connecting object menghubungkan *flow object*, *swimlanes*, atau *artefacts*. *Connecting object* terdiri dari *sequence flows*, *message flows*, *association*, dan *data associations*.

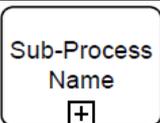
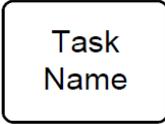
4. *Swimlanes*.

Swimlanes terdiri dari *pools* dan *lanes*. *Pools* merepresentasikan organisasi yang berpartisipasi dalam interaksi beberapa proses bisnis, masing-masing *pools* dilakukan oleh satu organisasi. *Lanes* merepresentasikan entitas organisasi, seperti departemen yang ada dalam organisasi tersebut.

Tabel 2.2 Notasi BPMN

Nama	Notasi	Deskripsi
<i>Start Event</i>		Digunakan untuk memulai proses
<i>End Events</i>		Digunakan untuk mengakhiri proses.
<i>Intermediate Events</i>		Menunjukkan suatu kejadian (<i>Event</i>) yang ada di antara awal dan akhir proses. <i>Intermediate event</i> hanya mempengaruhi aliran proses
<i>Activity</i>		Notasi ini merupakan kegiatan / aktivitas yang dilakukan perusahaan dalam sebuah proses.

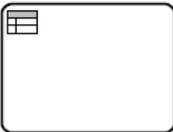
Tabel 2.2 Notasi BPMN (lanjutan)

Nama	Notasi	Deskripsi
<i>Collapsed Sub-Process</i>		Notasi ini merupakan aktivitas dimana didalam aktivitas tersebut terdapat proses yang terjadi.
<i>Task</i>		<i>Task</i> adalah bentuk <i>atomic</i> dari <i>Activity</i> . <i>Task</i> digunakan ketika pekerjaan dalam proses



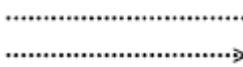
		tidak dapat dipecah lagi ke tingkat yang lebih detail.
<i>Service Task</i>		Task yang menggunakan layanan otomatis, dapat berupa <i>web service</i> atau aplikasi.
<i>Send Task</i>		Task ini digunakan untuk mengirim pesan pada pihak eksternal dan selesai ketika pesan telah dikirim.
<i>Receive Task</i>		Task ini digunakan untuk menunggu dan menerima pesan dari pihak eksternal. Task ini selesai ketika pesan telah diterima.
<i>User Task</i>		Task ini menunjukkan bahwa tugas yang dilakukan melibatkan interaksi antara manusia dengan aplikasi.
<i>Manual Task</i>		Task ini menunjukkan bahwa tugas dilakukan tanpa bantuan alat proses bisnis atau aplikasi.

Tabel 2.2 Notasi BPMN (lanjutan)

Nama	Notasi	Deskripsi
<i>Business Rule Task</i>		Task ini menyediakan mekanisme proses untuk memberikan masukan ke <i>Business Rule Engine</i> dan mendapatkan <i>output</i> yang diberikan oleh <i>Business Rule Engine</i> .

<i>Script Task</i>		<i>Task</i> yang menunjukkan tugas dilakukan oleh sebuah mesin proses bisnis dengan menjalankan suatu <i>script</i> .
<i>Sequence Flow</i>		Digunakan untuk menunjukkan urutan <i>Flow Elements</i> dalam sebuah proses dan hanya memiliki satu sumber dan satu target. <i>Flow Elements</i> terdiri dari: <i>Events</i> (<i>Start</i> , <i>Intermediate</i> , dan <i>End</i>), <i>Aktivitas</i> (<i>Task</i> dan <i>Sub-Process</i>), <i>Choreography Activities</i> (<i>Choreography Task</i> dan <i>Sub-Choreography</i>); untuk <i>Choreographies</i>), dan <i>Gateways</i> .
<i>Conditional Flow</i>		<i>Sequence flow</i> yang digunakan ketika ada sebuah kondisi, aliran akan diteruskan hanya jika kondisi bernilai <i>true</i> .
<i>Default Flow</i>		Notasi ini digunakan hanya ketika semua <i>conditional flow</i> bernilai <i>false/not true</i> pada saat runtime.

Tabel 2.2 Notasi *BPMN* (lanjutan)

Nama	Notasi	Deskripsi
<i>Message Flow</i>		Digunakan untuk menunjukkan aliran pengiriman pesan antara dua partisipan.
<i>Association / Asosiasi</i>		Notasi yang digunakan untuk menghubungkan

		informasi dan <i>artifact</i> dengan <i>flow object</i> .
<i>Gateway</i>		<i>Gateway</i> ini menunjukkan pemecahan (karna adanya kondisi) atau penggabungan <i>sequence flow</i> dalam sebuah proses.
<i>Gateway Control Types</i>	<p>Exclusive  or </p> <p>Event-Based  </p> <p>Parallel Event-Based </p> <p>Inclusive </p> <p>Complex </p> <p>Parallel </p>	<p><i>Exclusive</i> (Keputusan): digunakan untuk membuat jalur alternatif dalam sebuah aliran proses.</p> <p><i>Event-based</i></p> <p><i>Inclusive</i>: digunakan untuk membuat jalur alternatif atau paralel dalam sebuah aliran proses, semua ekspresi kondisi dievaluasi.</p> <p><i>Complex</i>: digunakan untuk sinkronisasi jalur kompleks.</p> <p><i>Parallel</i>: digunakan untuk menggabungkan dan membuat aliran paralel.</p>

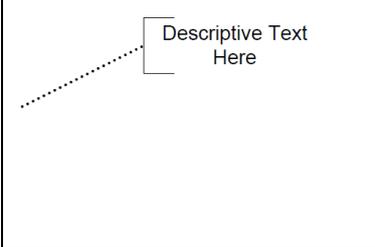
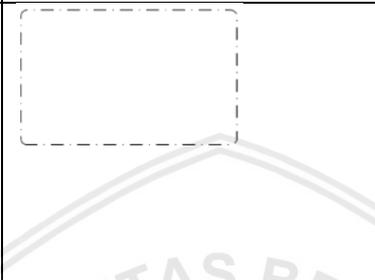
Tabel 2.2 Notasi BPMN (lanjutan)

Nama	Notasi	Deskripsi
<i>Pool</i>		<i>Pool</i> adalah representasi grafis dari partisipan. Partisipan dapat berupa sebuah perusahaan/organisasi atau mewakili peran yang lebih umum (misalnya,

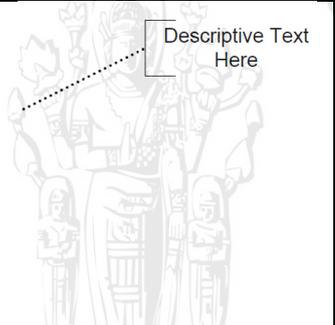
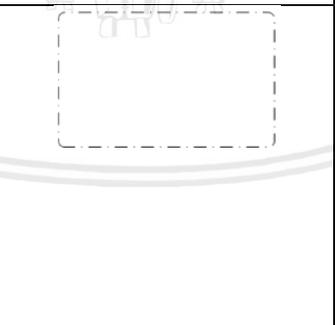
		pembeli, penjual, atau produsen).
<i>Message</i>		Digunakan untuk menunjukkan isi dari komunikasi yang dilakukan antar dua partisipan.
<i>Lane</i>		<i>Lane</i> adalah sebuah bagian dalam proses atau <i>pool</i> yang panjangnya sama dengan panjang proses baik secara vertikal maupun horizontal. <i>Lane</i> digunakan untuk mengatur dan mengkategorikan aktivitas dalam sebuah <i>pool</i> . <i>Lane</i> terkadang digunakan untuk mewakili peran internal perusahaan

Tabel 2.2 Notasi BPMN (lanjutan)

Nama	Notasi	Deskripsi
<i>Data object</i>	<p>Data Object</p>  <p>Data Objec (Collection)</p>  <p>Data Input Data Output</p> 	<p><i>Data Object</i>: menunjukkan sebuah informasi yang ada pada sebuah proses seperti surat, dokumen bisnis, <i>e-mail</i>.</p> <p><i>Collection</i>: menunjukkan sekumpulan informasi.</p> <p><i>Data Input-Output</i>: <i>Data input</i> adalah masukan untuk seluruh proses, <i>data ouput</i> adalah data hasil dari seluruh proses.</p>
<i>Group</i>		Digunakan untuk mengelompokkan elemen grafis yang mempunyai kategori yang sama. Nama kategori muncul dalam

		diagram sebagai label grup.
<i>Text Annotation</i>		Notasi ini digunakan untuk memberikan informasi teks tambahan dalam sebuah proses. <i>Text Annotation</i> dihubungkan menggunakan Asosiasi.
<i>Group</i>		Digunakan untuk mengelompokkan elemen grafis yang mempunyai kategori yang sama. Nama kategori muncul dalam diagram sebagai label grup.

Tabel 2.2 Notasi BPMN (lanjutan)

<i>Text Annotation</i>		Notasi ini digunakan untuk memberikan informasi teks tambahan dalam sebuah proses. <i>Text Annotation</i> dihubungkan menggunakan Asosiasi.
<i>Group</i>		Digunakan untuk mengelompokkan elemen grafis yang mempunyai kategori yang sama. Nama kategori muncul dalam diagram sebagai label grup.

Sumber: Object Management Group (OMG) (2011)

2.8 Perancangan Sistem

Perancangan adalah sebuah proses mendefinisikan sesuatu yang akan dikerjakan dengan teknik bervariasi serta di dalamnya melibatkan deskripsi mengenai arsitektur serta detail komponen dan juga keterbatasan yang akan dialami dalam proses pengerjaannya. Menurut Pressman (2010), perancangan perangkat lunak adalah proses dimana analisis diterjemahkan menjadi blueprint untuk membangun perangkat lunak. Awalnya, *blueprint* menggambarkan

pandangan menyeluruh perangkat lunak, yaitu desain pada abstraksi tingkat tinggi yang dapat langsung ditelusuri pada sistem tertentu, objektif, data rinci, dan fungsionalitas. Namun perbaikan berikutnya mengarah pada representasi desain dengan tingkat abstraksi yang jauh lebih rendah.

Berikut ini merupakan prinsip dalam melakukan perancangan (Pressman, 2010):

- a. Hasil rancangan harus dapat dilacak dalam model persyaratan.
- b. Arsitektur sistem yang akan dibangun selalu dipertimbangkan.
- c. Rancangan dari data dan rancangan fungsi memiliki urgensi yang sama.
- d. Harus hati – hati dalam merancang antarmuka.
- e. Desain antarmuka pengguna harus sesuai dengan kebutuhan pengguna dan berfokus pada kemudahan pengguna.
- f. Komponen harus memiliki keterikatan yang rendah antara satu sama lain dan dengan lingkungan eksternal.
- g. Model rancangan harus mudah dimengerti.
- h. Rancangan harus dikembangkan secara iteratif.

Pada tahap ini memungkinkan tim pengembang untuk menilai dan meningkatkan kualitas dari hasil perancangan, sebelum lanjut ke tahap selanjutnya. Model rancangan dinilai oleh tim pengembang sebagai upaya untuk menentukan apakah terdapat kesalahan, inkonsistensi, kelalaian, adakah alternatif yang lebih baik? Dan apakah model rancangan tersebut dapat dilaksanakan dalam jadwal, biaya, dan batasan yang telah ditetapkan.

McGlaughlin dalam Pressman (2010) menyarankan tiga karakteristik yang berfungsi sebagai panduan untuk evaluasi perancangan yang baik, yaitu:

1. Sebuah rancangan harus mengimplementasikan semua persyaratan eksplisit yang ada pada model persyaratan, dan harus mengakomodasi semua persyaratan implisit yang diinginkan oleh para pemangku kepentingan.
2. Sebuah rancangan harus dapat dibaca, dipahami untuk mereka yang akan melakukan tahap implementasi dan pengujian serta mereka yang akan mendukung perangkat lunak.
3. Sebuah rancangan harus menyediakan gambaran lengkap dari perangkat lunak, data, fungsional, dan domain perilaku dari perspektif implementasi.

2.9 UML (*Unified Modelling Language*)

UML (*Unified Modelling Language*) merupakan standar bahasa yang digunakan untuk kebutuhan pendefinisian rancangan dalam sebuah perangkat lunak. Bahasa ini digunakan untuk melakukan spesifikasi, visualisasi, konstruksi, dan pendokumentasian artefak dari perangkat lunak yang akan dikembangkan

(Pressman, 2010). UML memiliki diagram-diagram yang digunakan untuk mendesain suatu sistem atau perangkat lunak, diantaranya yaitu:

1. *Use Case Diagram*
2. *Activity Diagram*
3. *Sequence Diagram*
4. *Class Diagram*

2.9.1 Use Case Diagram

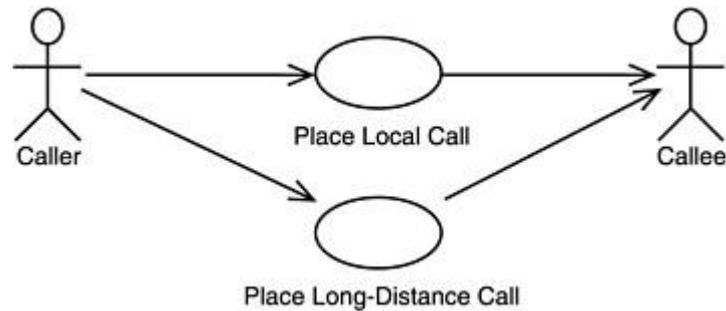
Use Case Diagram merupakan cara sederhana dan terbaik untuk mengekspresikan kebutuhan fungsional, maupun perilaku dari sistem (Bittner et.al, 2003). Sedangkan *use case specification* menggambarkan urutan kegiatan yang dilakukan oleh actor dan sistem untuk mencapai suatu tujuan tertentu. *Use case specification* terdiri dari nama *use case*, deskripsi singkat (*Brief Description*), Aliran Normal (*Basic flow*), Aliran Alternatif (*Alternate Flow*), *Special Requirement*, *Pre-Condition* dan *Post-Condition* (Pressman, 2010). Namun, *use case* hanya menjelaskan tentang apa saja yang dilakukan oleh Stakeholder dan sistem dan tidak menjelaskan bagaimana Stakeholder dan sistem melakukan kegiatan tersebut. Simbol-simbol yang digunakan pada diagram *use case* dapat dilihat pada Tabel 2.3

Tabel 2.3 Simbol – Simbol dalam Use Case Diagram

No	Simbol	Deskripsi
1	 An Actor	Actors. Mendefinisikan suatu peran yang mana pengguna dapat berinteraksi dengan sistem. Pengguna dapat berupa manusia ataupun sistem yang lain
2	 A Use Case	Use Case. Mendeskripsikan bagaimana Stakeholder menggunakan sistem untuk mencapai tujuan dan apa yang sistem lakukan untuk Stakeholder untuk mendapatkan tujuannya.
3		Communicates association. Stakeholder dan <i>use case</i> yang berinteraksi akan terhubung melalui simbol ini. Simbol asosiasi dengan panah mengindikasikan elemen mana yang memulai interaksi.

Sumber: Kurt Bittner & Ian Spence (2003)

Contoh penggambaran diagram *Use Case*:



Gambar 2.4 Contoh penggambaran diagram *Use Case*

Sumber: Kurt Bittner & Ian Spence (2003)

2.9.2 Activity Diagram

Activity Diagram merupakan salah satu dari diagram UML yang digunakan untuk memodelkan aspek dinamis (*Scenario-based elements*) dari sistem. Representasi grafis berupa aliran dari interaksi berdasarkan scenario spesifik (Pressman 2010). Simbol-simbol yang digunakan pada *activity diagram* dapat dilihat pada Tabel 2.4.

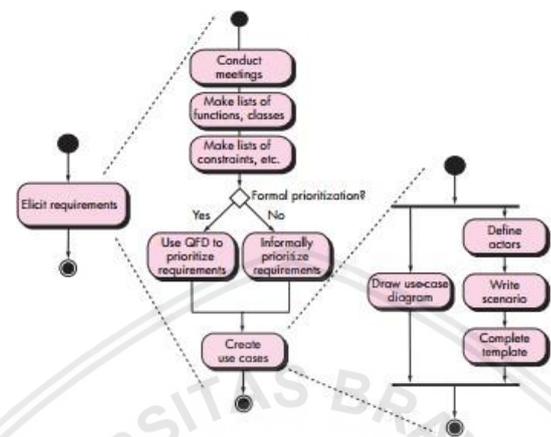
Tabel 2.4 Simbol – simbol dalam *Activity Diagram*

No	Simbol	Deskripsi
1		Initial node. Mendefinisikan titik mulai suatu proses.
2		Actions. Mendefinisikan notasi yang menggambarkan langkah-langkah yang terjadi.
3		Flow. Merupakan panah dalam diagram yang mengindikasikan alur antar- <i>actions</i> .
4		Decisions. Merupakan kondisi tertentu yang menentukan alur keluar.
5		Merge. Merupakan penggabungan alur yang sebelumnya terpisah oleh <i>decisions</i> .
6		Fork. Merupakan <i>bar</i> hitam untuk percabangan aktivitas yang secara bersamaan.
7		Join. Untuk menyatukan lagi alur yang terpisah oleh <i>fork</i> .

8		Activity Final. Merupakan titik akhir proses.
---	---	--

Sumber: Booch et. Al. (2005)

Contoh penggambaran diagram *Activity*:



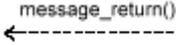
Gambar 2.5 Contoh penggambaran diagram *Activity*

Sumber: Pressman (2010)

2.9.3 Sequence Diagram

Sequence Diagram merupakan diagram interaksi yang menekankan pada saat permintaan pesan yang menunjukkan serangkaian peranan dan pesan yang dikirim dan diterima oleh bagian yang ada pada peranan tersebut. Pesan dapat berupa sinyal atau panggilan terhadap suatu operasi. Notasi pesan untuk panggilan terhadap suatu operasi dapat dituliskan dalam sintaks UML atau sintaks bahasa pemrograman tertentu. Diagram sekuen digunakan untuk menggambarkan *dynamic view* dari sistem (Booch, et al., 2005). Simbol-simbol yang digunakan pada diagram *Sequence* ditunjukkan pada Tabel 2.5.

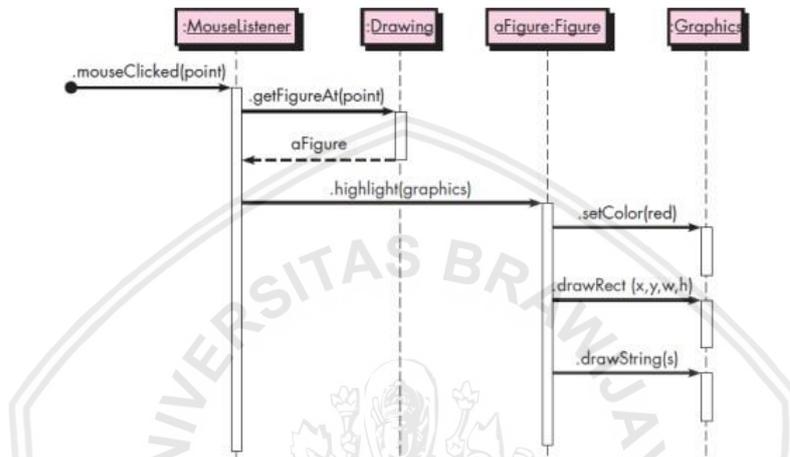
Tabel 2.5 Simbol – Simbol dalam *Sequence Diagram*

No	Simbol	Deskripsi
1		Entity. Mendefinisikan suatu entitas yang mempunyai atribut yang memiliki data yang bisa direkam.
2		Boundary. Mendefinisikan suatu <i>interface</i> yang menghubungkan pengguna dengan sistem.
3		Control. Mendefinisikan kontrol atas aktivitas yang dilakukan oleh sebuah kegiatan.
4		Message. Mendefinisikan pengiriman pesan.
5		Return Values. Mendefinisikan hasil dari pengiriman pesan.

6	-	Lifelines. Mengindikasikan urutan dari kehidupan.
7	█	Bar. Menunjukkan periode waktu ketika aktif dalam interaksi.

Sumber: Booch, et.al (2005)

Contoh penggambaran diagram *Sequence*:



Gambar 2.6 Contoh penggambaran diagram *Sequence*

Sumber: Pressman (2010)

2.9.4 Class Diagram

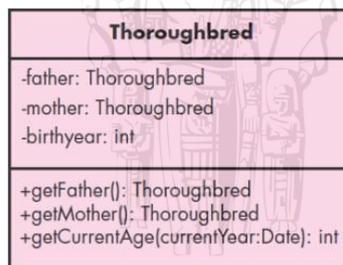
Class diagram atau diagram kelas digunakan untuk memvisualisasikan tampilan statis dari sebuah sistem. Diagram kelas tidak hanya penting untuk memvisualisasikan, menentukan, dan mendokumentasikan model struktural, akan tetapi penting untuk menyusun sistem yang dapat dijalankan melalui teknik *forward* maupun *reverse*. *Class diagram* secara umum memuat tiga hal, yaitu kelas, *interfaces*, dan hubungan dependensi, generalisasi, serta asosiasi (Booch, et al., 2005). Pada Tabel 2.7 menjelaskan tentang simbol-simbol yang digunakan dalam pembuatan *class diagram* pada umumnya.

Tabel 2.6 Simbol – Simbol dalam *Class Diagram*

No	Simbol	Deskripsi
1		<p>Attribute. Mendefinisikan sekumpulan data yang dimiliki oleh objek.</p> <p>Behavior. Kumpulan dari sesuatu yang dapat dilakukan oleh objek dan terkait dengan fungsi-fungsi yang bertindak pada data objek (atribut). Pada siklus berorientasi objek, perilaku objek</p>

		merujuk kepada metode, operasi, atau fungsi.
2		Inheritance. Menunjukkan bahwa satu kelas merupakan turunan dari kelas lain.
3		Association. Menunjukkan bahwa objek dari satu kelas berhubungan dengan kelas lain.
4		Aggregation. Menunjukkan bahwa contoh objek dari satu kelas terdiri dari contoh objek dari kelas lain.
5		Composition. Menunjukkan hubungan dimana satu kelas bertanggung jawab atas pembuatan dan perusakan bagian-bagian dalam kelas lainnya. Jika satu kelas rusak, maka kelas lain juga rusak.

Contoh penggambaran diagram *Class*:



Gambar 2.7 Contoh penggambaran diagram *Class*

Sumber: Pressman (2010)

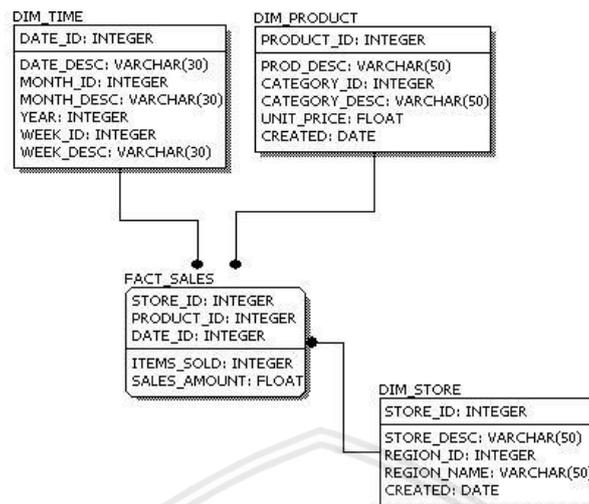
2.9.5 Physical Data Model

Physical data model merepresentasikan bagaimana model akan dibangun di *database*. Sebuah *physical data model* menampilkan struktur tabel yang terdiri dari nama kolom, tipe data kolom, *constraints* kolom, *primary key*, *foreign key*, dan relasi antara tabel (1keydata,2017). Terdapat beberapa tahap dalam pembuatan *physical data model*, yakni:

1. Merubah entitas menjadi tabel.
2. Merubah relasi menjadi *foreign key*.
3. Merubah atribut menjadi kolom.
4. Memodifikasi *physical data model* sesuai dengan kebutuhan.

Pada gambar 2.8 dapat dilihat contoh *physical data model*.





Gambar 2.8 Contoh *physical data model*

Sumber: 1keydata (2017)

2.10 Implementasi Sistem

Implementasi sistem merupakan salah satu tahapan dari siklus pengembangan sistem, dimana desain / rancangan sistem akan diterjemahkan kedalam bentuk bahasa pemrograman tertentu, sehingga menjadikan sistem yang berfungsi secara utuh.

2.10.1 Object Oriented Programming (OOP)

Object Oriented Programming merupakan salah satu paradigma dalam pembangunan sistem yang memandang segala sesuatu merupakan objek dan mempunyai atribut yang melekat, serta memiliki *behavior* (Shalahudin & S, 2015). Keuntungan mengembangkan sistem dengan menggunakan pemrograman berorientasi objek yaitu:

1. Meningkatkan produktivitas karena objek yang ditemukan dapat dipakai ulang (*reusable*).
2. Meningkatkan kecepatan dalam mengembangkan sistem karena telah dirancang dengan baik dan benar sehingga kesalahan dalam implementasi dapat diminimalisir.
3. Mudah dalam pemeliharaan karena menggunakan objek.
4. Adanya konsistensi dengan menggunakan konsep pewarisan (*inheritance*).
5. Meningkatkan kualitas perangkat lunak karena pengembangan didasari atas kejadian nyata, sehingga kebutuhan pengguna terpenuhi.

2.10.2 Codeigniter

Codeigniter adalah *framework* untuk membangun sistem berbasis web dengan menggunakan bahasa PHP. Codeigniter menggunakan aritektur *Model-*

View-Controller (MVC) adalah pola arsitektur yang membagi aplikasi ke dalam 3 komponen yang saling berinteraksi satu dengan yang lainnya (Sommerville, 2011). Masing-masing komponen pada pola arsitektur MVC memiliki peran yang berbeda. *Model* berfungsi untuk mengurus bagian penyimpanan dan pemberian data dari *database*. *View* berfungsi untuk mengurus bagian tampilan data dari *model*, dan *Controller* berfungsi untuk mengatur alur data dari *model* ke *view*.

2.10.3 Aplikasi berbasis Web

Aplikasi berbasis web / *Web Service* merupakan komponen *software* yang berbasis *framework* web dan standar *object-oriented* dan teknologi untuk penggunaan web yang secara elektronik menghubungkan aplikasi *user* yang berbeda dan *platform* yang berbeda (O'Brien, 2010). *Web service* dapat menghubungkan fungsi bisnis untuk pertukaran data secara *real time* dalam aplikasi berbasis web.

2.11 Pengujian Sistem

Pengujian sistem adalah proses menjalankan suatu program dengan tujuan untuk menemukan kesalahan dari sistem yang telah dikembangkan, dan bukan untuk menunjukkan bahwa program tersebut salah. Pengujian yang baik ialah yang memiliki kemungkinan tinggi untuk menemukan kesalahan yang belum pernah ditemukan sebelumnya, dan memiliki probabilitas yang tinggi untuk menemukan kesalahan (Pressman, 2010). Menurut Pressman, terdapat beberapa prinsip-prinsip dalam melakukan pengujian sistem antara lain:

Pengujian berfokus dalam usaha menemukan kesalahan pada modul yang terintegrasi, dan akhirnya pada sistem secara keseluruhan.

1. Pengujian yang dilakukan memiliki kemungkinan tinggi untuk menemukan kesalahan.
2. Pengujian harus sesuai dengan persyaratan pelanggan.
3. Para ahli sistem harus mengetahui spesifikasi dari produk perangkat lunak yang diinginkan pelanggan.
4. Perakayasa perangkat lunak bukanlah orang yang paling tepat untuk melakukan semua pengujian bagi perangkat lunak.

2.11.1 Black-Box Testing

Black-Box Testing merupakan teknik pengujian perangkat lunak yang fokus kepada pengecekan kebutuhan fungsional sistem. Pengujian ini dilakukan untuk mengetahui kesalahan *error* yang ada pada perangkat lunak. *Error* yang dimaksud antara lain kesalahan atau hilangnya fungsi, kesalahan antarmuka, kesalahan pada *database*, kesalahan kinerja, dan kesalahan inisialisasi dan terminasi (Pressman, 2010). Pengujian *black box* memiliki beberapa kelebihan sebagai berikut:

1. Pengujian dilakukan dari sudut pandang pengguna dan akan membantu dalam mengungkap ketidaksesuaian dalam spesifikasi perangkat lunak.

2. *Tester* tidak perlu tahu bahasa pemrograman atau bagaimana perangkat lunak diimplementasikan.
3. Pengujian dapat dilakukan oleh badan independen dari pengembang, memungkinkan untuk perspektif objektif.
4. *Test case* dapat langsung dirancang setelah spesifikasi telah lengkap.

2.11.1.1 *Validation Testing*

Validation Testing merupakan merupakan teknik pengujian dalam lingkup metode *Black Box* yang mana kesesuaian di uji. Pengujian ini dilakukan untuk mengetahui apakah sistem yang dibangun berjalan sesuai dengan kebutuhan yang telah didefinisikan atau tidak.

Pengujian ini termasuk dalam *black-box testing*, karena pengujian dilakukan tanpa menunjukkan proses berjalannya sebuah fungsi yang sebenarnya, tetapi pengujian validasi melihat hasil yang dikeluarkan apakah sesuai dengan kebutuhan atau tidak, dan lebih ditekankan untuk menemukan konformitas antara kinerja sistem dengan daftar spesifikasi kebutuhan (Indriati, 2010).

Validation testing perangkat lunak dapat dilakukan dengan menggunakan *use case* sebagai panduan pengujian. *Use case* menyediakan skenario yang mendeteksi kemungkinan kesalahan pada perangkat lunak. Setelah pengujian selesai dilakukan, akan didapatkan dua hasil, yaitu (Pressman, 2010):

1. Fungsi atau performa sistem telah sesuai dengan spesifikasi dan diterima oleh pelanggan
2. Kesalahan pada perangkat lunak yang dikembangkan terlihat dan kemudian akan dicatat.

2.11.2 *User Acceptance Testing (UAT)*

User Acceptance Testing (UAT) adalah pengujian yang menangani *requirement, business process*, dan kebutuhan pengguna. Pengujian ini dilakukan untuk mengetahui apakah suatu sistem yang dibangun telah memenuhi kriteria untuk diterima oleh pengguna atautakah tidak (Hambling, 2013).

User Acceptance Testing biasanya dilakukan sebelum produk dirilis dan dilakukan setelah pengujian sistem. UAT dilakukan oleh pengguna / *stakeholder* yang terlibat dengan sistem yang dibangun.

Pada pengujian UAT terdapat beberapa kriteria yang dapat membantu dalam proses penilaian penerimaan sistem. Akan tetapi pada penerapannya, tidak harus menguji semua kriteria pada sistem, karena mengakibatkan terlalu banyaknya hasil pengujian. Tim pengembang dapat memilih bagian penting yang dapat mewakili hasil penerimaan sistem. Berikut merupakan kriteria yang ada pada pengujian UAT (Naik & Tripathy, 2008).

1. *Functional Correctness and Completeness*, berfokus pada apakah sebuah sistem telah sesuai dengan harapan pengguna berdasarkan hasil analisis kebutuhan.

2. *Accuracy*, berfokus pada apakah perhitungan atau hasil dari sistem telah sesuai dengan nilai yang seharusnya.
3. *Data Integrity*, berfokus pada performa sistem dalam melakukan pengolahan data.
4. *Data Conversion*, berfokus pada performa sistem dalam menjaga data ketika terjadi perubahan.
5. *Backup and Recovery*, berfokus pada proses *backup* serta pemulihan data.
6. *Competitive Edge*, berfokus pada keunggulan sebuah sistem dibandingkan dengan model yang lain.
7. *Usability*, berfokus pada kemudahan pengguna dalam menggunakan sistem.
8. *Performance*, berfokus pada kinerja dari sistem.
9. *Start-Up Time*, berfokus pada waktu yang dibutuhkan sebuah sistem untuk mulai beroperasi.
10. *Stress*, berfokus pada kemampuan sebuah sistem dalam menangani beban proses yang sangat tinggi dan menguji batas kemampuannya.
11. *Reliability and Availability*, berfokus pada keandalan sebuah sistem untuk melakukan eksekusi tanpa kegagalan pada waktu tertentu dengan lingkungan yang telah ditentukan.
12. *Maintainability and Serviceability*, berfokus kepada kemampuan sistem untuk menjalani perbaikan dan proses evolusinya.
13. *Robustness*, berfokus pada kemampuan sistem untuk kembali pulih ketika terjadi kesalahan, tetap beroperasi dalam kemungkinan terburuk, dan tetap beroperasi dalam jangka waktu yang panjang.
14. *Timeliness*, berfokus pada waktu yang dibutuhkan sistem dalam perhitungan *time to market*.
15. *Confidentiality and Availability*, berfokus pada kerahasiaan persyaratan bahwa data harus dilindungi dari pengungkapan tidak sah dan persyaratan bahwa data harus dilindungi dari pelunakan layanan (DoS) kepada pengguna yang berwenang.
16. *Compatibility and Interoperability*, berfokus pada kompatibilitas sistem atau kemampuan sistem untuk beroperasi dengan cara yang sama pada berbagai platform dan konfigurasi jaringan serta di berbagai campuran aplikasi lainnya.
17. *Compliance*, berfokus pada kesesuaian sistem terhadap standar teknis yang relevan.

- 18. *Installability and Upgradability*, berfokus pada memastikan sistem dapat diinstal dengan benar dan dapat dilakukan pada proses *upgrade* pada lingkungan pengguna.
- 19. *Scalability*, berfokus pada kemampuan sistem untuk secara efektif memberikan kinerja yang dapat diterima.
- 20. *Documentation*, berfokus pada tingkat kualitas panduan untuk menggunakan sistem.

2.11.2.2 Skala Likert

Setelah sudah menentukan kriteria pada pengujian UAT, kemudian menghitung data dengan menggunakan skala *likert*. Skala *Likert* merupakan salah satu cara yang berguna dalam mempelajari hubungan antara sikap dan atau persepsi individu diperiksa dan dianalisis (McLeod, et al, 2011). Pada implementasinya, skala ini digunakan dengan cara meminta beberapa responden untuk memberikan opini mengenai pertanyaan yang diajukan. Responden akan menilai sejauh mana mereka setuju ataupun tidak setuju mengenai solusi dari permasalahan yang ditentukan. Proses perhitungan pada skala *likert* diawali dengan memberikan bobot nilai apda setiap jawaban. Tabel 2.7 berisi rincian bobot nilai jawaban, sedangkan tabel 2.8 berisi persentase untuk setiap jawaban.

Tabel 2.7 Rincian Bobot Nilai

A = Sangat Setuju	5
B = Setuju	4
C = Netral	3
D = Kurang	2
E = Sangat Kurang	1

Sumber: Choizes (2017)

Tabel 2.8 Rincian Bobot Nilai

80 sampai 100%	Sangat Setuju
60 sampai 79,99%	Setuju
40 sampai 59,99%	Netral
20 sampai 39,99%	Tidak Setuju
0 sampai 19,99%	Sangat Kurang Setuju

Sumber: Choizes (2017)

Setelah bobot dan persentase nilai ditentukan, selanjutnya menghitung jumlah jawaban, dikalikan dengan bobot nilai. Berikut merupakan perhitungannya:

1. Jawaban Sangat Setuju (A) = $nx5$



2. Jawaban Setuju (B) = $nx4$
3. Jawaban Netral (C) = $nx3$
4. Jawaban Tidak Setuju (D) = $nx2$
5. Jawaban Sangat Kurang Setuju (E) = $nx1$

$$TotalNilai = (nx5) + (nx4) + (nx3) + (nx2) + (nx1) \quad (2.1)$$

n merupakan jumlah responden yang menjawab. Untuk melakukan perhitungan selanjutnya harus diketahui terlebih dahulu nilai terendah (Y) dan nilai tertinggi (X). rumus yang digunakan dapat dilihat pada Rumus 2.2 dan 2.3, dimana $N1$ merupakan nilai tertinggi *likert*, $N2$ merupakan nilai terendah *likert*, n merupakan jumlah responden, dan U merupakan jumlah uji kasus.

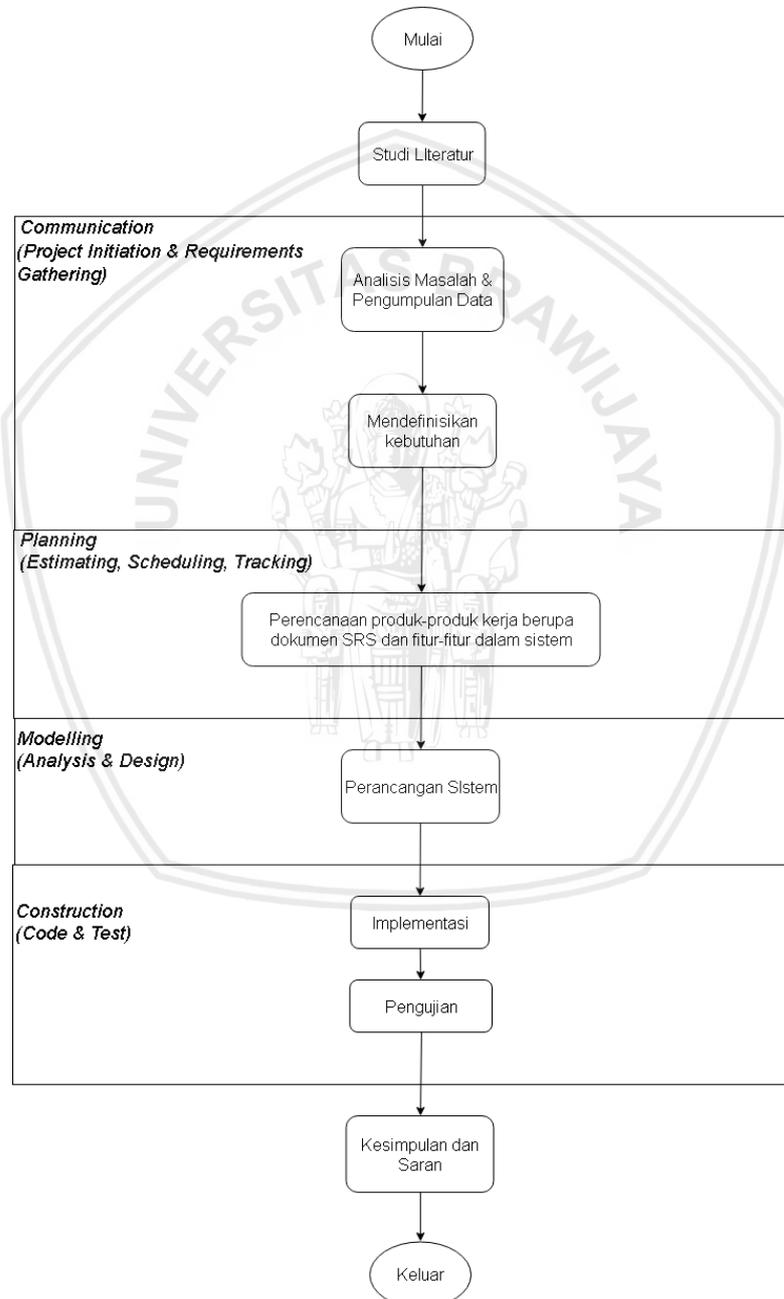
$$Y = N1nxU \quad (2.2)$$

$$X = N2nxU \quad (2.3)$$



BAB 3 METODOLOGI

Penelitian ini merupakan penelitian implementatif-konstruksi, dimana produk yang dihasilkan berupa sistem informasi yang siap untuk digunakan. Pendekatan implementatif yang digunakan dalam penelitian ini menggunakan pendekatan berorientasi objek. Langkah - langkah yang akan dilakukan dalam penelitian ini adalah *Communication*, *Planning*, *Modeling*, dan *Construction*. Gambar 3.1 menyajikan alur penelitian yang akan dilakukan dalam penelitian ini.



Gambar 3.1 Alur Penelitian

3.1 Communication (Requirements Gathering)

Dalam tahap *Communication* merupakan tahap mencari, mempelajari dan menyusun teori dasar dan referensi yang mendukung pengembangan sistem informasi. Literatur yang digunakan berasal dari buku, jurnal atau *paper*, atau penelitian yang telah dilakukan peneliti sebelumnya. Kemudian melakukan analisis masalah dan pengumpulan data. Analisis masalah tersebut dilalui dengan melakukan wawancara kepada pihak sekolah untuk mengumpulkan informasi dan masalah yang kemudian di analisis manakah masalah yang dapat dijadikan sebagai latar belakang penelitian dan bisa diselesaikan dengan menggunakan metode serta literatur yang digunakan, dan pengumpulan data berupa data rapor dan dokumen-dokumen yang dibutuhkan dari SMA Sejahtera 1 Depok. Kemudian setelah dilakukan analisis masalah dan pengumpulan data, mendefinisikan kebutuhan untuk mencari apa yang dibutuhkan, apa yang bisa dilakukan sistem, baik berupa kebutuhan fungsional maupun kebutuhan nonfungsional.

3.2 Planning (Estimating, Scheduling, Tracking)

Tahap berikutnya adalah tahap *planning* merupakan tahap perencanaan produk-produk kerja yang akan dihasilkan berupa dokumen SRS, dan fitur-fitur dalam sistem yang akan dilaksanakan dengan pihak SMA Sejahtera 1 Depok.

3.3 Modeling (Analysis & Design)

Kemudian tahap *modelling* merupakan tahap perancangan dan permodelan arsitektur sistem seperti perancangan *Unified Modelling Language* (UML) berupa *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram*, dan *physical data model*. Kemudian basis data yang disesuaikan dengan kebutuhan pengguna.

3.4 Construction (Code & Test)

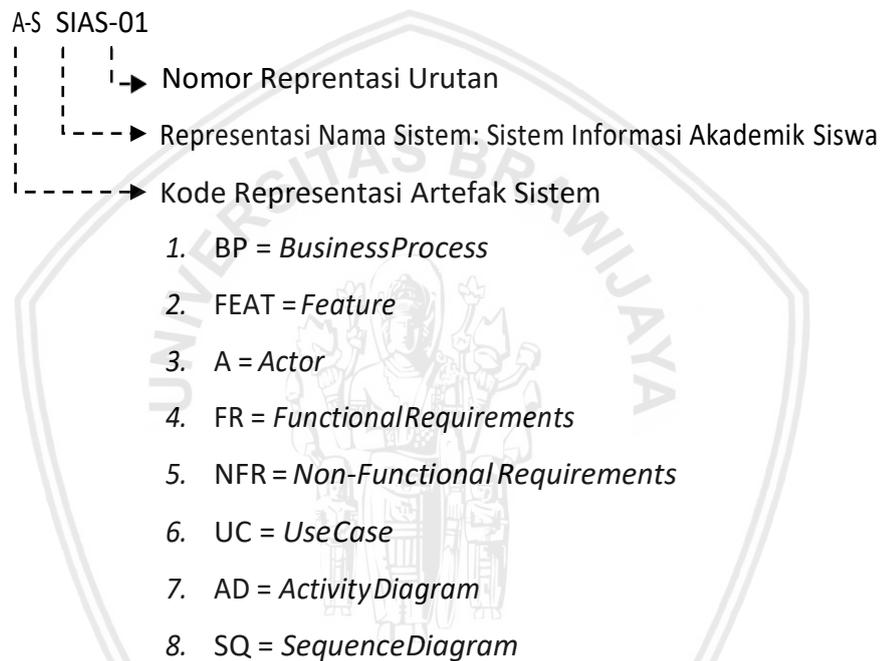
Setelah sudah melakukan perancangan dan permodelan arsitektur sistem. Tahap selanjutnya ialah *Construction*. Pada tahap ini sudah dimulainya implementasi sistem. Setelah implementasi sistem sudah selesai, selanjutnya dilakukan pengujian terhadap sistem yang telah dibuat. Tujuan pengujian adalah menemukan kesalahan-kesalahan terhadap sistem untuk kemudian bisa diperbaiki. Dalam tahap ini terdapat tiga kegiatan, yaitu implementasi *database*, implementasi program dan pengujian. Pengujian sistem pada penelitian ini menggunakan *validation testing* dan *User Acceptance Test*.

BAB 4 ANALISIS KEBUTUHAN

Dalam pengembangan sistem informasi manajemen akademik yang akan dilakukan, Analisis kebutuhan dilakukan mulai aturan penomoran, pemodelan proses bisnis, identifikasi *stakeholder*, menspesifikasikan kebutuhan fungsional maupun nonfungsional sistem, menggambarkan kebutuhan kedalam *use case diagram*, hingga menuliskan *use case scenario*.

4.1 Aturan Penomoran

Aturan penomoran digunakan untuk memudahkan kodifikasi dan *traceability*. Berikut aturan penomoran yang digunakan, serta realisasi aturan pada tabel 4.1



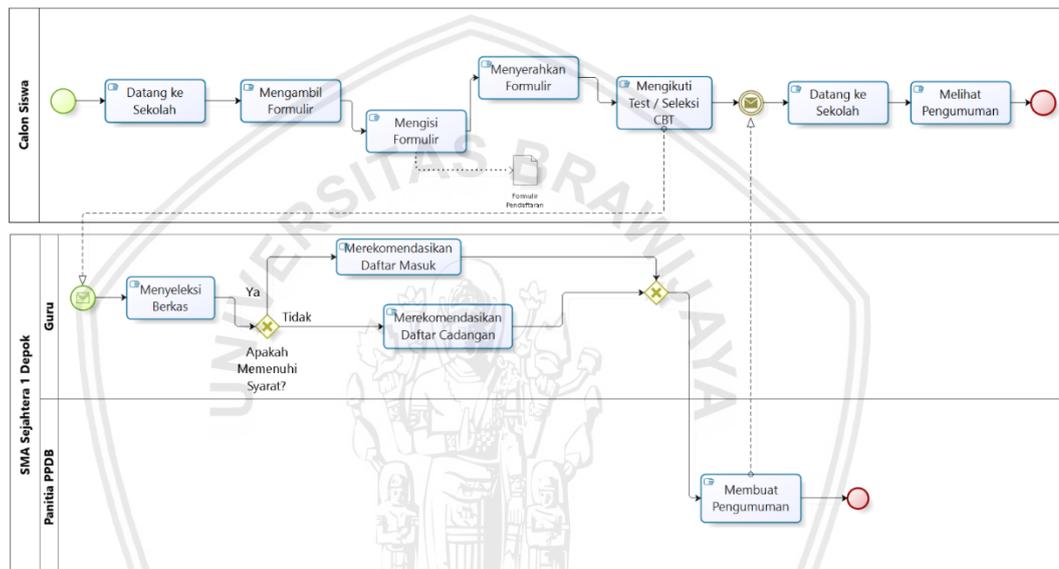
Tabel 4.1 Realisasi Aturan Penomoran

Artefak	Aturan Penomoran	Contoh Realisasi
<i>Business Process</i>	BP-SIAS-x	BP-SIAS-1
Identifikasi Fitur	FEAT-SIAS-x	FEAT-SIAS-1
Identifikasi Stakeholder	A-SIAS-x	A-SIAS-1
Kebutuhan Fungsional	F-SIAS-x	F-SIAS-1
Kebutuhan nonfungsional	NF-SIAS-x	NF-SIAS-1
<i>Use Case</i>	UC-SIAS-x	UC-SIAS-1
<i>Activity Diagram</i>	AD-SIAS-x	AD-SIAS-1
<i>Sequence Diagram</i>	SD-SIAS-x	SD-SIAS-1

4.2 Pemodelan Proses Bisnis Saat Ini

4.2.1 Proses Bisnis Pendaftaran Peserta Didik Baru Saat ini

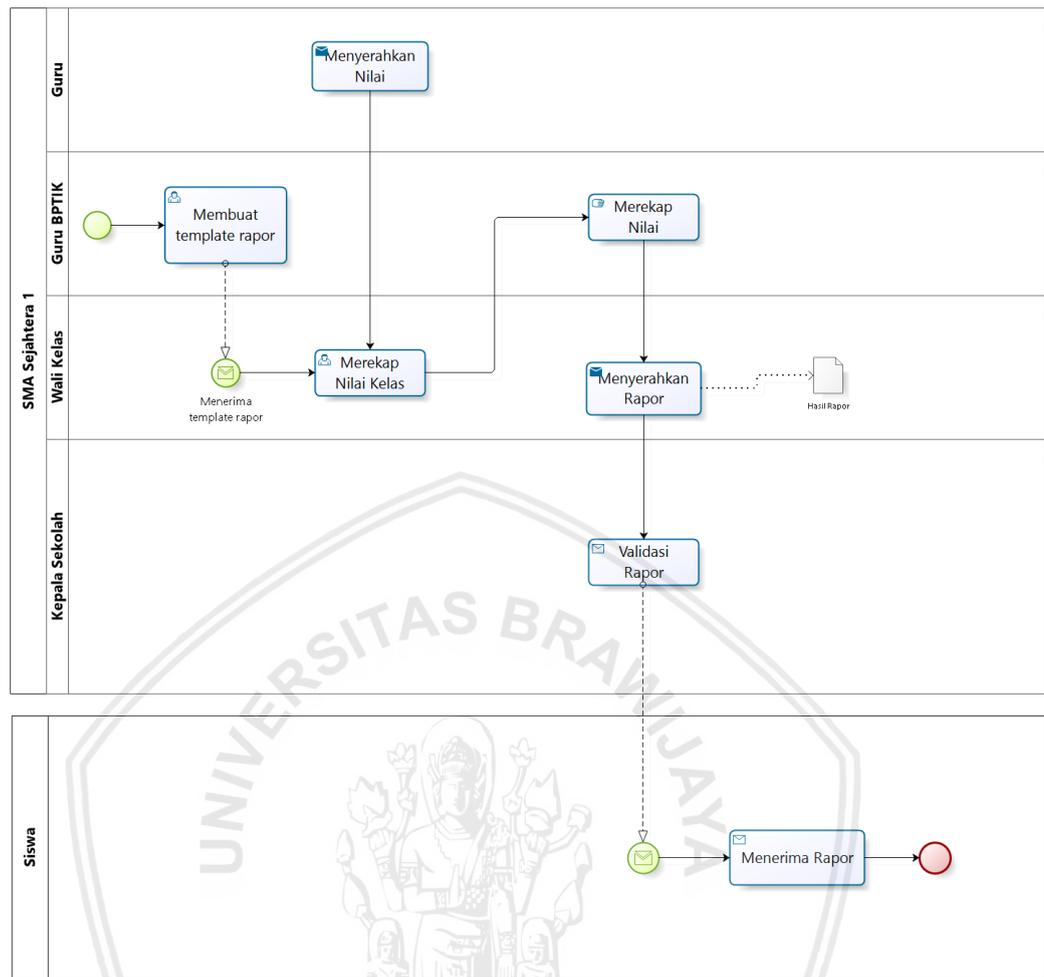
Berdasarkan hasil wawancara dengan Guru Bimbingan dan Penyuluhan TIK (BPTIK) terdapat beberapa proses Pendaftaran Peserta Didik Baru (PPDB) di SMA Sejahtera 1, yaitu diawali dengan calon siswa datang ke sekolah untuk mengambil formulir pendaftaran PPDB. Kemudian, calon siswa mengisi formulir pendaftaran PPDB, setelah sudah mengisi formulir tersebut, calon siswa menyerahkan formulir pendaftaran PPDB tersebut kepada panitia PPDB. Kemudian calon siswa mengikuti test / seleksi menggunakan CBT (*Computer Based System*) di SMA Sejahtera 1 Depok.



Gambar 4.1 Proses Bisnis Pendaftaran Peserta Didik Baru Saat Ini

4.2.2 Proses Bisnis Mengevaluasi Siswa Saat Ini

Berdasarkan hasil wawancara dengan Guru Bimbingan dan Penyuluhan TIK (BPTIK), terdapat beberapa proses mengevaluasi siswa, yaitu diawali dengan guru BPTIK membuat *template* rapor, kemudian guru menyerahkan nilai per mata pelajaran ke wali kelas untuk merekap nilai per kelas. Lalu wali kelas selesai merekap nilai per kelas, sesuai *template* yang sudah ada. Setelahnya dikirim ke guru BPTIK, untuk diolah dan direkap hasil entri nilai dari wali kelas. Setelah itu, hasil rekap nilai tersebut guru BPTIK memberikan hasil rekap rapor ke wali kelas, dan wali kelas memberikan hasil rapor ke kepala sekolah untuk validasi. Setelahnya siswa menerima laporan penyampaian nilai (rapor).



Powered by bizagi Modeler

Gambar 4.2 Proses Bisnis Mengevaluasi Siswa Saat Ini

4.3 Analisis Proses Bisnis Yang Sedang Berjalan (as-is)

Analisis permasalahan proses bisnis yang sedang berjalan bertujuan untuk mencari tahu kendala dan permasalahan apa saja yang timbul pada proses bisnis yang sedang berjalan, dan juga mencari dampak dari kendala dan permasalahan yang terjadi terhadap Sekolah. Proses bisnis ialah proses bisnis Pendaftaran Peserta Didik Baru (PPDB) dan proses bisnis mengevaluasi siswa. Kendala atau permasalahan yang terjadi beserta dampak dan solusinya dideskripsikan pada tabel 4.2 dan tabel 4.3.



Tabel 4.2 Analisis Proses Bisnis Pendaftaran Peserta Didik Baru

Kendala / Permasalahan	<ol style="list-style-type: none"> 1. Pengambilan formulir pendaftaran masih manual, jadi calon siswa harus mengambil formulir pendaftaran ke sekolah 2. Jika ingin melihat pengumuman, calon siswa harus datang ke sekolah
Dampak	<ol style="list-style-type: none"> 1. Dikarenakan formulir pendaftaran masih manual / berbentuk kertas dan harus diambil ke sekolah, maka membutuhkan waktu untuk ke sekolah dan tidak fleksibel harus mempersiapkan waktu untuk datang ke sekolah
Solusi	<ol style="list-style-type: none"> 1. Membuat sistem informasi yang dapat memfasilitasi proses Pendaftaran Peserta Didik Baru (PPDB) 2. Membuat fitur untuk mendaftar secara <i>online</i> dan mencetak formulir secara otomatis

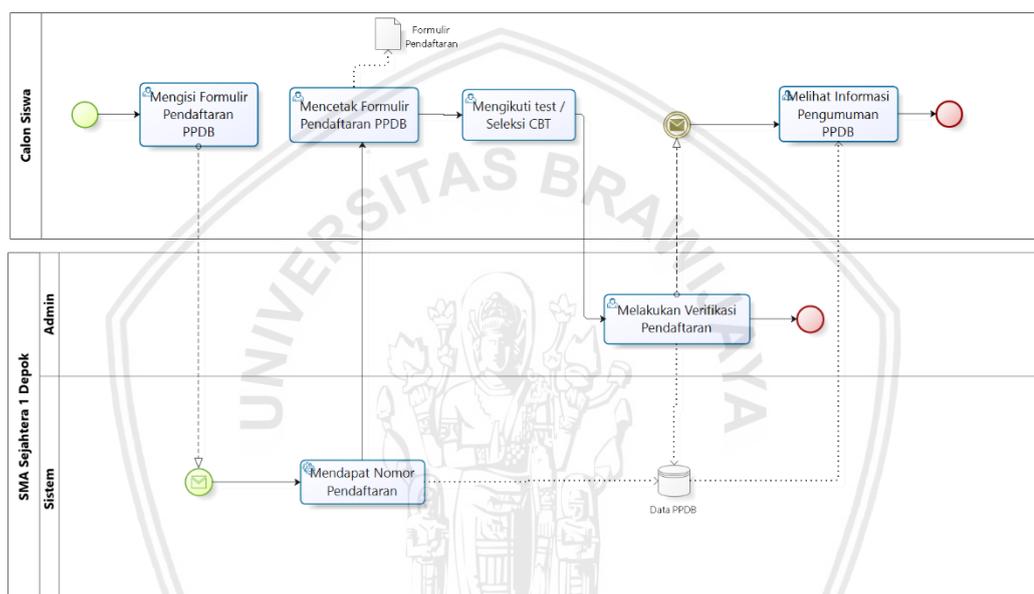
Tabel 4.3 Analisis Proses Bisnis Mengevaluasi Siswa

Kendala / Permasalahan	<ol style="list-style-type: none"> 1. Distribusi penilaian untuk rapor siswa sangat kompleks, jadi harus melalui berbagai macam tahap, seperti wali kelas mengentri nilai siswa dan diberikan ke guru BPTIK, kemudian guru BPTIK merekap hasil nilai tersebut. Kemudian diberikan kembali ke wali kelas.
Dampak	<ol style="list-style-type: none"> 1. Dikarenakan distribusi penilaian rapor siswa sangat kompleks, dan harus melalui berbagai macam tahap. Maka membutuhkan waktu yang cukup lama untuk mengevaluasi siswa.
Solusi	<ol style="list-style-type: none"> 1. Membuat sistem informasi yang dapat memfasilitasi proses evaluasi rapor siswa 2. Membuat fitur untuk mencetak rapor secara otomatis

4.4 Pemodelan Proses Bisnis Usulan

4.4.1 Proses Bisnis Pendaftaran Peserta Didik Baru Usulan

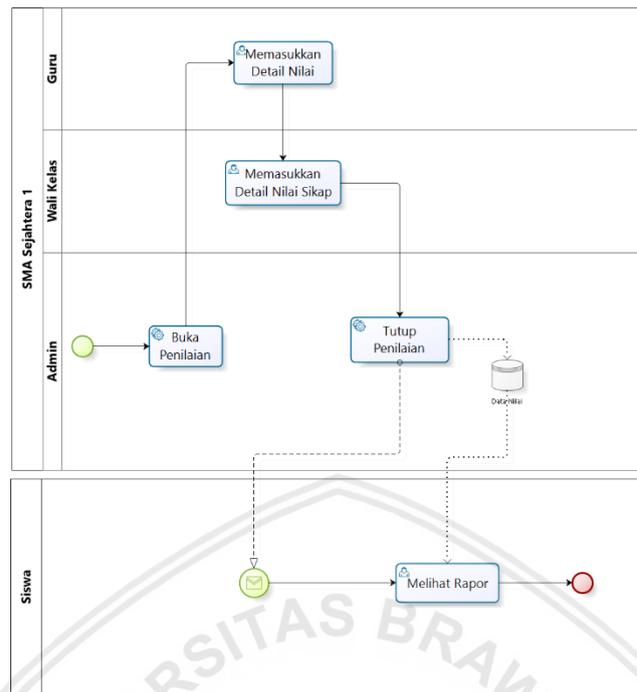
Proses Pendaftaran Peserta Didik Baru (PPDB) usulan, diawali dengan calon siswa mengisi formulir pendaftaran PPDB. Kemudian sistem memproses nomor pendaftaran. Selanjutnya calon siswa mencetak nomor pendaftaran. Setelahnya calon siswa mengikuti test / seleksi dengan CBT (*Computer Based Test*) di SMA Sejahtera 1 Depok. Admin kemudian melakukan verifikasi pendaftaran PPDB yang sudah mengisi formulir dan melakukan test CBT oleh calon siswa. Selanjutnya, calon siswa dapat melihat informasi pengumuman PPDB.



Gambar 4.3 Proses Bisnis Pendaftaran Peserta Didik Baru Usulan

4.4.2 Proses Bisnis Mengevaluasi Siswa Usulan

Proses mengevaluasi siswa usulan, diawali dengan admin membuka akses penilaian, selanjutnya guru memasukkan detail nilai dan wali kelas memasukkan detail sikap. Kemudian, admin melakukan tutup penilaian yang dilakukan otomatis oleh sistem. Lalu, siswa dapat melihat rapor pada sistem.



Gambar 4.4 Proses Bisnis Mengevaluasi Siswa Usulan

Aktivitas proses bisnis usulan dikodekan dengan BP-SIAS-x, dan diidentifikasi aktivitas proses bisnis mana yang akan difasilitasi sistem informasi yang akan dibuat. Tabel 4.4 menunjukkan kodifikasi dari aktivitas proses bisnis yang akan difasilitasi oleh sistem informasi.

Tabel 4.4 Kodifikasi Aktivitas Proses Bisnis

Proses Bisnis	Aktivitas	Kode Aktivitas
Pendaftaran Peserta Didik Baru	Mengisi Formulir Pendaftaran Calon Siswa	BP-SIAS-1
Pendaftaran Peserta Didik Baru	Melakukan Verifikasi Pendaftaran	BP-SIAS-2
Mengevaluasi Calon Siswa	Memasukkan Detail Nilai	BP-SIAS-3
Mengevaluasi Calon Siswa	Memasukkan Detail Nilai Sikap	BP-SIAS-4
Mengevaluasi Calon Siswa	Melihat Detail Nilai	BP-SIAS-5

4.5 Identifikasi Stakeholder

Stakeholder yang terlibat dalam sistem diidentifikasi berdasarkan hasil pemodelan proses bisnis, dan dijelaskan dalam tabel 4.4.

Tabel 4.5 Identifikasi Stakeholder

Kode Stakeholder	Pool / Lane	Stakeholder	Deskripsi
A-SIAS-1	Admin	Admin	Admin merupakan pengurus yang dikhususkan untuk mengatur aktivitas akademik seperti memverifikasi calon siswa, menambah data siswa, data guru, memverifikasi Calon Siswa baru dan menentukan akses penilaian.
A-SIAS-2	Guru	Guru	Guru adalah tenaga pengajar SMA Sejahtera 1 Depok, yang memasukkan detail nilai siswa.
A-SIAS-3	Wali Kelas	Wali Kelas	Wali Kelas adalah Guru yang bertanggung jawab dalam satu kelas, yang memasukkan detail nilai sikap.
A-SIAS-4	Calon Siswa	Calon Siswa	Calon Siswa adalah calon peserta didik SMA Sejahtera 1 Depok, yang mendaftar sebagai siswa SMA Sejahtera 1 Depok
A-SIAS-5	Siswa	Siswa	Siswa adalah siswa SMA Sejahtera 1 Depok, yang melihat Detail Nilai Siswa

4.6 Identifikasi Fitur

Fitur merepresentasikan solusi yang ditawarkan, dan dijelaskan pada tabel 4.3.

Tabel 4.6 Identifikasi Fitur

Kode Fitur	Fitur	Stakeholder	Deskripsi
FEAT-SIAS-1	Mengisi Formulir Pendaftaran Peserta Didik Baru	Calon Siswa	Sistem dapat digunakan untuk mendaftar Calon Siswa baru
FEAT-SIAS-2	Memverifikasi Pendaftaran	Admin	Sistem dapat digunakan untuk memverifikasi pendaftaran Calon Siswa baru
FEAT-SIAS-3	Memasukkan Detail Nilai	Guru	Sistem dapat digunakan untuk memasukkan detail nilai Siswa
FEAT-SIAS-4	Memasukkan Detail Nilai Sikap	Wali Kelas	Sistem dapat digunakan untuk memasukkan detail nilai sikap Siswa
FEAT-SIAS-5	Melihat Detail Nilai	Siswa	Sistem dapat digunakan untuk melihat detail nilai Siswa



FEAT-SIAS-6	Autentikasi	Admin Guru Wali Kelas Calon Siswa Siswa	Sistem dapat digunakan untuk mengautentikasi pengguna
-------------	-------------	---	---

4.7 Kebutuhan Fungsional

Kebutuhan fungsional didapatkan berdasarkan hasil identifikasi fitur, dan dinyatakan dalam *Software Requirement Statement* seperti yang tertera pada tabel 4.5.

Tabel 4.7 Kebutuhan Fungsional

Kode Proses	Kode SRS	Fitur	Stakeholder	<i>Software Requirement Statement</i>	<i>Level</i>
BP-SIAS-1	F-SIAS-1	Mengisi Formulir Pendaftaran Peserta Didik Baru	Calon Siswa	Sistem harus bisa menampilkan formulir untuk menangani pendaftaran Calon Siswa	Normal
BP-SIAS-2	F-SIAS-2	Memverifikasi Pendaftaran	Admin	Sistem harus bisa menangani verifikasi pendaftaran Calon Siswa	Normal
BP-SIAS-3	F-SIAS-3	Memasukkan Detail Nilai	Guru	Sistem harus bisa menampilkan formulir untuk menangani pemasukan detail nilai Siswa	Normal
BP-SIAS-4	F-SIAS-4	Memasukkan Detail Nilai Sikap	Wali Kelas	Sistem harus bisa menampilkan formulir untuk menangani pemasukan detail nilai sikap Siswa	Normal
BP-SIAS-5	F-SIAS-5	Melihat Detail Nilai	Calon Siswa Siswa	Sistem harus bisa menampilkan daftar detail nilai Siswa	Expected

-	F-SIAS-6	Autentikasi	Admin Guru Wali Kelas Calon Siswa Siswa	Sistem harus bisa menampilkan form untuk memasukkan nomor induk dan password, untuk masuk kedalam sistem	Expected
---	----------	-------------	---	--	----------

4.8 Kebutuhan nonfungsional

Kebutuhan nonfungsional dinyatakan dalam *Software Requirement Statement* seperti yang tertera pada tabel 4.6 berikut dibawah ini:

Tabel 4.8 Kebutuhan Non-Fungsional

Kode SRS	<i>Software Requirement Statement</i>	Level
NF-SIAS-1	<i>Compability</i> - Sistem dapat dijalankan di beberapa macam aplikasi <i>browser</i>	Normal

Terdapat satu buah kebutuhan nonfungsional yang harus dipenuhi oleh sistem, yaitu kebutuhan terhadap komabilitas sistem, dimana sistem dapat dijalankan dengan fungsionalitas yang sama pada beberapa aplikasi *browser* yang umum digunakan dalam mengakses internet yaitu Google Chrome dan Mozilla Firefox.

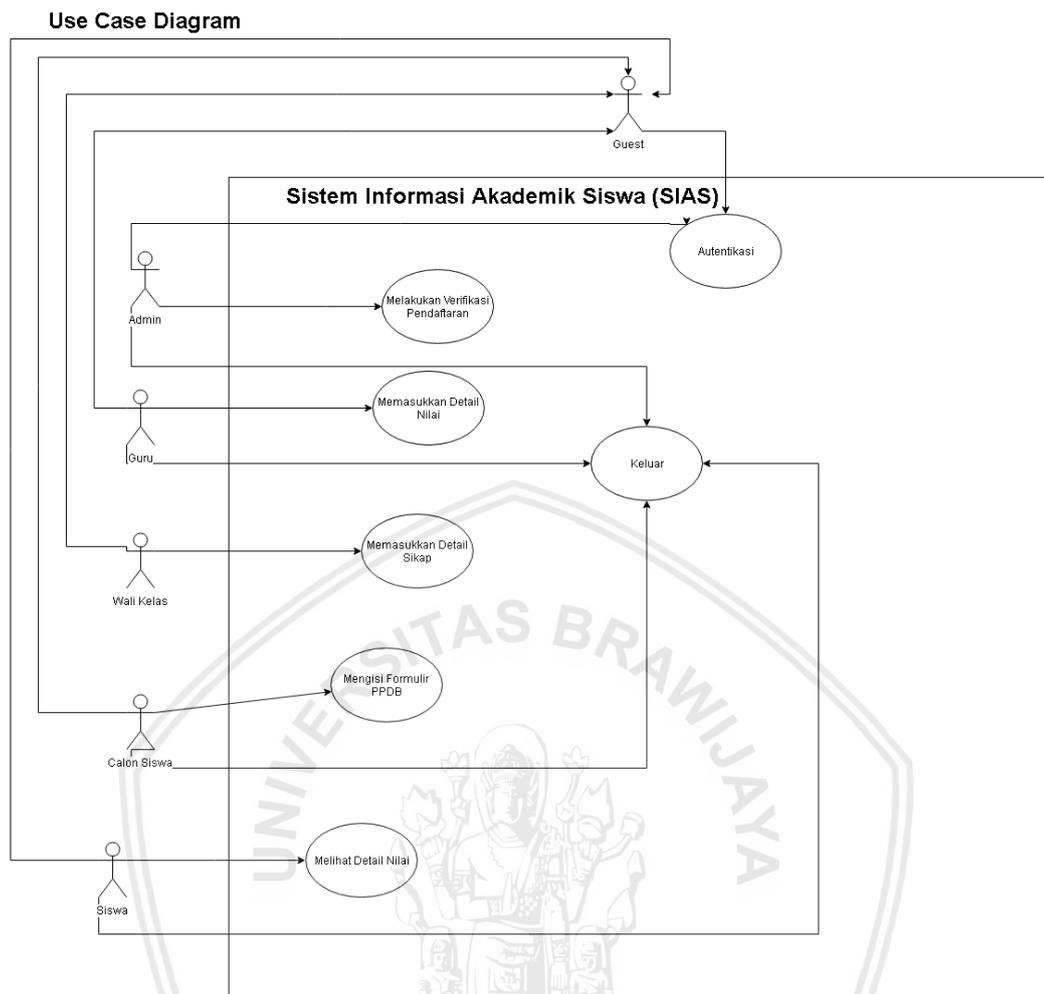
4.9 Use Case Diagram

Berdasarkan kebutuhan fungsional yang telah didefinisikan, dapat diidentifikasi *Use Case* yang dituliskan pada tabel 4.7

Tabel 4.9 Identifikasi Use Case

Kode Use Case	Kode SRS	Use Case
UC-SIAS-1	F-SIAS-1	Mengisi Formulir Pendaftaran Calon Siswa
UC-SIAS-2	F-SIAS-2	Melakukan Verifikasi Pendaftaran
UC-SIAS-3	F-SIAS-3	Memasukkan Detail Nilai
UC-SIAS-4	F-SIAS-4	Memasukkan Detail Sikap
UC-SIAS-5	F-SIAS-5	Melihat Detail Nilai
UC-SIAS-6	F-SIAS-6	Autentikasi
UC-SIAS-7	-	Keluar

Dalam ilustrasi dari sudut pandang perlakuan *Stakeholder* terhadap sistem, dapat digambarkan menggunakan *use case diagram* pada gambar 4.5



Gambar 4.5 Use Case Diagram

4.10 Use Case Scenario

Use Case yang telah dianalisis, akan dijelaskan alur jalannya dari sisi Stakeholder dan sistem, menggunakan Use Case Scenario.

4.10.1 Use Case Scenario Mengisi Formulir Pendaftaran Calon Siswa (UC-SIAS-1)

Use Case Scenario mengisi formulir pendaftaran calon siswa yang merupakan proses untuk mendaftarkan calon siswa, dijelaskan pada Tabel 4.10.

Tabel 4.10 Use Case Scenario Mengisi Formulir Pendaftaran Peserta Didik Baru

<i>Use Case Name</i>	Mengisi Formulir Pendaftaran Peserta Didik Baru
<i>Brief Description</i>	Merupakan proses untuk mendaftarkan Calon Siswa, yang dilakukan oleh Calon Siswa dengan cara mengisi formulir pendaftaran Calon Siswa
<i>Actor</i>	Calon Siswa
<i>Pre-Condition</i>	Aktor terhubung dengan SIAS

<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Calon Siswa mengakses menu formulir Pendaftaran Peserta Didik Baru 2. Sistem menampilkan halaman formulir Pendaftaran Peserta Didik Baru 3. Calon Siswa mengisi seluruh isian formulir Pendaftaran Peserta Didik Baru dan menyertakan berkas yang perlu diunggah 4. Sistem menyimpan seluruh isian formulir Pendaftaran Peserta Didik Baru (PPDB)
<i>Alternative Flows</i>	<p>Alt1 Calon Siswa tidak mengisi formulir PPDB dengan lengkap</p> <p>Jika pada langkah 3, calon siswa tidak mengisi formulir dengan lengkap, maka ketika proses daftar, sistem akan menampilkan sebuah pesan untuk mengisikan bagian dari formulir yang belum terisi / kosong</p>
<i>Post Condition</i>	Penambahan pada data PPDB

4.10.2 Use Case Scenario Melakukan Verifikasi Pendaftaran (UC-SIAS-2)

Use Case Scenario melakukan verifikasi pendaftaran merupakan proses untuk memeriksa formulir yang diajukan oleh calon siswa, dijelaskan pada Tabel 4.11.

Tabel 4.11 Use Case Scenario Melakukan Verifikasi Pendaftaran

<i>Use Case Name</i>	Melakukan Verifikasi Pendaftaran
<i>Brief Description</i>	Merupakan proses untuk memeriksa formulir yang diajukan oleh calon siswa untuk dipilih apakah data yang diisi telah memenuhi kriteria atau tidak.
<i>Actor</i>	Admin
<i>Pre-Condition</i>	Admin telah masuk kedalam sistem (otentikasi)
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Admin mengakses menu Verifikasi PPDB 2. Sistem menampilkan daftar data PPDB 3. Admin menekan tombol <i>Checklist</i> / Silang untuk memverifikasi calon siswa 4. Sistem mengubah status data Pendaftaran Peserta Didik Baru (PPDB) terpilih menjadi sudah diverifikasi. Sistem memberikan pengumuman kepada calon siswa

<i>Alternative Flows</i>	Alt1 Admin tidak menekan tombol <i>checklist</i> / silang pada menu verifikasi PPDB Jika pada langkah 3 Admin tidak menekan tombol <i>checklist</i> / silang pada menu verifikasi PPDB, maka sistem tidak mengubah status data PPDB.
<i>Post Condition</i>	Penambahan pada data Calon Siswa

4.10.3 Use Case Scenario Memasukkan Detail Nilai (UC-SIAS-3)

Use Case Scenario memasukkan detail nilai merupakan proses untuk memasukkan detail nilai ke dalam sistem, dijelaskan pada Tabel 4.12.

Tabel 4.12 Use Case Scenario Memasukkan Detail Nilai

<i>Use Case Name</i>	Memasukkan Detail Nilai
<i>Brief Description</i>	Merupakan proses untuk memasukkan detail nilai ke dalam sistem, sehingga dapat disimpan dan diakses
<i>Actor</i>	Guru
<i>Pre-Condition</i>	Guru telah masuk kedalam sistem (otentikasi)
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Guru mengakses menunilai 2. Sistem menampilkan halaman daftar kelas yang diajar 3. Guru memilih kelas yang diajar 4. Sistem menampilkan data siswa dengan rincian isian komponen nilai 5. Guru mengisi seluruh komponen nilai 6. Sistem menyimpan data komponen nilai
<i>Alternative Flows</i>	-
<i>Post Condition</i>	Penambahan pada data nilai

4.10.4 Use Case Scenario Memasukkan Detail Nilai Sikap (UC-SIAS-4)

Use Case Scenario memasukkan detail nilai merupakan proses untuk memasukkan detail sikap ke dalam sistem, dijelaskan pada Tabel 4.13.

Tabel 4.13 Use Case Scenario Memasukkan Detail Nilai Sikap

<i>Use Case Name</i>	Memasukkan Detail Nilai Sikap
<i>Brief Description</i>	Merupakan proses untuk memasukkan detail nilai sikap ke dalam sistem, sehingga dapat disimpan dan diakses
<i>Actor</i>	Wali Kelas
<i>Pre-Condition</i>	Wali Kelas telah masuk kedalam sistem (otentikasi)

<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Wali Kelas mengakses menu nilai sikap 2. Sistem menampilkan halaman daftar kelas yang diajar 3. Wali Kelas memilih kelas yang diajar 4. Sistem menampilkan data siswa dengan rincian isian komponen nilai sikap 5. Wali Kelas mengisi seluruh komponen nilai sikap 6. Sistem menyimpan data komponen nilai sikap
<i>Alternative Flows</i>	-
<i>Post Condition</i>	Penambahan pada data nilai sikap

4.10.5 Use Case Scenario Melihat Detail Nilai (UC-SIAS-5)

Use Case Scenario melihat detail nilai merupakan proses untuk menampilkan detail nilai beserta komponennya, dijelaskan pada Tabel 4.14.

Tabel 4.14 Use Case Scenario Melihat Detail Nilai

<i>Use Case Name</i>	Melihat Detail Nilai
<i>Brief Description</i>	Merupakan proses untuk menampilkan detail nilai beserta komponennya
<i>Actor</i>	Siswa
<i>Pre-Condition</i>	Siswa telah masuk kedalam sistem (otentikasi), dan nilai telah diisi oleh guru
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Siswa mengakses menu nilai 2. Sistem menampilkan halaman rincian nilai 3. Siswa melihat rincian nilai
<i>Alternative Flows</i>	-
<i>Post Condition</i>	-

4.10.6 Use Case Scenario Autentikasi (UC-SIAS-6)

Use Case Scenario autentikasi merupakan proses untuk masuk ke dalam sistem, dijelaskan pada Tabel 4.15.

Tabel 4.15 Use Case Scenario Autentikasi

<i>Use Case Name</i>	Autentikasi
<i>Brief Description</i>	Merupakan proses untuk masuk ke dalam sistem
<i>Actor</i>	<i>Guest</i>

<i>Pre-Condition</i>	<i>Guest</i> telah terdaftar sebagai Calon Siswa, Admin, Guru, Wali Kelas maupun Siswa
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. <i>Guest</i> mengakses menu login 2. Sistem menampilkan <i>form</i> login 3. <i>Guest</i> mengisi <i>form</i> login 4. <i>Guest</i> mengakses menu masuk
<i>Alternative Flows</i>	<p>Alt1 <i>Guest</i> mengisi <i>form</i> login dengan data guru dan Wali Kelas</p> <p>Jika pada langkah 1 <i>guest</i> mengisi <i>form</i> login dengan nip dan <i>password</i> guru, maka ketika mengakses menu masuk, sistem akan menampilkan halaman dasbor guru</p> <p>Alt2 <i>Guest</i> mengisi <i>form</i> login dengan data Admin Jika pada langkah 1, <i>guest</i> mengisi <i>form</i> login dengan nip dan <i>password</i> Admin, maka ketika mengakses menu masuk, sistem akan menampilkan halaman dasbor Admin</p> <p>Alt3 <i>Guest</i> mengisi <i>form</i> login dengan data Calon Siswa</p> <p>Jika pada langkah 1 <i>guest</i> mengisi <i>form</i> login dengan nisp dan <i>password</i> Calon Siswa, maka ketika mengakses menu masuk, sistem akan menampilkan halaman dasbor Calon Siswa</p> <p>Alt4 <i>Guest</i> mengisi <i>form</i> login dengan data Siswa</p> <p>Jika pada langkah 1 <i>guest</i> mengisi <i>form</i> login dengan nomor KK dan <i>password</i> siswa, maka ketika mengakses menu masuk, sistem akan menampilkan halaman dasbor Siswa</p> <p>Alt5 <i>Guest</i> mengisi <i>form</i> login dengan data yang salah</p> <p>Jika pada langkah 1 <i>guest</i> mengisi <i>form</i> login dengan nomor induk dan <i>password</i> yang salah, maka ketika mengakses menu masuk, sistem akan menampilkan halaman utama (<i>home</i>)</p>
<i>Post Condition</i>	-

4.10.7 Use Case Scenario Keluar (UC-SIAS-6)

Use Case Scenario keluar merupakan proses untuk keluar dari sistem, dijelaskan pada Tabel 4.15.

Tabel 4.16 Use Case Scenario Keluar

<i>Use Case Name</i>	Keluar
<i>Brief Description</i>	Merupakan proses untuk keluar dari sistem
<i>Actor</i>	Admin / Guru / Wali Kelas / Calon Siswa / Siswa
<i>Pre-Condition</i>	Admin / Guru / Wali Kelas / Calon Siswa/ Siswa telah masuk kedalam sistem (otentikasi)



<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Admin / Guru / Wali Kelas / Calon Siswa / Siswa mengakses menu logout 2. Sistem menampilkan halaman utama (<i>home</i>)
<i>Alternative Flows</i>	-
<i>Post Condition</i>	-



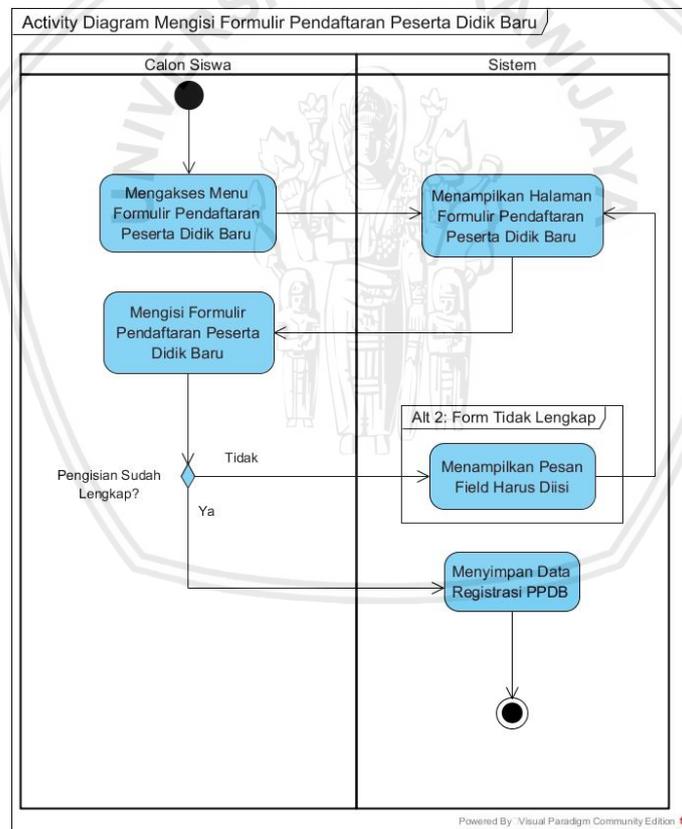
BAB 5 PERANCANGAN SISTEM

Pada bab ini menjelaskan mengenai tahapan perancangan sistem yang merupakan penerapan dari metode *waterfall* pada tahap *modelling*. Perancangan dibuat berdasarkan hasil dari tahap analisis kebutuhan. Bab perancangan berisi pembahasan mengenai hasil pemodelan dengan menggunakan UML sebagai notasi pemodelan. Hasil pemodelan yang dijelaskan pada bab ini di antaranya adalah perancangan dan pemodelan arsitektur sistem, *Activity Diagram*, *Sequence Diagram*, *Class Diagram*, *Physical Data Model*, dan perancangan antarmuka.

5.1 Activity Diagram

Activity Diagram merupakan diagram yang menggambarkan aktivitas yang terjadi pada suatu proses (Rumbaugh, Jacobson, & Booch, 2005). Berikut adalah *Activity Diagram* yang digunakan pada sistem informasi manajemen sekolah:

5.1.1 Mengisi Formulir Pendaftaran Peserta Didik Baru / PPDB

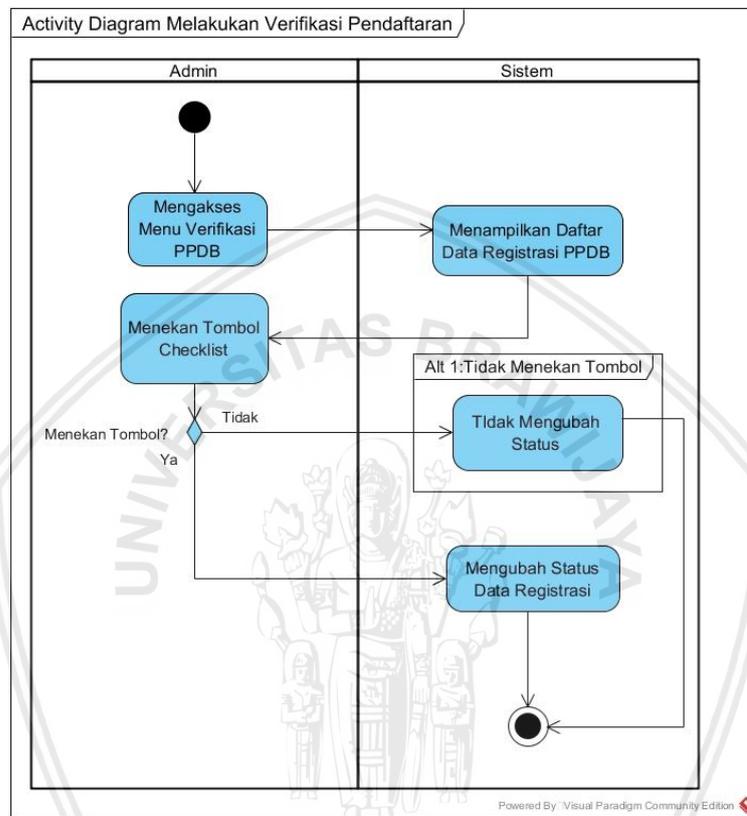


Gambar 5.1 Activity Diagram Mengisi Formulir Pendaftaran Peserta Didik Baru

Pada Gambar 5.1 terdapat penjelasan mengenai *Activity Diagram* mengisi formulir Pendaftaran Peserta Didik Baru. Diagram tersebut menggambarkan proses mengisi formulir Pendaftaran Peserta Didik Baru yang dilakukan oleh calon siswa untuk mendaftar sebagai calon siswa SMA Sejahtera 1 Depok. Proses dimulai dari calon siswa mengakses menu formulir Pendaftaran Peserta Didik Baru.

Kemudian sistem menampilkan halaman formulir Pendaftaran Peserta Didik Baru. Lalu calon siswa mengisi formulir Pendaftaran Peserta Didik Baru, jika calon siswa tidak mengisi secara lengkap, sistem akan menampilkan pesan field harus diisi. Jika sudah melengkapi formulir Pendaftaran Peserta Didik Baru, formulir Pendaftaran Peserta Didik Baru tersebut akan disimpan oleh sistem.

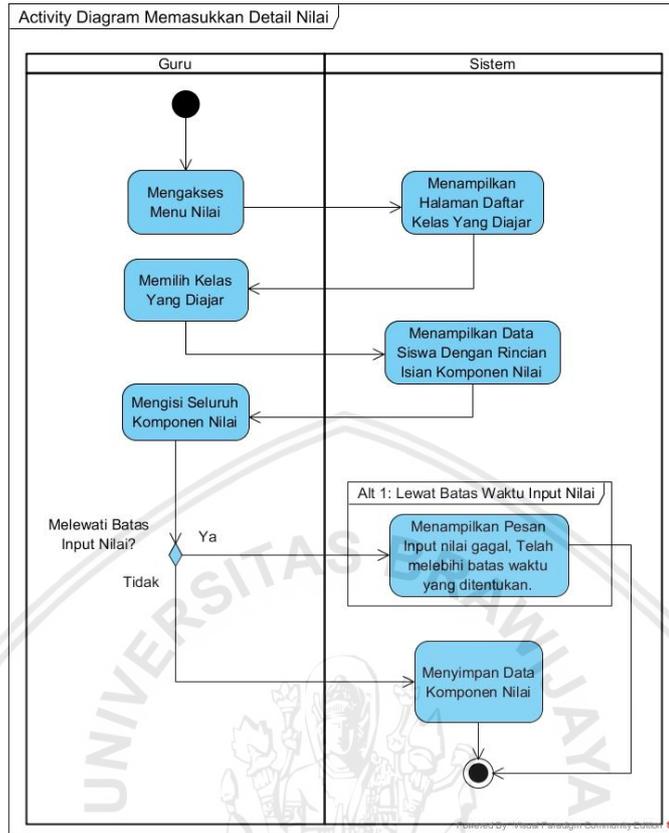
5.1.2 Melakukan Verifikasi Pendaftaran



Gambar 5.2 Activity Diagram Melakukan Verifikasi Pendaftaran

Pada Gambar 5.2 terdapat penjelasan mengenai *Activity Diagram* melakukan verifikasi pendaftaran. Diagram tersebut menggambarkan proses melakukan verifikasi pendaftaran PPDB. Proses dimulai dari Admin mengakses menu verifikasi PPDB. Setelah itu sistem menampilkan Daftar Data PPDB. Selanjutnya Admin menekan tombol *checklist* untuk memverifikasi pendaftaran PPDB, jika Admin tidak menekan tombol *checklist*, maka sistem tidak akan mengubah status PPDB. Jika Admin menekan tombol *checklist*, maka sistem akan mengubah status PPDB. Kemudian sistem memberikan pengumuman kepada calon siswa.

5.1.3 Memasukkan Detail Nilai



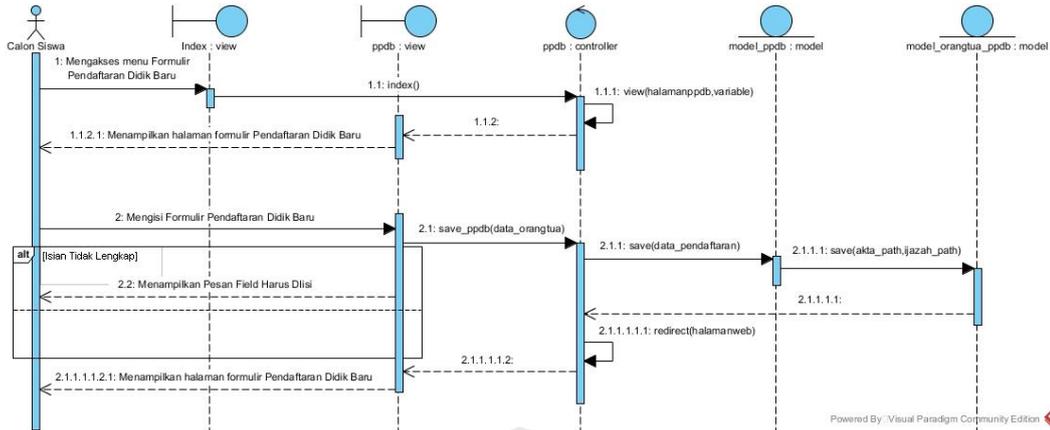
Gambar 5.3 Activity Diagram Memasukkan Detail Nilai

Pada Gambar 5.3 terdapat penjelasan mengenai *Activity Diagram* memasukkan detail nilai. Diagram tersebut menggambarkan proses memasukkan komponen nilai siswa. Proses dimulai guru mengakses menu nilai. Setelah itu sistem menampilkan halaman daftar kelas yang diajar. Selanjutnya guru memilih kelas yang diajar. Kemudian sistem menampilkan data siswa dengan rincian isian komponen nilai dan guru mengisi seluruh komponen nilai tersebut, jika Guru melewati batas input nilai, maka sistem akan menampilkan pesan input gagal, telah melebihi batas waktu yang ditentukan. Setelah sudah mengisi semua komponen nilai, sistem menyimpan data komponen nilai.

5.2 Sequence Diagram

Sequence Diagram merupakan diagram interaksi yang menekankan pada saat permintaan pesan yang menunjukkan serangkaian peranan dan pesan yang dikirim dan diterima oleh bagian yang ada pada peranan tersebut. Pesan dapat berupa sinyal atau panggilan terhadap suatu operasi. Notasi pesan untuk panggilan terhadap suatu operasi dapat dituliskan dalam sintaks UML atau sintaks bahasa pemrograman tertentu. Diagram sekuen digunakan untuk menggambarkan *dynamic view* dari sistem (Booch, et al., 2005).

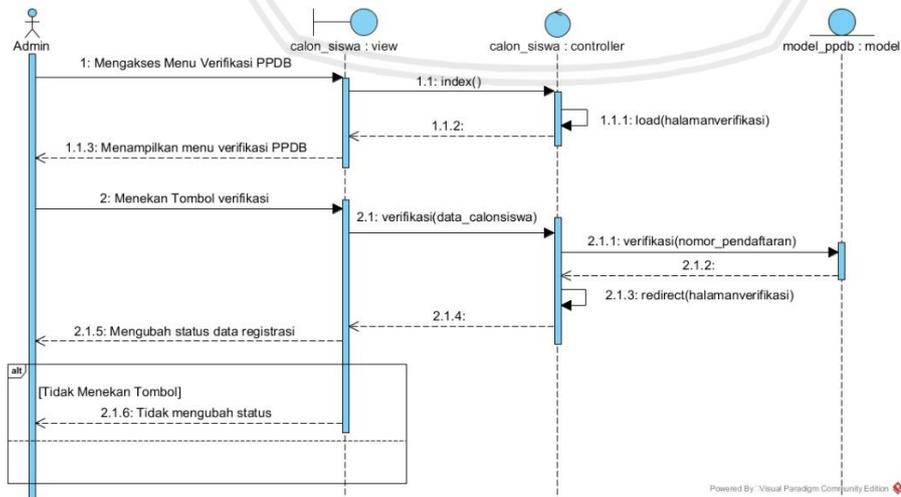
5.2.1 Mengisi Formulir Pendaftaran Peserta Didik Baru / PPDB



Gambar 5.4 Sequence Diagram Mengisi Formulir Pendaftaran Peserta Didik Baru

Gambar 5.4 menjelaskan alur mekanisme interaksi sekuensial antar aktor calon siswa, dengan *controller* *ppdb* untuk menangani permintaan *boundary* *ppdb*, *boundary* *ppdb* untuk menampilkan data dari seluruh formulir PPDB, *model* *ppdb* untuk mengelola data PPDB, serta *model* *model_orangtua_ppdb* untuk mengelola data orangtua calon siswa. Pada atribut *data_orangtua* berisi variabel nomor pendaftaran, nama orang tua, kewarganegaraan orang tua, tempat lahir orang tua, tanggal lahir orang tua, pendidikan orang tua, pekerjaan orang tua, penghasilan orang tua. Selanjutnya atribut *data_pendaftaran* berisi variabel nomor pendaftaran, tanggal mendaftar, nama pendaftar, jenis kelamin, nisn, kewarganegaraan, tempat lahir, tanggal lahir, agama, alamat pendaftar, tinggal bersama, jarak tempat tinggal, transportasi pendaftar, anak ke, jumlah saudara, status anak, bahasa sehari-hari, berat badan, tinggi badan, golongan darah, penyakit calonsiswa, asal sekolah, tahun lulus, alamat asal sekolah, status, file akta kelahiran, dan file ijazah smp.

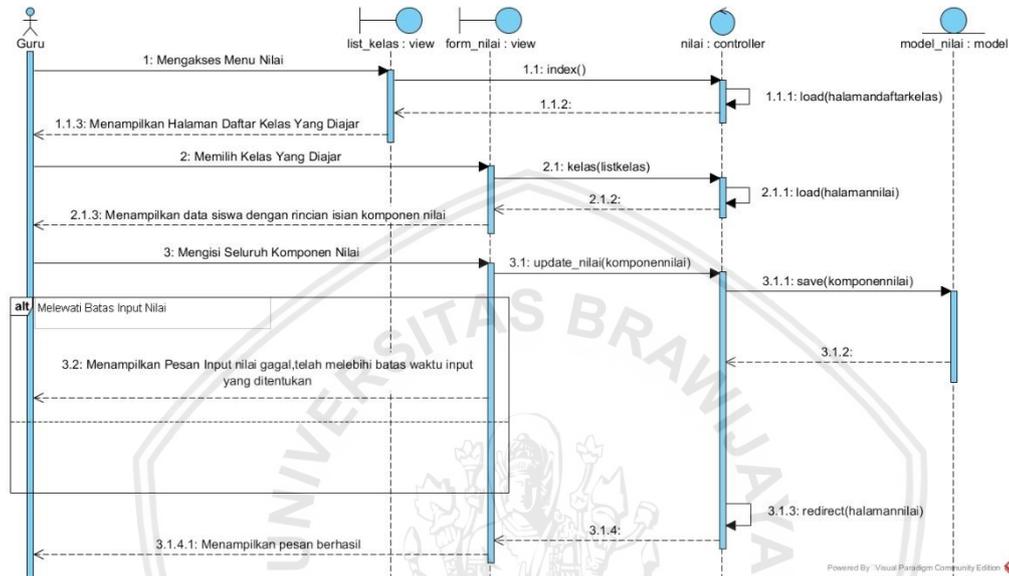
5.2.2 Melakukan Verifikasi Pendaftaran



Gambar 5.5 Sequence Diagram Melakukan Verifikasi Pendaftaran

Gambar 5.5 menjelaskan alur mekanisme interaksi sekuensial antar aktor admin, dengan *controller* calon_siswa untuk menangani permintaan *boundary* calon_siswa, *boundary* calon_siswa untuk menampilkan data dari pendaftar PPDB, serta *model* model_ppdb untuk mengelola data Pendaftaran Peserta Didik Baru (PPDB). Pada atribut data_calonsiswa berisi variabel nomor pendaftaran, nama calon siswa, dan status.

5.2.3 Memasukkan Detail Nilai



Gambar 5.6 Sequence Diagram Memasukkan Detail Nilai

Gambar 5.6 menjelaskan alur mekanisme interaksi sekuensial antar aktor guru, dengan *boundary* list_kelas untuk menampilkan data dari kelas yang diajar, dan *boundary* form_nilai untuk menampilkan data dari siswa yang diampu per mata pelajaran dalam satu kelas, *controller* nilai untuk menangani permintaan *boundary* form_nilai, serta *model* model_nilai untuk mengelola data orangtua calon siswa. Pada atribut komponennilai berisi variabel komponen nilai siswa yaitu nis, pengetahuan, keterampilan, dan sikap.

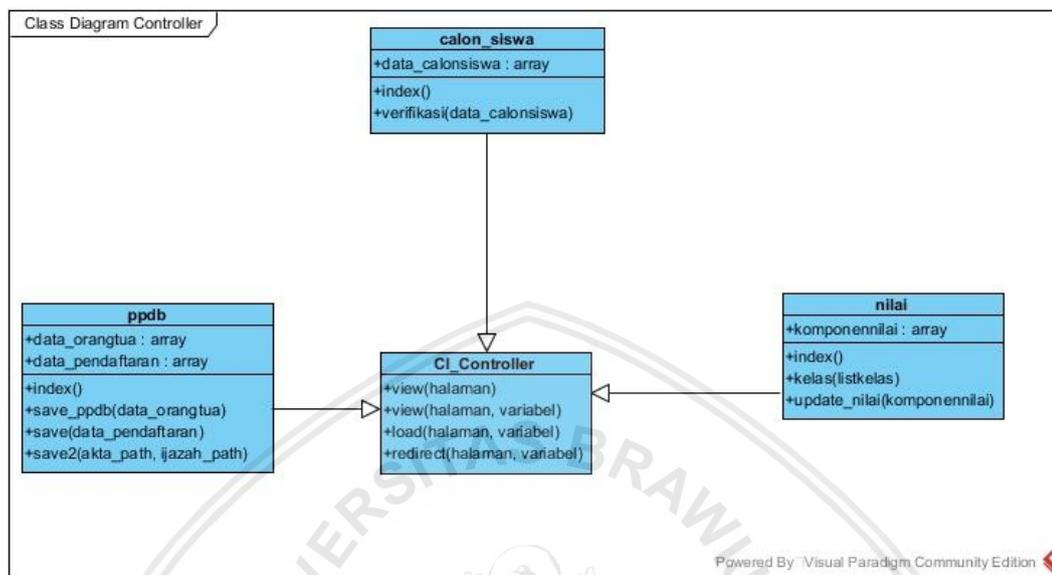
5.3 Class Diagram

Class diagram atau diagram kelas merupakan sebuah diagram yang digunakan untuk menggambarkan pandangan statis ataupun struktural terhadap sebuah sistem yang akan dikembangkan (Pressman, 2010). *Class Diagram* yang akan dimodelkan terbagi menjadi 2 yaitu *controller* dan *model*.

1. Class Diagram controller

Pada Gambar 5.7 menggambarkan class diagram controller yang memiliki 3 subkelas. Ketiga subkelas controller memiliki hubungan generalisasi terhadap kelas CI_Controller. Secara umum di class diagram controller menyediakan fungsi logika yang merepresentasikan kemampuan sistem untuk memenuhi kebutuhan pengguna. Fungsi yang merepresentasikan kemampuan sistem untuk

menampilkan formulir pendaftaran, pengumuman pendaftaran, cetak formulir pendaftaran, dan memasukkan data pendaftaran yang ditangani oleh fungsi pada subkelas `ppdb`. Fungsi pada subkelas `calon_siswa` menangani kemampuan sistem untuk mengelola data pendaftar `ppdb`.

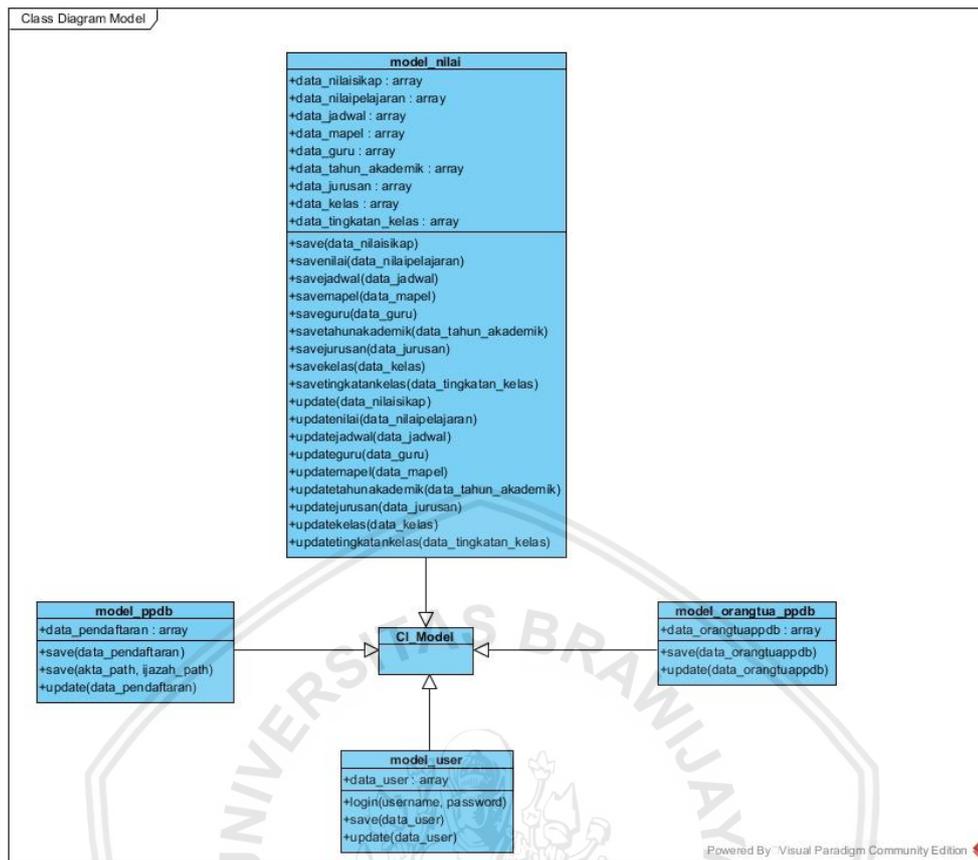


Gambar 5.7 Class Diagram Controller

Pada kegiatan memasukkan nilai detail nilai siswa yang ditangani oleh subkelas `nilai`. Sedangkan subkelas `nilai` merepresentasikan kemampuan sistem untuk mengelola data nilai siswa, dan menginput data nilai siswa yang dilakukan oleh aktor guru.

2. Class Diagram Model

Class Diagram model secara umum menyediakan fungsi untuk melakukan akses data yang tersimpan pada basis data. Pada Gambar 5.8 menggambarkan class diagram model yang memiliki 4 subkelas. Subkelas `model_nilai` memiliki fungsi akses data pada memasukkan detail nilai sikap. Fungsi akses data pada kegiatan Pendaftaran Peserta Didik Baru / PPDB ditangani oleh subkelas `model_model_ppdb` dan `model_orangtua_ppdb`. Salah satu fungsi pada subkelas `model_model_ppdb` maupun `model_orangtua_ppdb` adalah `save` yang menangani akses data untuk memasukkan data. Dan terakhir `model_user` memiliki fungsi menyimpan data user.



Gambar 5.8 Class Diagram Model

Tabel 5.1 Pendefinisian Atribut

No	Atribut	Variabel
1.	data_pendaftaran	<ul style="list-style-type: none"> • nomor pendaftaran • tanggal mendaftar • nama pendaftar • jenis kelamin • nisn • kewarganegaraan • tempat lahir • tanggal lahir • agama • alamat pendaftar • tinggal bersama • jarak tempat tinggal • transportasi pendaftar • anak ke • jumlah saudara • status anak • bahasa sehari-hari • berat badan • tinggi badan • golongan darah • penyakit calonsiswa • asal sekolah



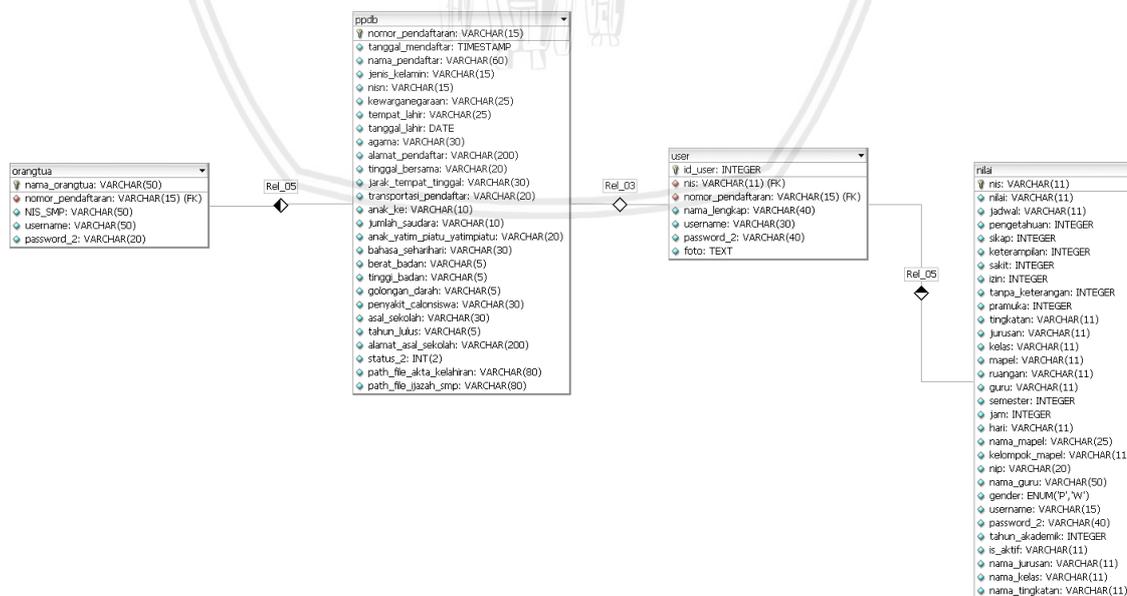
		<ul style="list-style-type: none"> • tahun lulus • alamat asal sekolah • status • lokasi file akta kelahiran • lokasi file ijazah smp
2.	data_orangtuappdb	<ul style="list-style-type: none"> • nama orang tua • kewarganegaraan orang tua • tempat lahir orang tua • tanggal lahir orang tua • pendidikan orang tua • pekerjaan orang tua • penghasilan orang tua
3.	data_nilaipeajaran	<ul style="list-style-type: none"> • nis • jadwal • pengetahuan • keterampilan • sikap
4.	data_nilaisikap	<ul style="list-style-type: none"> • nis • sakit • izin • tanpa keterangan • pramuka • imtaq
5.	data_jadwal	<ul style="list-style-type: none"> • nis • nip • tingkatan • jurusan • kelas • mapel • ruangan • tahun akademik • semester • jam • hari
6.	data_mapel	<ul style="list-style-type: none"> • mapel • nama mapel • kelompok mapel
7.	data_guru	<ul style="list-style-type: none"> • nip • mapel • nama guru • gender • username • password
8.	data_tahun_akademik	<ul style="list-style-type: none"> • tahun akademik • is aktif • semester
9.	data_jurusan	<ul style="list-style-type: none"> • jurusan • nama jurusan



10.	data_kelas	<ul style="list-style-type: none"> • kelas • tingkatan • jurusan • nama kelas
11.	data_tingkatan_kelas	<ul style="list-style-type: none"> • tingkatan • kelas • nama tingkatan

5.4 Physical Data Model

Physical data model merepresentasikan bagaimana *model* akan dibangun di *database*. Sebuah *physical data model* menampilkan struktur tabel yang terdiri dari nama kolom, tipe data kolom, *constraints* kolom, *primary key*, *foreign key*, dan relasi antara tabel (1keydata,2017). Pada gambar 5.9 menggambarkan *entity relationship diagram* berupa *physical data model* yang sebelum di normalisasi berisi 4 tabel. Tabel yang terdapat pada pemodelan diagram tersebut terdiri dari tabel nilai, user, ppdb, dan orangtua. Tabel yang pertama ialah tabel nilai, tabel tersebut menyimpan data nilai yang sudah diinput oleh guru dari tabel guru dengan *primary key* berupa nis. Kemudian terdapat tabel user, tabel tersebut menyimpan data user dengan *primary key* id_user. Tabel selanjutnya ialah tabel ppdb, tabel tersebut menyimpan data pendaftaran peserta didik baru yang diinput oleh pengguna calon siswa dengan *primary key* nomor_pendaftaran. Tabel yang terakhir ialah tabel orangtua, tabel tersebut menyimpan data orang tua dan inputan pendaftaran peserta didik baru dengan *primary key* nama_orangtua. Kemudian pada gambar 5.10 menggambarkan *entity relationship diagram* berupa *physical data model* yang sudah di normalisasi, berisi 12 tabel serta relasinya yang akan diimplementasikan pada sistem.



Gambar 5.9 Physical Data Model

identitas sebelum dapat mengakses informasi melalui sistem berdasarkan hak aksesnya. Antarmuka pengguna autentikasi merupakan bagian dari *use case* autentikasi. Komponen antarmuka autentikasi terdapat bagian yang berisikan logo sistem, judul sistem, tombol pilihan untuk masuk kedalam menu PPDB (Pendaftaran Peserta Didik Baru) dan SIAS (Sistem Informasi Akademik Sekolah).

Gambar 5.11 Antarmuka Pengguna autentikasi

5.5.2 Antarmuka Pengguna Mengisi Formulir Pendaftaran Peserta Didik Baru

Rancangan antarmuka pengguna mengisi formulir pendaftaran peserta didik baru merupakan tampilan sistem yang akan dilihat oleh pengguna, pada saat pengguna menggunakan sistem untuk mengisi formulir pendaftaran peserta didik baru. Antarmuka pengguna pendaftaran peserta didik baru merupakan bagian dari *use case* mengisi formulir pendaftaran peserta didik baru.

Antarmuka pengguna pendaftaran peserta didik baru yang ditujukan gambar 5.12 menunjukkan dua kelompok komponen antarmuka yang berbentuk persegi di sisi kiri dan kanan. Di sisi kiri terdapat *sidebar* yang memiliki komponen di antaranya judul sistem dan beberapa submenu, yaitu Formulir Pendaftaran, Pengumuman Pendaftaran, Cetak Formulir Pendaftaran. Sedangkan di sisi kanan terdapat dua kelompok komponen. Komponen pertama menunjukkan nomor pendaftaran calon siswa dan komponen tanggal. Sedangkan komponen kedua terdapat *dropdown* yang berisi formulir pendaftaran. Selain itu terdapat juga tombol untuk menjalankan fungsi pendaftaran peserta didik baru.

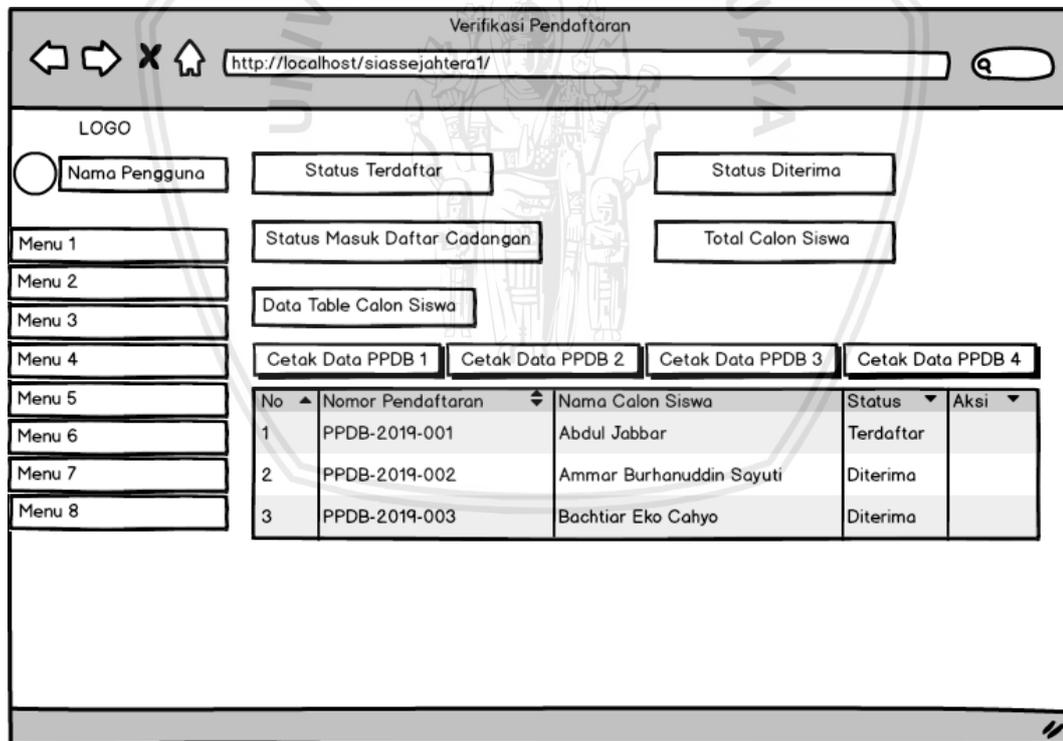
The image shows a web browser window titled "Formulir PPDB" with the URL "http://localhost/siassejahtera1/". The page content is for "SMA Sejahtera 1". On the left, there is a sidebar with three menu items: "Formulir Pendaftaran", "Pengumuman Pendaftaran", and "Cetak Formulir Pendaftaran". The main area contains a registration form with the following elements: "Nomor Pendaftaran" and "Tanggal" input fields; three expandable sections labeled "Data Pribadi", "Data Orang Tua / Wali", and "Upload Berkas", each with a plus sign icon; and a "Simpan" button at the bottom center.

Gambar 5.12 Antarmuka Pengguna Mengisi Pendaftaran Peserta Didik Baru

5.5.3 Antarmuka Pengguna Melakukan Verifikasi Pendaftaran

Rancangan Antarmuka pengguna melakukan verifikasi pendaftaran merupakan tampilan sistem yang akan dilihat oleh admin, pada saat admin menggunakan sistem untuk melakukan verifikasi pendaftaran. Antarmuka pengguna verifikasi pendaftaran merupakan bagian dari *use case* melakukan verifikasi pendaftaran.

Antarmuka pengguna melakukan verifikasi pendaftaran yang ditunjukkan gambar 5.13 menunjukkan dua kelompok komponen antarmuka di sisi kiri dan kanan. Pada sisi kiri yaitu berisi nama sistem informasi dan logo sistem di bagian atas, dan beberapa menu sistem. Sedangkan di sisi kanan terdapat 2 kelompok komponen. Komponen pertama menunjukkan status terdaftar, status diterima, status masuk daftar cadangan, dan total calon siswa. Selanjutnya komponen kedua terdapat tombol cetak data PPDB diterima, cetak data PPDB masuk daftar cadangan, cetak data PPDB terdaftar, dan cetak data PPDB total calon siswa. Kemudian terdapat tabel, tabel tersebut menunjukkan beberapa kolom, yaitu nomor pendaftaran, nama calon siswa, status, dan aksi. Pada kolom aksi terdapat beberapa tombol, yaitu tombol diterima, tombol masuk daftar cadangan, dan tombol delete.

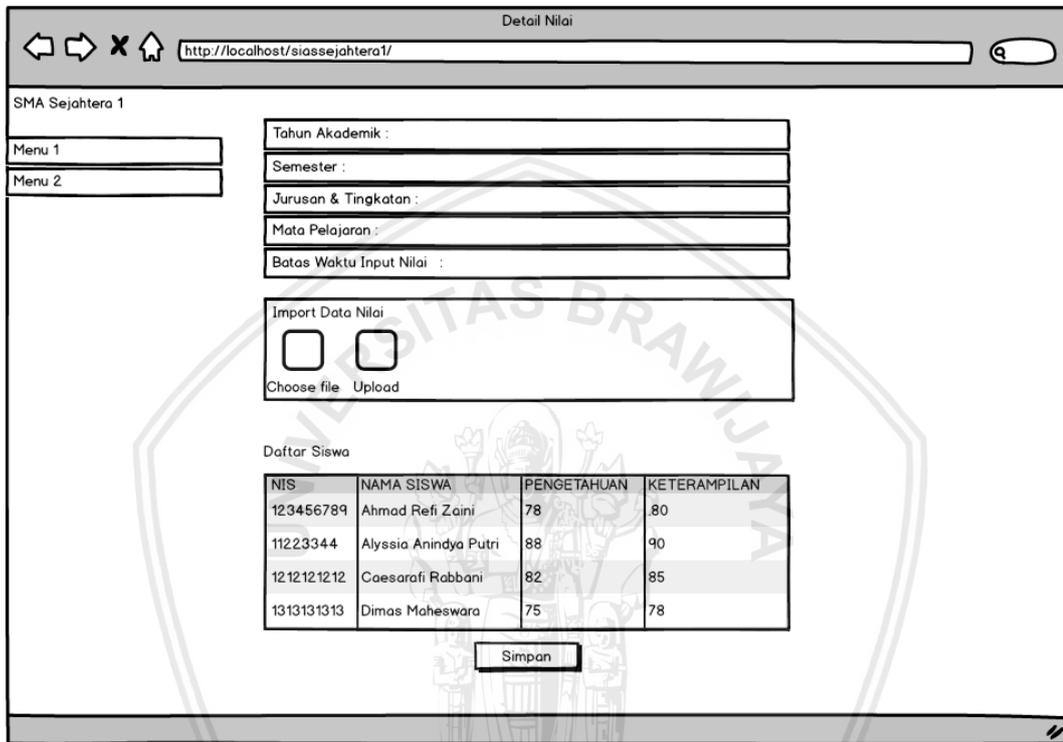


Gambar 5.13 Antarmuka Pengguna Melakukan Verifikasi Pendaftaran

5.5.4 Antarmuka Pengguna Memasukkan Detail Nilai

Rancangan antarmuka pengguna memasukkan detail nilai merupakan tampilan sistem yang akan dilihat oleh guru, pada saat guru menggunakan sistem untuk memasukkan detail nilai. Antarmuka pengguna memasukkan detail nilai merupakan bagian dari *use case* memasukkan detail nilai.

Antarmuka pengguna memasukkan detail nilai yang ditujukan pada gambar 5.14 menunjukkan dua kelompok komponen antarmuka di sisi kiri dan kanan. Di sisi kiri terdapat *sidebar* yang memiliki komponen di antaranya judul sistem, dan beberapa submenu. Sedangkan di sisi kanan terdapat dua kelompok komponen. Komponen pertama menunjukkan tahun akademik, semester, jurusan & tingkatan, dan mata pelajaran. Sedangkan komponen kedua terdapat tabel siswa yang berisi NIS, nama siswa, dan komponen penilaian (pengetahuan & keterampilan).



Gambar 5.14 Antarmuka Pengguna Memasukkan Detail Nilai

BAB 6 IMPLEMENTASI

Bab implementasi berisi penjelasan mengenai kebutuhan implementasi dan hasil implementasi sistem informasi yang dikembangkan dalam penelitian. Bab implementasi menjelaskan mengenai spesifikasi sistem, batasan implementasi, struktur artefak sistem, algoritme, beberapa kode program, dan beberapa gambar antarmuka pengguna sistem informasi yang dikembangkan. Implementasi dilakukan berdasarkan hasil perancangan yang telah dijelaskan pada bab sebelumnya.

6.1 Spesifikasi Sistem

Subbab spesifikasi sistem menjelaskan spesifikasi perangkat keras dan perangkat lunak yang digunakan selama proses pengembangan sistem informasi.

6.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras merupakan pembahasan dari komponen perangkat keras yang digunakan untuk mengembangkan sistem informasi. Komponen spesifikasi perangkat keras yang digunakan tersebut didefinisikan pada table 6.1.

Tabel 6.1 Spesifikasi Perangkat Keras

Komponen	Spesifikasi
Model <i>Laptop</i>	Lenovo Y50
<i>Processor</i>	Intel® Core™ i7-4710HQ @ 2.50GHz
<i>Graphic Card</i>	NVIDIA GeForce GTX 860M
<i>Random Access Memory (RAM)</i>	16GB DDR3
<i>Hardisk</i>	1TB
<i>Resolution</i>	1920x1080 <i>pixels</i>

Tabel 6.1 mendeskripsikan kompoen spesifikasi perangkat keras yang digunakan untuk mengembangkan sistem informasi. Perangkat tersebut terdiri dari laptop Lenovo Y50, dimana didalam perangkat tersebut terdiri dari komponen *processor* berupa Intel® Core™ i7-4710HQ dengan kecepatan 2.50GHz, lalu *graphic card* berupa NVIDIA Geforce GTX 860M, kemudian RAM sebesar 16GB, selanjutnya dengan kapasitas *hardisk* sebesar 1TB, dan *Resolution* 1920x1080 *pixels*.

6.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak merupakan pembahasan dari komponen perangkat lunak yang digunakan untuk mengembangkan sistem informasi. Komponen spesifikasi perangkat lunak yang digunakan tersebut didefinisikan pada tabel 6.2.

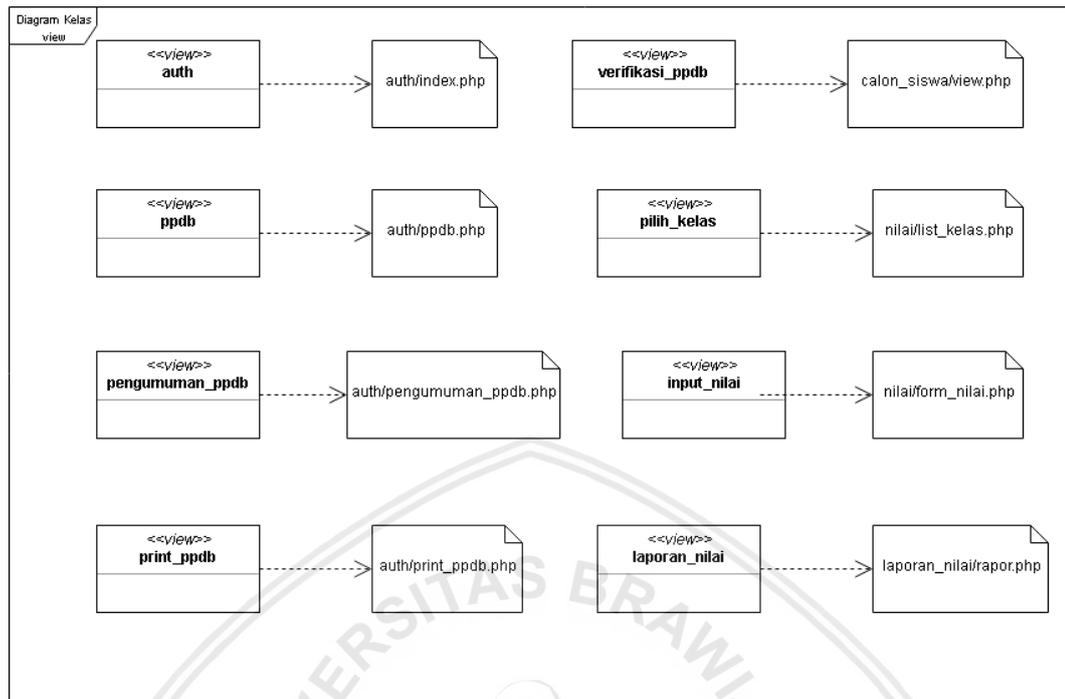
Tabel 6.2 Spesifikasi Perangkat Lunak

Komponen	Spesifikasi
Sistem Operasi	Windows 10 Home Single Language
Bahasa Pemrograman	PHP v7.0.9, HTML, Javascript, dan ajax
<i>Tools Development</i>	Atom,phpMyAdmin, Visual paradigm, DBDesigner 4, dan Balsamiq Mockups 3
<i>Tools Server</i>	XAMPP v3.2.2
<i>Database Management System (DBMS)</i>	MySQL v15.1
<i>Web Browser</i>	Google Chrome v73.0.3683.86

Tabel 6.2 mendeskripsikan komponen spesifikasi perangkat lunak yang digunakan untuk mengembangkan sistem informasi. Sistem operasi yang digunakan pada saat proses pengembangan ialah Windows 10 Home Single Language. Lalu bahasa pemrograman yang dipilih ialah PHP, HTML, Javascript, dan ajax. Bahasa pemrograman PHP, HTML, Javascript, dan ajax dipilih karena bersifat *open source* serta mudah untuk digunakan ditambah banyaknya komunitas atau forum yang membahas keempat bahasa pemrograman tersebut apabila terjadi kendala saat proses pengkodean. Dan juga bahasa pemrograman PHP menempati posisi pertama sebagai bahasa paling banyak yang digunakan dalam pemrograman web (W3Techs,2018). Kemudian *tools development* yang digunakan pada pengembangan ini ialah atom, phpMyAdmin, Visual paradigm, DBDesigner 4, dan Balsamiq Mockups 3. *Tools* tersebut dipilih karena menyediakan versi *open source*, kecuali balsamiq mockups 3 yang menyediakan *trial* selama 30 hari. Selanjutnya *tools server* yang digunakan ialah XAMPP. XAMPP dipilih karena kepraktisannya menyediakan server serta DBMS secara utuh yaitu MySQL, serta aplikasi web untuk mengelola MySQL tersebut bernama phpMyAdmin. Dan yang terakhir *web browser* yang digunakan sebagai *platform* untuk menajalnkkan sistem ialah Google Chrome. Google Chrome dipilih karena merupakan *web browser* paling banyak yang digunakan pada tahun 2019 (GlobalStats, 2019).

6.2 Struktur Artefak Sistem

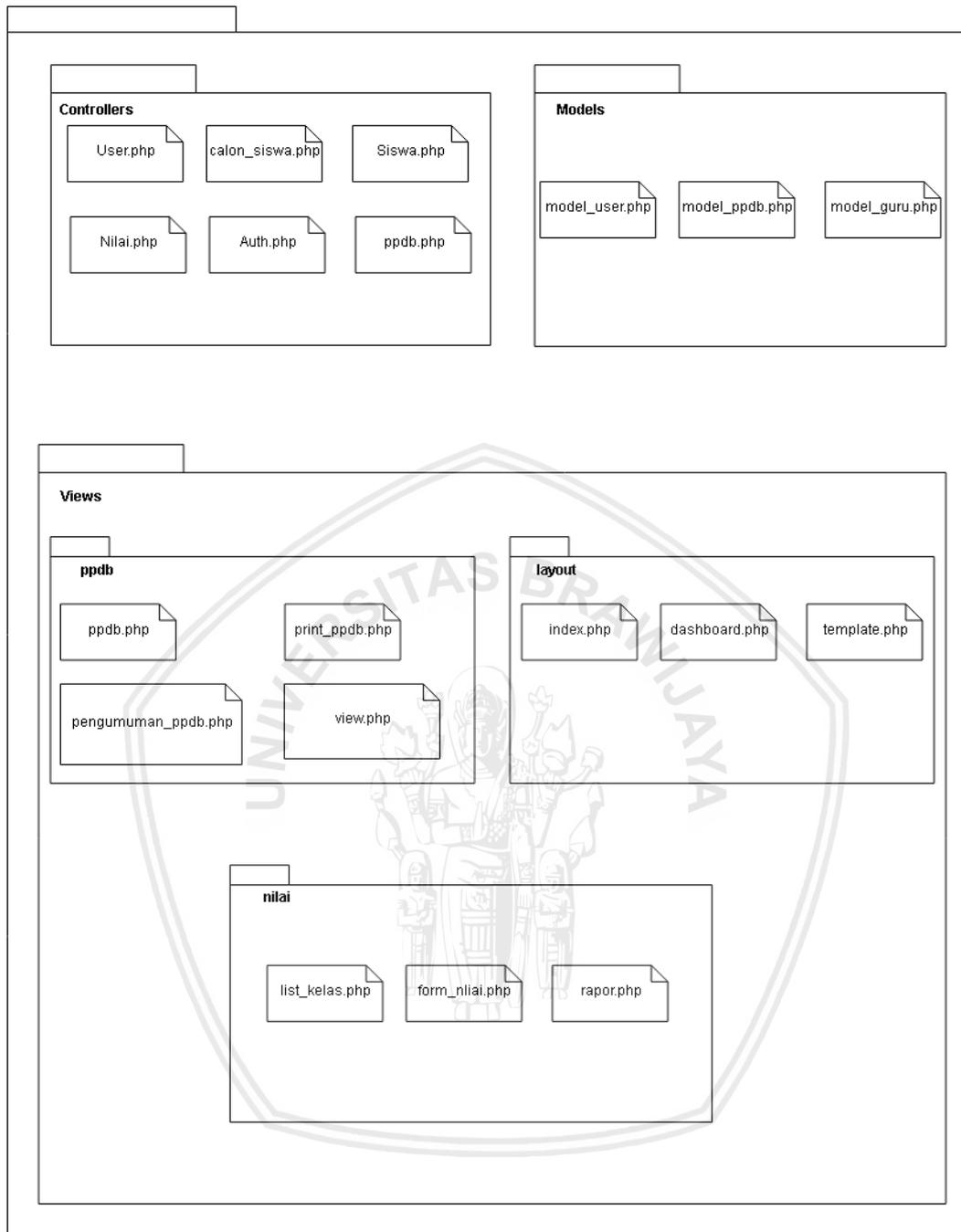
Struktur artefak sistem memberikan informasi mengenai artefak sistem dalam sistem informasi yang dikembangkan. Informasi mengenai struktur artefak sistem didapatkan dari implementasi *class diagram view* yang digambarkan pada Gambar 6.1. Pada Gambar 6.1 menggambarkan implementasi *class diagram view* yang dimodelkan dengan hubungan depedensi terhadap file bertipe php.



Gambar 6.1 Implementasi Class Diagram untuk View

Struktur artefak sistem pada Gambar 6.2 menunjukkan terdapat 3 *package* yang terdiri dari *package* controllers, models, dan views. *Package* controllers berisikan implementasi dari *class diagram controller* yang terdiri dari file `User.php`, `calon_siswa.php`, `Siswa.php`, `Nilai.php`, `Auth.php`, dan `ppdb.php`.

Package models berisikan implementasi dari *class diagram model* yang terdiri dari file `model_user.php`, `model_ppdb.php`, dan `model_guru.php`. Sedangkan *package* views berisikan implementasi dari antarmuka pengguna. *Package* views terdapat 3 *package* yang terdiri dari *package* ppdb, *package* layout, dan *package* nilai. Pada *package* ppdb berisikan file `ppdb.php`, `print_ppdb.php`, `pengumuman_ppdb.php`, dan `view.php`. Lalu pada *package* layout berisikan file `index.php`, `dashboard.php`, dan `template.php`. Dan terakhir pada *package* nilai berisikan file `list_kelas.php`, `form_nilai.php`, dan `rapor.php`.



Gambar 6.2 Struktur Artefak Sistem

6.3 Implementasi Algoritme

Subbab implementasi algoritme berisi beberapa kode program. Beberapa hasil implementasi algoritme terdiri dari kode program mengisi formulir pendaftaran peserta didik baru, melakukan verifikasi pendaftaran, memasukkan detail nilai, dan melihat detail nilai.

6.3.1 Mengisi Formulir Pendaftaran Peserta Didik Baru

Kode program mengisi formulir pendaftaran peserta didik baru merupakan bagian dari *use case* mengisi formulir pendaftaran peserta didik baru yang disediakan oleh sistem. Tabel 6.3 merupakan implementasi algoritme yang terdapat pada *class* controller ppdb.

Tabel 6.3 Implementasi Algoritma Mengisi Formulir Pendaftaran Didik Baru

No	Kode Program
1	<?php
2	/**
3	*
4	*/
5	class ppdb extends CI_Controller
6	{
7	function __construct()
8	{
9	parent::__construct();
10	\$this->load->model('model_ppdb');
11	\$this->load->model('model_orangtua_ppdb');
12	\$this->load->model('model_wali_ppdb');
13	}
14	function index ()
15	{
16	\$data['nomor_pendaftaran'] = \$this->
17	generate_nomor_pendaftaran();
18	\$this->load->view('auth/ppdb', \$data);
19	}
20	function print_ppdb()
21	{
22	\$this->load->view('auth/print_ppdb');
23	}
24	function save_ppdb(){
25	\$config = array();
26	\$config['file_name'] = 'akta_'.\$this->input->
27	post('nomor_pendaftaran', TRUE).'jpg';
28	\$config['allowed_types'] = 'jpg';
29	\$config['upload_path'] = './uploads/akta/';
30	\$this->load->library('upload', \$config, 'akta');
	\$this->akta->initialize(\$config);
	//proses upload

```
31     $this->akta->do_upload('akta_kelahiran');
32     $upload1 = $this->akta->data();
33     $akta_path = $upload1['full_path'];
34     $config = array();
35     $config['file_name'] = 'ijazah_smp_'. $this->
>input->post('nomor_pendaftaran', TRUE).' .jpg';
36     $config['allowed_types'] = 'jpg';
37     $config['upload_path'] =
'./uploads/ijazah_smp/';
38     $this->load->library('upload', $config,
39 'ijazah_smp');
40     $this->ijazah_smp->initialize($config);
41     //proses upload
42     $this->ijazah_smp->do_upload('ijazah_smp');
43     $upload2 = $this->ijazah_smp->data();
44     $ijazah_path = $upload2['full_path'];
45     $this->model_ppdb->save($akta_path, $ijazah_path);
46     $this->model_orangtua_ppdb->save();
47     redirect('http://localhost/siassejahtera1/ppdb');
48 }
49 function pengumuman_ppdb()
50 {
51     $data['code_page'] = "Halaman Pengumuman";
52     $this->load->view('auth/pengumuman_ppdb', $data);
53 }
54 function generate_nomor_pendaftaran() {
55     // $nomor = count($this->m_pendaftar-
>get_last_number())+1;
56     $last_number = $this->model_ppdb->get_last_number();
57     $number = $last_number+1;
58     $nomor = str_pad($number, 3, '0', STR_PAD_LEFT);
59     $kode_pendaftaran = "PPDB";
60     $tahun = date("Y");
61     $nomor_pendaftaran = "$kode_pendaftaran-" . "$tahun-"
. "$nomor";
62     return $nomor_pendaftaran;
63     // $cek = $this->m_pendaftar-
>get_all_id_pendaftar($nomor_pendaftaran);
64     // if($cek != null){
65     //     $this->generate_nomor_pendaftaran();
66 }
```

```
64 // }else{
65 //     return $nomor_pendaftaran;
66 // }
67 }
68 function show_pengumuman()
69 {
70     $data['code_page'] = "Halaman Pengumuman";
71     $this->template->view('auth/pengumuman_ppdb',$data);
72 }
73 function hasil_pendaftaran()
74 {
75     $no = $this->input->post("nomor");
76     $tgl = $this->input->post("tgl");
77     $data = $this->model_ppdb->get_pengumuman($no,$tgl);
78     echo json_encode($data);
79 }
80 function print_one()
81 {
82     $no = $this->input->post("nomor_pendaftar_pengumuman");
83     $tgl = $this->input->post("tanggal_lahir_pengumuman");
84     $calonsiswa = $this->model_ppdb->get_print_ppdb($no);
85     $orangtuasiswa = $this->model_orangtua_ppdb->get_print_ppdb($no);
86     $dokumen = base_url('\application\third_party\data_siswa.docx');
87     include_once APPPATH.'third_party/docxtemplate.class.php';
88     //siswa
89     $docx = new DOCXTemplate('C:\xampp\htdocs\siassejahtera\application\third_party\data_siswa.docx');
90     $docx->set('nomor_pendaftaran', $calonsiswa['data']->nomor_pendaftaran);
91     $docx->set('nama_pendaftar', $calonsiswa['data']->nama_pendaftar);
92     $docx->set('nisn', $calonsiswa['data']->nisn);
```

```
96     $docx->set('tempat_lahir',          $calonsiswa['data']-
97     >tempat_lahir);
98     $docx->set('tanggal_lahir',          date("d-m-Y",
99     strtotime($calonsiswa['data']->tanggal_lahir));
100     $docx->set('jenis_kelamin',          $calonsiswa['data']-
101     >jenis_kelamin);
102     $docx->set('agama', $calonsiswa['data']->agama);
103     $docx->set('kewarganegaraan',       $calonsiswa['data']-
104     >kewarganegaraan);
105     $docx->set('anak_ke', $calonsiswa['data']->anak_ke);
106     $docx->set('jumlah_saudara',        $calonsiswa['data']-
107     >jumlah_saudara);
108     $docx->set('anak_yatim_piatu_yatimpiatu',
109     $calonsiswa['data']->anak_yatim_piatu_yatimpiatu);
110     $docx->set('bahasa_seharihari',     $calonsiswa['data']-
111     >bahasa_seharihari);
112     $docx->set('alamat_pendaftar',      $calonsiswa['data']-
113     >alamat_pendaftar);
114     $docx->set('tinggal_bersama',       $calonsiswa['data']-
115     >tinggal_bersama);
116     $docx->set('jarak_tempat_tinggal',  $calonsiswa['data']->jarak_tempat_tinggal);
117     $docx->set('transportasi_pendaftar',
118     $calonsiswa['data']->transportasi_pendaftar);
119     $docx->set('berat_badan',           $calonsiswa['data']-
120     >berat_badan);
121     $docx->set('tinggi_badan',          $calonsiswa['data']-
122     >tinggi_badan);
123     $docx->set('golongan_darah',        $calonsiswa['data']-
124     >golongan_darah);
125     $docx->set('penyakit_calonsiswa',   $calonsiswa['data']-
126     >penyakit_calonsiswa);
127     $docx->set('asal_sekolah',          $calonsiswa['data']-
128     >asal_sekolah);
129     $docx->set('tahun_lulus',           $calonsiswa['data']-
130     >tahun_lulus);
131     $docx->set('alamat_asal_sekolah',   $calonsiswa['data']-
132     >alamat_asal_sekolah);
133     //orangtua
134     $docx->set('nama_orangtua',          $orangtuasiswa['data']-
135     >nama_orangtua);
136     $docx->set('tempat_lahir_orangtua',
137     $orangtuasiswa['data']->tempat_lahir_orangtua);
138     $docx->set('tanggal_lahir_orangtua',
139     $orangtuasiswa['data']->tanggal_lahir_orangtua);
```

```

118     $docx->set('kewarganegaraan_orangtua',
119     $orangtuasiswa['data']->kewarganegaraan_orangtua);
120     $docx->set('pendidikan_orangtua',
121     $orangtuasiswa['data']->pendidikan_orangtua);
122     $docx->set('pekerjaan_orangtua',
123     $orangtuasiswa['data']->pekerjaan_orangtua);
124     $docx->set('penghasilan_orangtua',
125     $orangtuasiswa['data']->penghasilan_orangtua);
126     $nomor_pendaftaran      =      $calonsiswa['data']-
127     >nomor_pendaftaran;
128     $nomor_pendaftaran      =      $orangtuasiswa['data']-
129     >nomor_pendaftaran;
130     $nama_file = 'data-'. $nomor_pendaftaran.'.docx';
131     $docx->saveAs($nama_file);
132     header("Content-Type:application/msword");
133     header("Content-Disposition:
134     attachment;filename=".$nama_file);
135     readfile($nama_file);
136     exit();
137 }
138 }

```

6.3.2 Melakukan Verifikasi Pendaftaran

Kode program melakukan verifikasi pendaftaran merupakan bagian dari *use case* melakukan verifikasi pendaftaran yang disediakan oleh sistem. Tabel 6.4 merupakan implementasi algoritme yang terdapat pada *class controller* calon_siswa.

Tabel 6.4 Implementasi Algoritme Melakukan Verifikasi Pendaftaran

No	Kode Program
1	<?php
2	
3	class calon_siswa extends CI_Controller
4	{
5	
6	function __construct()
7	{
8	parent::__construct();
9	//checkAksesModule();



```
10     $this->load->library('ssp');
11     $this->load->model('model_ppdb');
12     }
13     function data()
14     {
15
16         // nama table
17         $table      = 'tbl_ppdb';
18         // nama PK
19         $primaryKey = 'nomor_pendaftaran';
20         // list field yang mau ditampilkan
21         $columns    = array(
22             //tabel db(kolom di database) => dt(nama
23             datatable di view)
24             array('db' => 'nomor_pendaftaran', 'dt' =>
25             'nomor_pendaftaran'),
26             array('db' => 'nama_pendaftar', 'dt' =>
27             'nama_pendaftar'),
28             array('db' => 'status', 'dt' => 'status'),
29             //untuk menampilkan aksi(edit/delete dengan
30             parameter id guru)
31             array(
32                 'db' => 'nomor_pendaftaran',
33                 'dt' => 'aksi',
34                 'formatter' => function($d) {
35                     return
36                     anchor('calon_siswa/verifikasi/'. $d, '<i class="fa fa-
37                     check"></i>', 'class="btn btn-xs btn-primary" data-
38                     placement="top" title="Diterima"').'
39                     '.anchor('#'. $d, '<i class="fa fa-
40                     eye"></i>', 'class="btn btn-xs btn-primary" data-
41                     placement="top" title="Melihat Calon Siswa"').'
42
43                     '.anchor('calon_siswa/masuk_daftar_cadangan/'. $d, '<i
44                     class="fa fa-question"></i>', 'class="btn btn-xs btn-
45                     primary" data-placement="top" title="Masuk Daftar
46                     Cadangan"').'
47
48                     '.anchor('#'. $d, '<i class="fa fa-
49                     times fa fa-white"></i>', 'class="btn btn-xs btn-danger"
50                     data-placement="top" title="Delete"');
51                 }
52             )
53         );
54     };
```

```
35     $sql_details = array(  
36         'user' => $this->db->username,  
37         'pass' => $this->db->password,  
38         'db'   => $this->db->database,  
39         'host' => $this->db->hostname  
40     );  
41  
42     echo json_encode(  
43         SSP::simple($_GET, $sql_details, $table,  
44             $primaryKey, $columns)  
45     );  
46  
47     function index()  
48     {  
49         $datappdb['data'] = $this->model_ppdb->  
50         >get_data_ppdb();  
51  
52         $qterdaftar = "SELECT * FROM tbl_ppdb JOIN tbl_status  
53         ON tbl_ppdb.status = tbl_status.id_status WHERE  
54         nama_status = 'TERDAFTAR';"  
55         $datappdb['terdaftar'] = count($this->  
56         >db->query($qterdaftar)->result());  
57  
58         $qditerima = "SELECT * FROM tbl_ppdb JOIN  
59         tbl_status  
60         ON tbl_ppdb.status = tbl_status.id_status WHERE  
61         nama_status = 'DITERIMA';"  
62         $datappdb['diterima'] = count($this->db->  
63         >query($qditerima)->result());  
64  
65         $qdaftarcadangan = "SELECT * FROM  
66         tbl_ppdb JOIN tbl_status  
67         ON tbl_ppdb.status = tbl_status.id_status WHERE  
68         nama_status = 'MASUK DAFTAR CADANGAN';"  
69         $datappdb['masukdaftarcadangan'] =  
70         count($this->db->query($qdaftarcadangan)->result());  
71  
72         $qtotalcalonsiswa = 'SELECT * FROM  
73         tbl_ppdb';  
74         $datappdb['totalcalonsiswa'] =  
75         count($this->db->query($qtotalcalonsiswa)->result());  
76  
77         $this->template->load('template',  
78         'calon_siswa/view', $datappdb);
```

```
60     }
61
62     function verifikasi()
63     {
64         $nomor_pendaftaran = $this->uri->segment(3);
65         $this->model_ppdb->verifikasi($nomor_pendaftaran);
66         $this->session->set_flashdata('msg', 'Calon Siswa
67         Diterima!');
68         redirect('calon_siswa');
69     }
70     function masuk_daftar_cadangan()
71     {
72         $nomor_pendaftaran = $this->uri->segment(3);
73         $this->model_ppdb->masuk_daftar_cadangan($nomor_pendaftaran);
74         $this->session->set_flashdata('msg', 'Calon Siswa
75         Masuk Daftar Cadangan!');
76         redirect('calon_siswa');
77     }
78     function export_diterima()
79     {
80         $siswa = $this->db->get_where('tbl_ppdb',
81         array('status' => 2))->result();
82
83         $this->load->library('PHPExcel');
84         $objPHPExcel = $objPHPExcel =
85         PHPEXcel_IOFactory::load("laporanppdb/template_ppdb_diteri
86         ma.xlsx");
87         $objPHPExcel->getActiveSheet()-
88         >setCellValue('A2', 'TAHUN AJARAN
89         '.get_tahun_akademik('tahun_akademik'));
90
91         $no = 7;
92         $counter = 0;
93
94         foreach ($siswa as $row){
```

```

86         $objPHPExcel->getActiveSheet()-
>setCellValue('A'.$no, ++$counter.'.');
87         $objPHPExcel->getActiveSheet()-
>setCellValue('B'.$no, $row->nomor_pendaftaran);
88         $objPHPExcel->getActiveSheet()-
>setCellValue('C'.$no, $row->nama_pendaftar);
89         $objPHPExcel->getActiveSheet()-
>setCellValue('D'.$no, $row->asal_sekolah);
90         $no++;
91     }
92     $objWriter =
PHPExcel_IOFactory::createWriter($objPHPExcel,
'Excel2007');
93     $objWriter->save("laporan_ppdb_diterima.xlsx");
94     $this->load->helper('download');
95     force_download('laporan_ppdb_diterima.xlsx',
NULL);
96 }
97
function export_terdaftar()
{
98     $siswa = $this->db->get_where('tbl_ppdb',
array('status' => 1))->result();
99
100    $this->load->library('PHPExcel');
101    $objPHPExcel = $objPHPExcel =
PHPExcel_IOFactory::load("laporanppdb/template_ppdb_terdaf
tar.xlsx");
102    $objPHPExcel->getActiveSheet()-
>setCellValue('A2', 'TAHUN AJARAN
'.get_tahun_akademik('tahun_akademik'));
103
104    $no = 7;
105    $counter = 0;
106
    foreach ($siswa as $row){
107        $objPHPExcel->getActiveSheet()-
>setCellValue('A'.$no, ++$counter.'.');
108        $objPHPExcel->getActiveSheet()-
>setCellValue('B'.$no, $row->nomor_pendaftaran);

```

```

1207         $objPHPExcel->getActiveSheet()-
>setCellValue('C'.$no, $row->nama_pendaftar);
        $objPHPExcel->getActiveSheet()-
1208 >setCellValue('D'.$no, $row->asal_sekolah);
        $no++;
1209     }
1210     $objWriter =
1211     PHPExcel_IOFactory::createWriter($objPHPExcel,
'Excel2007');
        $objWriter->save("laporan_ppdb_terdaftar.xlsx");
1212     $this->load->helper('download');
1213     force_download('laporan_ppdb_terdaftar.xlsx',
NULL);
    }
1214
    function export_masukdaftarcadangan()
1215     {
1216
        $siswa = $this->db->get_where('tbl_ppdb',
1217     array('status' => 3))->result();
1218
        $this->load->library('PHPExcel');
        $objPHPExcel = $objPHPExcel =
1219     PHPExcel_IOFactory::load("laporanppdb/template_ppdb_masukd
aftarcadangan.xlsx");
        $objPHPExcel->getActiveSheet()-
1220 >setCellValue('A2', 'TAHUN AJARAN
'.get_tahun_akademik('tahun_akademik'));
1221
        $no = 7;
1222         $counter = 0;
1223
        foreach ($siswa as $row){
1224             $objPHPExcel->getActiveSheet()-
>setCellValue('A'.$no, ++$counter.'.');
            $objPHPExcel->getActiveSheet()-
1225 >setCellValue('B'.$no, $row->nomor_pendaftaran);
            $objPHPExcel->getActiveSheet()-
1226 >setCellValue('C'.$no, $row->nama_pendaftar);
            $objPHPExcel->getActiveSheet()-
1227 >setCellValue('D'.$no, $row->asal_sekolah);
            $no++;

```

```

    }

    $objWriter =
128    PHPExcel_IOFactory::createWriter($objPHPExcel,
129     'Excel2007');

    $objWriter-
130     >save("laporan_ppdb_masukdaftarcadangan.xlsx");
131     $this->load->helper('download');
132
    force_download('laporan_ppdb_masukdaftarcadangan.xlsx',
    NULL);
133     }

134     function export_ppdb()
    {
135         $this->db->select("*");
136         $this->db->from("tbl_ppdb");
137         $siswa = $this->db->get()->result();

138         $this->load->library('PHPExcel');
        $objPHPExcel = $objPHPExcel =
    PHPExcel_IOFactory::load("laporanppdb/template_ppdb_totalc
139     alonsiswa.xlsx");
        $objPHPExcel->getActiveSheet()-
140     >setCellValue('A2', 'TAHUN AJARAN
    '.get_tahun_akademik('tahun_akademik'));
141
        $no = 7;
142         $counter = 0;
143
144         foreach ($siswa as $row){
145             $objPHPExcel->getActiveSheet()-
    >setCellValue('A'.$no, ++$counter.'.');
146             $objPHPExcel->getActiveSheet()-
147     >setCellValue('B'.$no, $row->nomor_pendaftaran);
148             $objPHPExcel->getActiveSheet()-
149     >setCellValue('C'.$no, $row->nama_pendaftar);
150             $objPHPExcel->getActiveSheet()-
    >setCellValue('D'.$no, $row->asal_sekolah);

            $no++;
        }
    }

```

151	<code>\$objWriter</code>	=
152	<code>PHPExcel_IOFactory::createWriter(\$objPHPExcel, 'Excel2007');</code>	
153	<code>\$objWriter->save("laporan_ppdb_totalcalonsiswa.xlsx");</code>	
154	<code>\$this->load->helper('download');</code>	
155	<code>force_download('laporan_ppdb_totalcalonsiswa.xlsx', NULL);</code>	
156	<code>}</code>	
	<code>}</code>	
	<code>?></code>	

6.3.3 Memasukkan Detail Nilai

Kode program memasukkan detail nilai merupakan bagian dari *use case* memasukkan detail nilai yang disediakan oleh sistem. Tabel 6.5 merupakan implementasi algoritme yang terdapat pada *class controller* nilai.

Tabel 6.5 Implementasi Algoritme Memasukkan Detail Nilai

No	Kode Program
1	<code><?php</code>
2	
3	<code>class Nilai extends CI_Controller</code>
4	<code>{</code>
5	
6	<code>function __construct()</code>
7	<code>{</code>
8	<code>parent::__construct();</code>
9	<code>//checkAksesModule();</code>
10	<code>\$this->load->model('model_siswa');</code>
11	<code>}</code>
12	
13	<code>function index()</code>
14	<code>{</code>
15	<code>\$sql = "SELECT tj.kd_kelas, tj.id_jadwal, tju.nama_jurusan, ttk.nama_tingkatan, tm.nama_mapel, tj.jam, tr.nama_ruangan, tj.hari, tj.semester</code>
16	<code>FROM tbl_jadwal AS tj, tbl_jurusan AS tju, tbl_ruangan AS tr, tbl_mapel AS tm, tbl_tingkatan_kelas AS ttk</code>
17	<code>WHERE tj.kd_jurusan = tju.kd_jurusan</code>
	<code>AND tj.kd_ruangan = tr.kd_ruangan</code>



```
18         AND tj.kd_mapel = tm.kd_mapel
19         AND tj.kd_tingkatan = ttk.kd_tingkatan
20         AND      tj.id_guru      =      ".$this->session-
>userdata('id_guru');
21         $data['jadwal'] =$this->db->query($sql);
22         $this->template->load('template',
'nilai/list_kelas', $data);
23     }
24
25     function uploadFile($filename){
26         $this->load->library('upload'); // Load librari
upload
27
28         $config['upload_path'] = './excel_temp/';
29         $config['allowed_types'] = 'xls|xlsx|csv';
30         $config['max_size'] = '2048';
31         $config['overwrite'] = true;
32         $config['file_name'] = $filename;
33
34         $this->upload->initialize($config); // Load
konfigurasi uploadnya
35         if($this->upload->do_upload('file')){ //
Lakukan upload dan Cek jika proses upload berhasil
36             // Jika berhasil :
37             $return = array('result' => 'success',
'file' => $this->upload->data(), 'error' => '');
38             return $return;
39         }else{
40             // Jika gagal :
41             $return = array('result' => 'failed',
'file' => '', 'error' => $this->upload->display_errors());
42             return $return;
43         }
44     }
45
46     function form(){
47         $data = array(); // Buat variabel $data sebagai
array
48
49         $query = "SELECT * FROM tbl_verifikasi_nilai";
50         $batasAkses      =      $this->db->query($query)-
>row_array();
```

```
43
44         $sid_jadwal = $_POST['id_jadwal'];
45
46
47         if      ($batasAkses['tanggal_verifikasi'] >
date("Y-m-d")) {
48             if(isset($_POST['upload'])){ // Jika user
menekan tombol Preview pada form
49                 // lakukan upload file dengan memanggil
function upload yang ada di SiswaModel.php
50                 $upload = $this-
>uploadFile('excel_temp');
51
52                 if($upload['result'] == "success"){ //
Jika proses upload sukses
53                     // Load plugin PHPExcel nya
54                     include
APPATH.'third_party/PHPExcel/PHPExcel.php';
55                     $inputFileType =
PHPExcel_IOFactory::identify('./excel_temp/'.$this->upload-
>data('file_name'));
56                     $csvreader =
PHPExcel_IOFactory::createReader($inputFileType);
57                     $loadcsv = $csvreader-
>load('./excel_temp/'.$this->upload->data('file_name')); //
Load file yang tadi diupload ke folder csv
58                     $sheet = $loadcsv-
>getActiveSheet()->getRowIterator();
59
60                     // Masukan variabel $sheet ke dalam
array data yang nantinya akan di kirim ke file form.php
61                     // Variabel $sheet tersebut berisi
data-data yang sudah diinput di dalam csv yang sudah di
upload sebelumnya
62                     // $data['sheet'] = $sheet;
63
64                     // Buat sebuah variabel array untuk
menampung array data yg akan kita insert ke database
65                     $numrow = 1;
66                     try {
67                         foreach($sheet as $row){
```

```

65 // Cek $numrow apakah
lebih dari 1
66 // Artinya karena baris
pertama adalah nama-nama kolom
67 // Jadi dilewat saja,
tidak usah diimport
// if($numrow > 1){
// START -->
68 // Skrip untuk
mengambil value nya
$cellIterator =
69 $row->getCellIterator();
$cellIterator-
70 >setIterateOnlyExistingCells(false); // Loop all cells, even
if it is not set
71
72 // Valuenya akan di simpan kedalam array,dimulai dari index
ke 0
$iget = array();
73 foreach
($cellIterator as $cell) {
74     array_push($iget, $cell->getValue()); // Menambahkan
75     value ke variabel array $iget
76 }
77     if (!empty($iget))
78         break;
79     } else {
80         // <-- END
81         // Ambil
data value yang telah di ambil dan dimasukkan ke variabel
$iget
82         $nis =
$iget[0]; // Ambil data NIS dari kolom A di csv
83         $nama_siswa
= $iget[1]; // Ambil data nama dari kolom B di csv
84         $pengetahuan = $iget[2]; // Ambil data jenis kelamin
dari kolom C di csv
85
86         $keterampilan = $iget[3]; // Ambil data alamat dari
kolom D di csv
87
88         $parameter
= [

```

```

89         => $nis,                                     'nis'
90         'id_jadwal' => $id_jadwal,
91         'pengetahuan' => $pengetahuan,
92         'keterampilan' => $keterampilan
93                                                     ];
94
95         = array(                                     $validasi
96         => $nis,                                     'nis'
97         'id_jadwal' => $id_jadwal
98                                                     );
99         $check
100        = $this->db->get_where('tbl_nilai', $validasi);
101        if ($check-
102        >num_rows() > 0) {
103                                                     //
104        Apabila datanya besar dari 0 / ada maka akan melakukan proses
105        update
106
107        $this->db->where('nis', $nis);
108
109        $this->db->where('id_jadwal', $id_jadwal);
110
111        $this->db->update('tbl_nilai', array(
112
113        'pengetahuan' => $pengetahuan,
114
115        'keterampilan' => $keterampilan
116
117        //'sikap' => $sikap
118
119                                                     ));
120
121        } else {
122                                                     //
123        Jika datanya tidak ada maka akan melakukan proses insert
124
125        $this->db->insert('tbl_nilai', $parameter);
126
127        }

```

```

112 >where('nis', $nis);
113 >where('id_jadwal', $id_jadwal);
114 >update('tbl_nilai', array(
115     'pengetahuan' => $pengetahuan,
116     'keterampilan' => $keterampilan
117     //'sikap' => $sikap
118     ));
119 // Tambah 1 setiap kali looping
120 // } else {
121 //     continue;
122 // }
123 // }
124 // } catch (\Exception $e) {
125     $this->session-
126 >set_flashdata('true', 'Terjadi kesalahan! '.$e-
127 >getMessage());
128     redirect("nilai/kelas/".$id_jadwal);
129 }
130 // Panggil fungsi insert_multiple yg
131 telah kita buat sebelumnya di model
132 //
133 >insert_batch('tbl_nilai', $data);
134
135     $this->session-
136 >set_flashdata('true', 'Impor data berhasil!');
137
138     redirect("nilai/kelas/".$id_jadwal);
139 } else{ // Jika proses upload gagal
140     $data['upload_error'] =
141 $upload['error']; // Ambil pesan error uploadnya untuk
142 dikirim ke file form dan ditampilkan
143     $this->session-
144 >set_flashdata('true', 'Impor data gagal!
145 '.$data['upload_error']);
146
147     redirect("nilai/kelas/".$id_jadwal);

```

```

135         }
136     }
137     } else {
138         $this->session->set_flashdata('true',
139 'Telah melebihi batas waktu akses!');
140         redirect("nilai/kelas/".$sid_jadwal);
141     }
142
143     // $this->load->view('siswa/form', $data);
144 }
145
146 function kelas()
147 {
148     $sid_jadwal = $this->uri->segment(3);
149     $jadwal = $this->db->get_where('tbl_jadwal', array('id_jadwal' => $sid_jadwal))->row_array();
150     $kd_kelas = $jadwal['kd_kelas'];
151     // $kelas = "SELECT tk.*,
152     tj.nama_jurusan, ttk.nama_tingkatan
153     // FROM tbl_kelas AS tk,
154     tbl_jurusan AS tj, tbl_tingkatan_kelas AS ttk
155     // WHERE tk.kd_jurusan =
156     tj.kd_jurusan AND tk.kd_tingkatan = ttk.kd_tingkatan AND
157     kd_kelas = '$kd_kelas'";
158
159     // Punya akademik nuris
160
161     // $rombel = "SELECT
162     rb.nama_rombel,rb.kelas,jr.nama_jurusan, mp.nama_mapel
163     // FROM tbl_jadwal AS j,tbl_jurusan
164     as jr, tbl_rombel as rb,tbl_mapel as mp
165     // WHERE j.kd_jurusan=jr.kd_jurusan
166     and rb.id_rombel=j.id_rombel and mp.kd_mapel=j.kd_mapel
167     // and
168     j.id_jadwal=13='$sid_rombel'";
169
170     $kelas = "SELECT
171     tk.nama_kelas, tju.nama_jurusan, tm.nama_mapel,
172     ttk.nama_tingkatan
173     FROM tbl_jadwal AS tj, tbl_jurusan AS tju,
174     tbl_kelas AS tk, tbl_mapel AS tm, tbl_tingkatan_kelas AS ttk
175     WHERE tj.kd_jurusan = tju.kd_jurusan AND
176     tj.kd_kelas = tk.kd_kelas AND tj.kd_mapel = tm.kd_mapel AND

```

```

159 tj.kd_tingkatan = ttk.kd_tingkatan AND tj.id_jadwal =
160 $id_jadwal";
161         $siswa = "SELECT ts.nis,
ts.nama_siswa
162         FROM tbl_riwayat_kelas AS trk, tbl_siswa AS ts
163         WHERE trk.nis = ts.nis AND trk.kd_kelas =
'$kd_kelas' AND trk.id_tahun_akademik =".
get_tahun_akademik('id_tahun_akademik') ." ";
164         $batasAkses = "SELECT * FROM
tbl_verifikasi_nilai";
165         $data['kelas'] = $this->db->query($kelas)-
>row_array();
166         $data['siswa'] = $this->db->query($siswa)-
>result();
167         $data['batasAkses'] = $this->db-
>query($batasAkses)->row_array();
168         $this->template->load('template',
'nilai/form_nilai', $data);
169     }
170     function update_nilai()
171     {
172         $query = "SELECT * FROM tbl_verifikasi_nilai";
173         $batasAkses = $this->db->query($query)-
>row_array();
174         $nis = $_POST['nis'];
175         $idjadwal = $_POST['id_jadwal'];
176         $pengetahuan =
$_POST['pengetahuan'];
177         $keterampilan =
$_POST['keterampilan'];
178         if ($batasAkses['tanggal_verifikasi'] >
date("Y-m-d")) {
179             foreach ($nis as $key => $v) {
180                 $parameter = array(
181                     'nis' => $v,
182                     'id_jadwal' => $idjadwal,
183                     'pengetahuan' =>
$pengetahuan[$key],

```

```

179     $keterampilan[$key]                'keterampilan'                =>
180                                         //'sikap' => $sikap
181                                         );
182     $validasi    = array(
183         'nis' => $v,
184         'id_jadwal' => $idjadwal
185     );
186     $check      = $this->db-
>get_where('tbl_nilai', $validasi);
187     if ($check->num_rows() > 0) {
188         // Apabila datanya besar dari
0 / ada maka akan melakukan proses update
189         $this->db->where('nis', $v);
190         $this->db->where('id_jadwal',
191 $idjadwal);
192         $this->db-
>update('tbl_nilai', array(
193     $pengetahuan[$key],                'pengetahuan'                =>
194     $keterampilan[$key]                'keterampilan'                =>
195                                         //'sikap' => $sikap
196                                         ));
197         $this->session-
>set_flashdata('true', 'Edit Data Berhasil!');
198     } else {
199         // Jika datanya tidak ada maka
akan melakukan proses insert
200         $this->db-
>insert('tbl_nilai', $parameter);
201         $this->session-
>set_flashdata('true', 'Tambah Data Berhasil!');
202     }
203 }
204
205     redirect(base_url().'nilai/kelas/'.$idjadwal);
206     } else {
207         $this->session->set_flashdata('true',
'Input data gagal! Telah melewati batas waktu akses.');
```

208	<code> redirect(base_url().'/nilai/kelas/'. \$idjadwal);</code>
209	<code> }</code>
210	<code> }</code>
211	<code> }</code>
	<code>?></code>

6.3.4 Melihat Detail Nilai

Kode program melihat detail nilai merupakan bagian dari *use case* melihat detail nilai yang disediakan oleh sistem. Tabel 6.6 merupakan implementasi algoritme yang terdapat pada *class controller* Laporan_nilai.

Tabel 6.6 Implementasi Algoritme Melihat Detail Nilai

No	Kode Program
1	<code><?php</code>
2	
3	<code>class Nilai extends CI_Controller</code>
4	<code>{</code>
5	
6	<code> function __construct()</code>
7	<code> {</code>
8	<code> parent::__construct();</code>
9	<code> //checkAksesModule();</code>
10	<code> \$this->load->model('model_absensi');</code>
11	<code> }</code>
12	
13	<code> function index()</code>
14	<code> {</code>
15	<code> \$sql = "SELECT tj.kd_kelas, tj.id_jadwal, tju.nama_jurusan, ttk.nama_tingkatan, tm.nama_mapel, tj.jam, tr.nama_ruangan, tj.hari, tj.semester</code>
16	<code> FROM tbl_jadwal AS tj, tbl_jurusan AS tju, tbl_ruangan AS tr, tbl_mapel AS tm, tbl_tingkatan_kelas AS ttk</code>
17	<code> WHERE tj.kd_jurusan = tju.kd_jurusan</code>
18	<code> AND tj.kd_ruangan = tr.kd_ruangan</code>
19	<code> AND tj.kd_mapel = tm.kd_mapel</code>
20	<code> AND tj.kd_tingkatan = ttk.kd_tingkatan</code>
	<code> AND tj.id_guru = ".\$this->session->userdata('id_guru');</code>



```
21         $data['jadwal'] = $this->db->query($sql);
22         $this->template->load('template',
23         'nilai/list_kelas', $data);
24     }
25     function nilai_sikap()
26     {
27         $this->db->select("*");
28         $this->db->from("tbl_walikelas");
29         $this->db->where('id_guru',          $this->session-
30         >userdata('id_guru'));
31         $query = $this->db->get();
32
33         $kelas = $query->result();
34
35         foreach ($kelas as $row) {
36             $kelas = $row->kd_kelas;
37         }
38
39         $this->db->select("*");
40         $this->db->from("tbl_siswa");
41         $this->db->where('kd_kelas', $kelas);
42         $query1 = $this->db->get();
43
44         $siswa = $query1->result();
45
46         $this->db->select("*");
47         $this->db->from("tbl_siswa");
48         $this->db->where('kd_kelas', $kelas);
49         $query1 = $this->db->get();
50
51         if(isset($_POST['submit'])){
52             $query = "SELECT * FROM
53             tbl_verifikasi_nilai";
54             $batasAkses = $this->db->query($query)-
55             >row_array();
56             if ($batasAkses['tanggal_verifikasi'] >
57             date("Y-m-d")) {
58                 $this->model_absensi->save();
59             }
60         }
61     }
62 }
```

```
53         $this->session->set_flashdata('true',  
54 'Input nilai berhasil!');  
55  
56         redirect('nilai/nilai_sikap');  
57     } else {  
58         $this->session->set_flashdata('true',  
59 'Input nilai gagal! Telah melebihi batas waktu yang  
determined!');  
  
        redirect('nilai/absensi');  
60    }  
61 } else {  
  
62  
63     $data['kelas'] = $kelas;  
64     $data['siswa'] = $siswa;  
65  
66     $batasAkses = "SELECT * FROM  
tbl_verifikasi_nilai";  
67     $data['batasAkses'] = $this->db->  
>query($batasAkses)->row_array();  
68     $this->template->  
load('template','nilai/absensi', $data);  
69     }  
70 }  
71 function update_sikap()  
{  
72     $query = "SELECT * FROM tbl_verifikasi_nilai";  
73     $batasAkses = $this->db->query($query)->  
74     row_array();  
  
75     $nis = $_POST['nis'];  
76     $idjadwal = $_POST['id_jadwal'];  
77     $pengetahuan = $_POST['pengetahuan'];  
78     $keterampilan = $_POST['keterampilan'];  
  
79     if ($batasAkses['tanggal_verifikasi'] > date("Y-m-  
d")) {  
80         foreach ($nis as $key => $v) {  
            $parameter = array(  
                'nis' => $v,
```

```

81         'id_jadwal' => $idjadwal,
82         'pengetahuan' =>
83 $pengetahuan[$key],
84         'keterampilan' =>
85 $keterampilan[$key]
86         //'sikap' => $sikap
87     );
88
89     $validasi = array(
90         'nis' => $v,
91         'id_jadwal' => $idjadwal
92     );
93
94     $check = $this->db->
95 >get_where('tbl_nilai', $validasi);
96     if ($check->num_rows() > 0) {
97         // Apabila datanya besar dari 0 /
98         ada maka akan melakukan proses update
99         $this->db->where('nis', $v);
100        $this->db->where('id_jadwal',
101 $idjadwal);
102        $this->db->update('tbl_nilai',
103 array(
104         'pengetahuan' =>
105 $pengetahuan[$key],
106         'keterampilan' =>
107 $keterampilan[$key]
108         //'sikap' => $sikap
109     ));
110        $this->session->
111 >set_flashdata('true', 'Edit Data Berhasil!');
112    } else {
113        // Jika datanya tidak ada maka
114        akan melakukan proses insert
115        $this->db->insert('tbl_nilai',
116 $parameter);
117        $this->session->
118 >set_flashdata('true', 'Tambah Data Berhasil!');
119    }
120
121    }
122
123    redirect(base_url().'/nilai/kelas/'.$idjadwal);
124    } else {

```

```
108         $this->session->set_flashdata('true', 'Input
109 data gagal! Telah melewati batas waktu input nilai.');
```

```
110         redirect(base_url().'/nilai/kelas/'.$idjadwal);
111     }
112
113 }
114 function uploadFile($filename){
115     $this->load->library('upload'); // Load librari
116     upload
117     $config['upload_path'] = './excel_temp/';
118     $config['allowed_types'] = 'xls|xlsx|csv';
119     $config['max_size'] = '2048';
120     $config['overwrite'] = true;
121     $config['file_name'] = $filename;
122     $this->upload->initialize($config); // Load
123     konfigurasi uploadnya
124     if($this->upload->do_upload('file')){ // Lakukan
125     upload dan Cek jika proses upload berhasil
126         // Jika berhasil :
127         $return = array('result' => 'success', 'file'
128 => $this->upload->data(), 'error' => '');
129         return $return;
130     }else{
131         // Jika gagal :
132         $return = array('result' => 'failed', 'file'
133 => '', 'error' => $this->upload->display_errors());
134         return $return;
135     }
136 }
137
138 function form(){
139     $data = array(); // Buat variabel $data sebagai
140     array
141     $query = "SELECT * FROM tbl_verifikasi_nilai";
142     $batasAkses = $this->db->query($query)-
143     >row_array();
144
145     $id_jadwal = $_POST['id_jadwal'];
```

```
139
140         if ($batasAkses['tanggal_verifikasi'] > date("Y-m-
141 d")) {
142             if(isset($_POST['upload'])){ // Jika user
menekan tombol Preview pada form
143                 // lakukan upload file dengan memanggil
function upload yang ada di SiswaModel.php
144                 $upload = $this->uploadFile('excel_temp');
145
146                 if($upload['result'] == "success"){ // Jika
proses upload sukses
147                     // Load plugin PHPExcel nya
148                     include
APPATH.'third_party/PHPExcel/PHPExcel.php';
149
150                     $inputFileType =
PHPExcel_IOFactory::identify('./excel_temp/'.$this->upload-
>data('file_name'));
151
152                     $csvreader =
PHPExcel_IOFactory::createReader($inputFileType);
153
154                     $loadcsv = $csvreader-
>load('./excel_temp/'.$this->upload->data('file_name')); //
Load file yang tadi diupload ke folder csv
155                     $sheet = $loadcsv->getActiveSheet()-
>getRowIterator();
156
157                     // Masukan variabel $sheet ke dalam
array data yang nantinya akan di kirim ke file form.php
158                     // Variabel $sheet tersebut berisi
data-data yang sudah diinput di dalam csv yang sudah di upload
sebelumnya
159
160                     // $data['sheet'] = $sheet;
161
162                     // Buat sebuah variabel array untuk
menampung array data yg akan kita insert ke database
163                     $numrow = 1;
164                     try {
165                         foreach($sheet as $row){
166                             // Cek $numrow apakah lebih
dari 1
167                             // Artinya karena baris
pertama adalah nama-nama kolom
```

```
156 // Jadi dilewat saja, tidak
    usah diimport
157 // if($numrow > 1){
    // START -->
158 // Skrip untuk
    mengambil value nya
    $cellIterator =
159 $row->getCellIterator();
    $cellIterator-
160 >setIterateOnlyExistingCells(false); // Loop all cells, even if
161 it is not set
162
    $get = array(); //
163 Valuenya akan di simpan kedalam array,dimulai dari index ke 0
    foreach
    ($cellIterator as $cell) {
164     array_push($get, $cell->getValue()); // Menambahkan value
    ke variabel array $get
165     }
166     if (!empty($get)) {
167         break;
    } else {
168         // <-- END
    // Ambil data
169 value yang telah di ambil dan dimasukkan ke variabel $get
    $nis =
170 $get[0]; // Ambil data NIS dari kolom A di csv
    $nama_siswa =
171 $get[1]; // Ambil data nama dari kolom B di csv
    $pengetahuan =
    $get[2]; // Ambil data jenis kelamin dari kolom C di csv
    $keterampilan
    = $get[3]; // Ambil data alamat dari kolom D di csv
172     $sikap
    = $get[4];
173
174     $parameter = [
175         'nis' =>
    $nis,
176
177         'id_jadwal' => $id_jadwal,
178
179         'pengetahuan' => $pengetahuan,
```

```

180     'keterampilan' => $keterampilan,
181                                     => $sikap
182                                     ];
183                                     $validasi =
array(
184                                     'nis' =>
$nis,
185                                     'id_jadwal' => $id_jadwal
186                                     );
187                                     $check
= $this->db->get_where('tbl_nilai', $validasi);
188                                     if ($check-
>num_rows() > 0) {
189                                     //
190                                     Apabila datanya besar dari 0 / ada maka akan melakukan proses
update
191                                     $this-
>db->where('nis', $nis);
192                                     $this-
>db->where('id_jadwal', $id_jadwal);
193                                     $this-
>db->update('tbl_nilai', array(
194                                     'pengetahuan' => $pengetahuan,
195                                     'keterampilan' => $keterampilan,
196                                     'sikap' => $sikap
197                                     // 'sikap' => $sikap
198                                     ));
199                                     } else {
200                                     // Jika
201                                     datanya tidak ada maka akan melakukan proses insert
202                                     $this-
>db->insert('tbl_nilai', $parameter);
203                                     }
204                                     $this->db-
>where('nis', $nis);

```

```

199     $this->db-
>where('id_jadwal', $id_jadwal);
200     $this->db-
201     >update('tbl_nilai', array(
        'pengetahuan' => $pengetahuan,
        'keterampilan' => $keterampilan,
202                                     'sikap'
        => $sikap
203     //'sikap' => $sikap
204                                     ));
        $numrow++; //
205     Tambah 1 setiap kali looping
206                                     // } else {
207                                     //     continue;
208                                     // }
209                                     }
210     } catch (\Exception $e) {
211         $this->session-
>set_flashdata('true', 'Terjadi kesalahan! '.$e->getMessage());
212         redirect("nilai/kelas/".$id_jadwal);
213     }
214     // Panggil fungsi insert_multiple yg
telah kita buat sebelumnya di model
215     //
        $this->db-
>insert_batch('tbl_nilai', $data);
216     $this->session->set_flashdata('true',
'Impor data berhasil!');
        redirect("nilai/kelas/".$id_jadwal);
217     } else{ // Jika proses upload gagal
        $data['upload_error'] =
218     $upload['error']; // Ambil pesan error uploadnya untuk dikirim
ke file form dan ditampilkan
219     $this->session->set_flashdata('true',
'Impor data gagal! '.$data['upload_error']);
220     redirect("nilai/kelas/".$id_jadwal);
221     }
222     }
223     }

```

```

224         } else {
225             $this->session->set_flashdata('true', 'Telah
226 melebihi batas waktu input nilai!');
227             redirect("nilai/kelas/".$id_jadwal);
228         }
229
230         // $this->load->view('siswa/form', $data);
231     }
232
233     function kelas()
234     {
235         $id_jadwal      = $this->uri->segment(3);
236         $jadwal          = $this->db->get_where('tbl_jadwal', array('id_jadwal' => $id_jadwal))->row_array();
237         $kd_kelas        = $jadwal['kd_kelas'];
238         // $kelas         = "SELECT tk.*,
239 tj.nama_jurusan, ttk.nama_tingkatan
240 //                       FROM tbl_kelas AS tk,
241 tbl_jurusan AS tj, tbl_tingkatan_kelas AS ttk
242 //                       WHERE tk.kd_jurusan =
243 tj.kd_jurusan AND tk.kd_tingkatan = ttk.kd_tingkatan AND
244 kd_kelas = '$kd_kelas'";
245
246         // Punya akademik nuris
247
248         // $rombel        = "SELECT
249 rb.nama_rombel,rb.kelas,jr.nama_jurusan, mp.nama_mapel
250 //                       FROM tbl_jadwal AS j,tbl_jurusan
251 as jr, tbl_rombel as rb,tbl_mapel as mp
252 //                       WHERE j.kd_jurusan=jr.kd_jurusan
253 and rb.id_rombel=j.id_rombel and mp.kd_mapel=j.kd_mapel
254 //                       and j.id_jadwal=13='$id_rombel'";
255
256         $kelas          = "SELECT tk.nama_kelas,
257 tju.nama_jurusan, tm.nama_mapel, ttk.nama_tingkatan
258 //                       FROM tbl_jadwal AS tj, tbl_jurusan AS tju,
259 tbl_kelas AS tk, tbl_mapel AS tm, tbl_tingkatan_kelas AS ttk
260 //                       WHERE tj.kd_jurusan = tju.kd_jurusan AND
261 tj.kd_kelas = tk.kd_kelas AND tj.kd_mapel = tm.kd_mapel AND
262 tj.kd_tingkatan = ttk.kd_tingkatan AND tj.id_jadwal =
263 $id_jadwal";
264
265         $siswa          = "SELECT ts.nis,
266 ts.nama_siswa

```

```

FROM tbl_riwayat_kelas AS trk, tbl_siswa AS ts
249 WHERE trk.nis = ts.nis AND trk.kd_kelas =
'$kd_kelas' AND trk.id_tahun_akademik =".
get_tahun_akademik('id_tahun_akademik') ." ";
250
251 $batasAkses = "SELECT * FROM tbl_verifikasi_nilai";

252 $data['kelas'] = $this->db->query($kelas)-
>row_array();
253 $data['siswa'] = $this->db->query($siswa)-
>result();
$datenilai = $this->db->query($batasAkses)-
>row_array();
254 $this->template->load('template',
'nilai/form_nilai', $data);
255 }

function update_nilai()
{
257 $query = "SELECT * FROM tbl_verifikasi_nilai";
$batasAkses = $this->db->query($query)-
>row_array();
258
$nis = $_POST['nis'];
259 $idjadwal = $_POST['id_jadwal'];
$pengetahuan = $_POST['pengetahuan'];
$keterangan = $_POST['keterangan'];
260 $sikap = $_POST['sikap'];

261 if ($batasAkses['tanggal_verifikasi'] > date("Y-m-
d")) {
foreach ($nis as $key => $v) {
262 $parameter = array(
'nis' => $v,
'id_jadwal' => $idjadwal,
'pengetahuan' =>
$pengetahuan[$key],
'keterangan' =>
$keterangan[$key],
'sikap' =>
263 $sikap[$key]
//'sikap' => $sikap

```

```

);
264         $validasi = array(
                'nis' => $v,
265         'id_jadwal' => $idjadwal
266     );
        $check = $this->db-
267 >get_where('tbl_nilai', $validasi);
        if ($check->num_rows() > 0) {
267             // Apabila datanya besar dari 0 /
            ada maka akan melakukan proses update
                $this->db->where('nis', $v);
268             $this->db->where('id_jadwal',
269 $idjadwal);
                $this->db->update('tbl_nilai',
270 array(
271     'pengetahuan' =>
                $pengetahuan[$key],
272     'keterampilan' =>
273     $keterampilan[$key],
274     'sikap'
275     => $sikap[$key]
                // 'sikap' => $sikap
                ));
276             $this->session-
277 >set_flashdata('true', 'Edit Data Berhasil!');
            } else {
278             // Jika datanya tidak ada maka
279 akan melakukan proses insert
                $this->db->insert('tbl_nilai',
                $parameter);
280             $this->session-
281 >set_flashdata('true', 'Tambah Data Berhasil!');
282             }
283         }
284
            redirect(base_url().'/nilai/kelas/'.$idjadwal);
285         } else {
286             $this->session->set_flashdata('true', 'Input
            data gagal! Telah melewati batas waktu input nilai.');
```

	}
289	}
290	?>

6.4 Implementasi Antarmuka Pengguna

Subbab implementasi antarmuka pengguna menjelaskan beberapa hasil implementasi dalam potongan gambar antarmuka pengguna sistem informasi. Subbab berisi beberapa hasil implementasi antarmuka pengguna diantaranya antarmuka pengguna autentikasi, antarmuka pengguna *login*, antarmuka pengguna mengisi formulir pendaftaran peserta didik baru, antarmuka pengguna melakukan verifikasi pendaftaran, dan antarmuka pengguna memasukkan detail nilai. Implementasi antarmuka pengguna dilakukan berdasarkan hasil perancangan antarmuka pengguna yang telah dilakukan dalam penelitian ini.

6.4.1 Antarmuka Pengguna Autentikasi

Implementasi antarmuka pengguna autentikasi ditampilkan pengguna melalui peramban pada laptop atau komputer pada saat menggunakan sistem. Implementasi antarmuka pengguna autentikasi merupakan penerapan dari rancangan antarmuka pengguna autentikasi. Antarmuka pengguna pada Gambar 6.3 menunjukkan beberapa komponen antarmuka di antaranya judul sistem, logo sekolah, dan 2 tombol yaitu ppdb dan sias. Tombol ppdb digunakan untuk menjalankan fungsi ke dalam menu Pendaftaran Peserta Didik Baru atau PPDB, kemudian tombol sias digunakan untuk menjalankan fungsi ke dalam menu Sistem Informasi Akademik Siswa atau SIAS.



Gambar 6.3 Antarmuka Pengguna Autentikasi

6.4.2 Antarmuka Pengguna Mengisi Formulir Pendaftaran Peserta Didik Baru

Antarmuka pengguna mengisi formulir pendaftaran peserta didik baru merupakan tampilan sistem yang akan dilihat oleh pendaftar atau calon siswa,

pada saat pendaftar menggunakan sistem untuk mengisi formulir pendaftaran peserta didik baru. Antarmuka pengguna mengisi formulir pendaftaran didik baru merupakan penerapan rancangan antarmuka pengguna mengisi formulir pendaftaran peserta didik baru. Pada Gambar 6.4 menunjukkan beberapa komponen antarmuka pengguna mengisi formulir pendaftaran peserta didik baru.

Hasil implementasi antarmuka pengguna mengisi formulir pendaftaran peserta didik baru menunjukkan komponen *sidebar* yang berisi nama sistem informasi di bagian atas, dan beberapa menu sistem. Pada bagian utama halaman terdapat dua bagian, bagian pertama menunjukkan nomor pendaftaran calon siswa dan juga tanggal saat mendaftar. Setiap *dropdown* mengelompokkan kolom untuk memasukkan data diri, yang terdiri dari data pribadi, data orang tua / wali, dan data berkas. Selain itu terdapat juga tombol untuk menjalankan fungsi formulir pendaftaran peserta didik baru.



Gambar 6.4 Antarmuka Pengguna Mengisi Formulir Pendaftaran Peserta Didik Baru

6.4.3 Antarmuka Pengguna Melakukan Verifikasi Pendaftaran

Antarmuka pengguna melakukan verifikasi pendaftaran merupakan tampilan sistem yang akan dilihat oleh admin, pada saat pendaftar telah menyelesaikan pendaftaran peserta didik baru, data pendaftar tersebut masuk ke dalam verifikasi pendaftaran. Antarmuka pengguna melakukan verifikasi pendaftaran merupakan penerapan rancangan antarmuka pengguna melakukan verifikasi pendaftaran. Pada Gambar 6.5 menunjukkan beberapa komponen antarmuka pengguna melakukan verifikasi pendaftaran.

Hasil implementasi antarmuka pengguna melakukan verifikasi pendaftaran menunjukkan komponen *sidebar* yang berisi nama sistem informasi dan logo sistem di bagian atas, dan beberapa menu sistem. Sedangkan di sisi kanan terdapat 2 kelompok komponen. Komponen pertama menunjukkan status terdaftar, status diterima, status masuk daftar cadangan, dan total calon siswa.

Selanjutnya komponen kedua terdapat tombol cetak data PPDB diterima, cetak data PPDB masuk daftar cadangan, cetak data PPDB terdaftar, dan cetak data PPDB total calon siswa. Kemudian terdapat tabel, tabel tersebut menunjukkan beberapa kolom, yaitu nomor pendaftaran, nama calon siswa, status, dan aksi. Pada kolom aksi terdapat beberapa tombol, yaitu tombol diterima, tombol masuk daftar cadangan, dan tombol *delete*.



Gambar 6.5 Antarmuka Pengguna Melakukan Verifikasi Pendaftaran

6.4.4 Antarmuka Pengguna Memasukkan Detail Nilai

Antarmuka pengguna memasukkan detail nilai merupakan tampilan sistem yang akan dilihat oleh guru, pada saat guru telah menyelesaikan perhitungan nilai akhir, data nilai tersebut masuk ke dalam detail nilai yang bisa dilihat oleh masing – masing siswa. Antarmuka pengguna memasukkan detail nilai merupakan penerapan rancangan antarmuka pengguna memasukkan detail nilai. Pada Gambar 6.6 menunjukkan beberapa komponen antarmuka pengguna memasukkan detail nilai.

Hasil implementasi antarmuka pengguna memasukkan detail nilai menunjukkan dua kelompok komponen antarmuka di sisi kiri dan kanan. Di sisi kiri terdapat *sidebar* yang berisi nama sistem informasi dan logo sistem di bagian atas, dan beberapa menu sistem. Sedangkan di sisi kanan terdapat 3 kelompok komponen. Komponen pertama menunjukkan tahun akademik, semester, jurusan & tingkatan, mata pelajaran, dan batas waktu input nilai. Kemudian komponen kedua terdapat tombol untuk import data nilai, sesuai format yang sudah ditentukan. Selanjutnya komponen ketiga terdapat tabel siswa sesuai mata pelajaran yang diampu, yaitu berisi NIS, nama siswa, dan komponen penilaian (pengetahuan & keterampilan).

SIAS

Sulawati, S.Pd
Online

Dashboard Version 2.0

Tahun Akademik : 2019/2020
Semester : ghsap
Jurusan & Tingkat : Jurusan IPA Kelas 10 (0) (Kelas X IPA-1)
Mata Pelajaran : Bahasa Indonesia
Batas Waktu Input Nilai : 01 April 2019

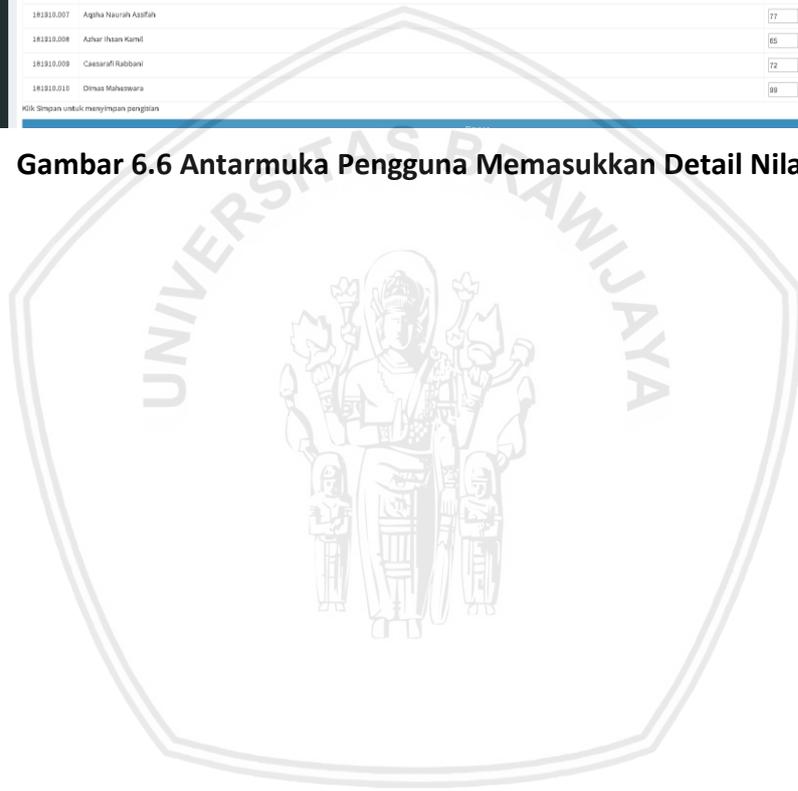
Import Data Nilai
 Create File: No file chosen
 Upload
 Klik Choose File untuk mengupload / mengunggah data penilaian dalam bentuk excel (.xlsx / .xls / .csv)

Daftar Siswa

NIS	NAMA SISWA	Pengertian	Keterampilan	Sikap
181910.001	Adhitya Fabrian D.S	97	90	85
181910.002	Ahmad Refi Zaini	69	92	89
181910.003	Akbar Adhi Pradana L.	77	83	9
181910.004	Alyssa Anindya Putri	76	89	9
181910.005	Amanda Yulia Tritantri	67	88	9
181910.006	Aprilia Kusumawingrum	77	83	9
181910.007	Aqila Naurah Asfah	77	88	9
181910.008	Azhar Ikhsan Kurni	85	88	9
181910.009	Cesarafi Rabbani	72	90	9
181910.010	Dimas Maheswara	99	88	9

Klik Simpan untuk menyimpan pengisian

Gambar 6.6 Antarmuka Pengguna Memasukkan Detail Nilai



BAB 7 PENGUJIAN

Pada bab pengujian menjelaskan hasil tahap pengujian yang dilakukan pada sistem informasi manajemen akademik, berdasarkan pada hasil perancangan dan analisis kebutuhan pengguna. Beberapa pengujian yang dilakukan terdiri dari *validation testing* dan *user acceptance testing*.

7.1 Validation Testing

Validation Testing merupakan merupakan teknik pengujian dalam lingkup metode *Black Box* yang mana kesesuaian di uji. Pengujian ini dilakukan untuk mengetahui apakah sistem yang dibangun berjalan sesuai dengan kebutuhan yang telah didefinisikan atau tidak. Subbab ini berisi hasil pengujian validasi terhadap beberapa fungsi yang digunakan oleh pengguna untuk mengisi formulir pendaftaran peserta didik baru, dan memasukkan detail nilai.



Validation testing dilakukan berdasarkan skenario yang didapatkan dari beberapa alur *use case* mengisi formulir pendaftaran peserta didik baru, dan memasukkan detail nilai. Kemudian kasus uji didapatkan berdasarkan skenario uji yang ditentukan.

7.1.1 Validation Testing Mengisi Formulir Pendaftaran Peserta Didik Baru

Validation testing mengisi formulir pendaftaran peserta didik baru menjelaskan pengujian beberapa fungsi yang dapat digunakan pengguna untuk mengisi formulir pendaftaran peserta didik baru. Beberapa skenario pada alur *use case* mengisi formulir pendaftaran peserta didik baru dijelaskan pada Tabel 7.1 yang digunakan untuk mengidentifikasi kasus uji.

Tabel 7.1 Skenario *Validation Testing* Mengisi Formulir Pendaftaran Peserta Didik Baru

Nomor Skenario	Alur Awal	Alur Alternatif
Skenario 1	<i>Basic Flow</i>	-
Skenario 2	<i>Basic Flow</i>	A1. Menangani tidak lengkapnya data

7.1.1.1 Validation Testing Fungsi Mengisi Formulir Pendaftaran Peserta Didik Baru

Tabel 7.2 berisi rencana pengujian dan kasus uji validasi untuk memastikan bahwa sistem dapat digunakan menyimpan data pendaftaran peserta didik baru



dengan kode persyaratan F-SIAS-1. Kasus uji didapatkan berdasarkan Skenario 1 pada Tabel 7.1 *Validation Testing* fungsi mengisi formulir pendaftaran peserta didik baru menunjukkan hasil valid.

Tabel 7.2 Skenario *Validation Testing* Mengisi Formulir Pendaftaran Peserta Didik Baru

Kode Pengujian	VT-SIAS-1
Nomor Skenario	Skenario 1
Kode SRS	F-SIAS-1
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat digunakan untuk memasukkan data pendaftaran peserta didik baru dan dapat menyimpan semua data pendaftaran.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji teridentifikasi sebagai Calon Siswa. 2. Penguji memilih fungsi menampilkan formulir pendaftaran peserta didik baru. 3. Penguji mengisi seluruh isian formulir Pendaftaran Peserta Didik baru dan menyertakan berkas yang perlu diunggah 4. Penguji memilih fungsi menyimpan data pendaftaran peserta didik baru.
Hasil yang diharapkan	Sistem berhasil menyimpan semua data pendaftaran dan menampilkan halaman pendaftaran sukses.
Hasil Pengujian	Sistem berhasil menyimpan semua data pendaftaran dan menampilkan halaman pendaftaran sukses.
Status Pengujian	Valid

7.1.1.2 *Validation Testing* Fungsi Menangani Tidak Lengkapnya Data

Tabel 7.3 berisi rencana pengujian dan kasus uji validasi untuk memastikan bahwa sistem dapat digunakan menampilkan pesan peringatan jika data wajib belum terisi, dengan kode persyaratan F-SIAS-1. Kasus uji didapatkan berdasarkan Skenario 2 pada Tabel 7.1 *Validation Testing* fungsi mengisi formulir pendaftaran peserta didik baru menunjukkan hasil valid.

Tabel 7.3 Skenario *Validation Testing* Mengisi Formulir Pendaftaran Peserta Didik Baru

Kode Pengujian	VT-SIAS-2
Nomor Skenario	Skenario 2
Kode SRS	F-SIAS-1
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memberikan pesan peringatan jika data wajib terisi.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji teridentifikasi sebagai Calon Siswa

	<ol style="list-style-type: none"> 2. Penguji memilih fungsi menampilkan formulir pendaftaran peserta didik baru 3. Penguji mengisi semua data pada formulir pendaftaran peserta didik baru kecuali data nama lengkap 4. Penguji memilih fungsi menyimpan data pendaftaran peserta didik baru
Hasil yang diharapkan	Sistem menampilkan pesan kolom nama lengkap harus diisi dan sistem tidak menyimpan data pendaftaran.
Hasil Pengujian	Sistem menampilkan pesan kolom nama lengkap harus diisi dan sistem tidak menyimpan data pendaftaran.
Status Pengujian	Valid

7.1.2 Validation Testing Memasukkan Detail Nilai

Validation testing memasukkan detail nilai menjelaskan pengujian beberapa fungsi yang dapat digunakan pengguna untuk memasukkan detail nilai. Beberapa skenario pada alur *use case* memasukkan detail nilai pada Tabel 7.4 yang digunakan untuk mengidentifikasi kasus uji.

Tabel 7.4 Skenario *Validation Testing* Memasukkan Detail Nilai

Nomor Skenario	Alur Awal	Alur Alternatif
Skenario 1	<i>Basic Flow</i>	-
Skenario 2	<i>Basic Flow</i>	A1. Memasukkan detail nilai yang sudah melewati batas waktu input nilai

7.1.2.1 Validation Testing Fungsi Memasukkan Detail Nilai

Tabel 7.5 berisi rencana pengujian dan kasus uji validasi untuk memastikan bahwa sistem dapat digunakan memasukkan detail nilai dengan kode persyaratan F-SIAS-2. Kasus uji didapatkan berdasarkan Skenario 1 pada Tabel 7.4 *Validation Testing* fungsi memasukkan detail nilai menunjukkan hasil valid.

Tabel 7.5 Skenario *Validation Testing* Memasukkan Detail Nilai

Kode Pengujian	VT-SIAS-3
Nomor Skenario	Skenario 1
Kode SRS	F-SIAS-3
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat digunakan untuk memasukkan data nilai siswa dan dapat menyimpan semua data nilai siswa.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji teridentifikasi sebagai guru.



	<ol style="list-style-type: none"> 2. Penguji memilih fungsi menampilkan halaman daftar kelas yang diajar. 3. Penguji memilih kelas yang diajar. 4. Penguji mengisi seluruh komponen nilai. 5. Penguji memilih fungsi menyimpan data siswa.
Hasil yang diharapkan	Sistem berhasil menyimpan data nilai siswa dan menampilkan halaman pendaftaran sukses.
Hasil Pengujian	Sistem berhasil menyimpan data nilai siswa dan menampilkan halaman pendaftaran sukses.
Status Pengujian	Valid

7.1.2.2 *Validation Testing* Fungsi Memasukkan Detail Nilai Yang Sudah Melewati Batas Waktu Input Nilai

Tabel 7.6 berisi rencana pengujian dan kasus uji validasi untuk memastikan bahwa sistem dapat digunakan menampilkan pesan peringatan telah melebihi batas waktu input nilai, dengan kode persyaratan F-SIAS-3. Kasus uji didapatkan berdasarkan Skenario 2 pada Tabel 7.4 *Validation Testing* fungsi memasukkan detail nilai menunjukkan hasil valid.

Tabel 7.6 Skenario *Validation Testing* Memasukkan Detail Nilai Yang Sudah Melewati Batas Waktu Input Nilai

Kode Pengujian	VT-SIAS-4
Nomor Skenario	Skenario 2
Kode SRS	F-SIAS-3
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memberikan pesan peringatan telah melebihi batas waktu input nilai.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji teridentifikasi sebagai guru. 2. Penguji memilih fungsi menampilkan halaman kelas yang diajar. 3. Penguji memilih kelas yang diajar. 4. Penguji memasukkan mengisi seluruh komponen nilai, tetapi sudah melewati batas waktu input nilai. 5. Penguji memilih fungsi menyimpan data siswa, dan menampilkan pesan peringatan telah melebihi batas waktu input nilai.
Hasil yang diharapkan	Sistem menampilkan pesan peringatan telah melebihi batas waktu input nilai.
Hasil Pengujian	Sistem menampilkan pesan peringatan telah melebihi batas waktu input nilai.
Status Pengujian	Valid

7.1.3 Analisis Hasil Pengujian Sistem

Ditinjau dari pengujian yang dilakukan pada tahapan sebelumnya didapatkan hasil pengujian dengan menggunakan metode *black box – validation testing*. Terdapat 4 uji kasus valid secara keseluruhan. Persentase validitas pengujian dapat dihitung dengan menggunakan cara sebagai berikut:

$$\text{Persentase valid} = (\text{Jumlah uji kasus valid} / \text{jumlah uji kasus diujikan}) \times 100\%$$

$$\text{Persentase valid} = (4 / 4) \times 100\%$$

$$\text{Persentase valid} = 100\%$$

Berdasarkan hasil perhitungan yang didapat, jumlah persentase valid pengujian yang dilakukan adalah sebesar 100%. Nilai tersebut didapatkan karena seluruh uji kasus sesuai dengan hasil yang diharapkan sesuai didefinisikan sebelumnya.

7.2 Compability Testing

-  Tidak ada kendala pada peramban
-  Critical Issue: Terdapat konten atau fungsi yang hilang pada beberapa peramban
-  Major Issues: Terdapat masalah utama pada konten atau performa pada beberapa peramban
-  Minor Issues: Terdapat masalah minor pada konten atau performa pada beberapa peramban

Gambar 7.1 Kategori Compability Pada Aplikasi SortSite

Subbab ini menjelaskan hasil *Compability testing* sistem informasi yang menunjukkan kesesuaian sistem dengan beberapa aplikasi peramban menggunakan aplikasi SortSite. Gambar 7.1 menjelaskan mengenai kategori skor untuk mengukur tingkat *Compability* pada aplikasi SortSite.

Browser	IE	Edge	Firefox	Safari	Opera	Chrome	iOS			Android	
Version	11	18	66	12	60	74	≤ 10	11	12	≤ 3	4*
Critical Issues											
Major Issues											
Minor Issues											

Gambar 7.2 Hasil Compability Testing

Compability Testing dilakukan pada peramban yang diakses oleh pengguna yang diidentifikasi sebagai admin. Gambar 7.2 merupakan hasil pengujian kompatibilitas peramban yang menunjukkan bahwa aplikasi *browser* yang direkomendasikan dalam penggunaan sistem informasi manajemen akademik sekolah tidak terdapat permasalahan ketika menjalankannya.

7.3 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) adalah pengujian yang menangani *requirement, business process*, dan kebutuhan pengguna. Pengujian ini dilakukan untuk mengetahui apakah suatu sistem yang dibangun telah memenuhi kriteria untuk diterima oleh pengguna atau tidak (Hambling, 2013).

7.3.1 Perancangan Pengujian UAT

Pengujian UAT dibagi menjadi 3 jenis berdasarkan klasifikasi aktor utama yang akan menggunakan sistem, diantaranya aktor admin, guru, dan calon siswa. Proses pengujian UAT diawali dengan mendefinisikan terlebih dahulu kriteria penerimaan pengguna yang akan digunakan untuk melakukan pengujian serta pernyataan yang akan dinilai oleh pengguna sistem dengan menggunakan skala *likert*. Tabel 7.7, tabel 7.8, dan tabel 7.9 mendeskripsikan kriteria yang akan digunakan pada pengujian ini serta pertanyaan yang akan dinilai oleh masing – masing aktor.

Tabel 7.7 Perancangan pengujian UAT untuk aktor admin

No	Kriteria	Deskripsi Kriteria	Pertanyaan
1	<i>Functional correctness and completeness</i>	Berfokus pada terpenuhinya kebutuhan yang didefinisikan sebelumnya	Apakah Sistem Informasi Manajemen Akademik Sekolah dapat diakses dengan cepat?
2	<i>Usability</i>	Berfokus pada tingkat kemudahan pengguna dalam menggunakan sistem	Apakah Sistem Informasi Manajemen Akademik Sekolah mudah digunakan atau dioperasikan?
			Mudah untuk berinteraksi dengan Sistem Informasi Manajemen Akademik Sekolah.
3	<i>Performance</i>	Berfokus pada kesesuaian kinerja sistem dan sistem sudah terdefinisi dan diinginkan oleh pengguna	Apakah dengan adanya Sistem Informasi Manajemen Akademik Sekolah dapat mempermudah admin dalam melakukan verifikasi pendaftaran?

Tabel 7.7 mendefinisikan perancangan pengujian UAT yang akan dilakukan kepada aktor admin. Terdapat 3 kriteria yang akan digunakan pada pengujian ini diantaranya *functional correctness and completeness*, *usability*, dan *performance*. Kriteria tersebut dipilih karena dinilai dapat menguji kebutuhan pengguna. Lalu terdapat 4 pertanyaan yang akan dinilai oleh aktor dengan menggunakan skala likert.

Tabel 7.8 Perancangan pengujian UAT untuk aktor guru

No	Kriteria	Deskripsi Kriteria	Pertanyaan
1	<i>Functional correctness and completeness</i>	Berfokus pada terpenuhinya kebutuhan yang didefinisikan sebelumnya	Apakah Sistem Informasi Manajemen Akademik Sekolah dapat diakses dengan cepat?
2	<i>Usability</i>	Berfokus pada tingkat kemudahan pengguna dalam menggunakan sistem	Apakah Sistem Informasi Manajemen Akademik Sekolah mudah digunakan atau dioperasikan?
			Mudah untuk berinteraksi dengan Sistem Informasi Manajemen Akademik Sekolah.
3	<i>Performance</i>	Berfokus pada kesesuaian kinerja sistem dan sistem sudah terdefinisi dan diinginkan oleh pengguna	Apakah dengan adanya Sistem Informasi Manajemen Akademik Sekolah dapat mempermudah guru dalam memasukkan detail nilai?

Tabel 7.8 mendefinisikan perancangan pengujian UAT yang akan dilakukan kepada aktor guru. Terdapat 3 kriteria yang akan digunakan pada pengujian ini diantaranya *functional correctness and completeness*, *usability*, dan *performance*. Kriteria tersebut dipilih karena dinilai dapat menguji kebutuhan pengguna. Lalu terdapat 4 pertanyaan yang akan dinilai oleh aktor dengan menggunakan skala likert.

Tabel 7.9 Perancangan pengujian UAT untuk aktor calon siswa

No	Kriteria	Deskripsi Kriteria	Pertanyaan
1	<i>Functional correctness and completeness</i>	Berfokus pada terpenuhinya kebutuhan yang didefinisikan sebelumnya	Apakah formulir Pendaftaran Peserta Didik Baru dapat diakses dengan cepat?
2	<i>Usability</i>	Berfokus pada tingkat kemudahan pengguna dalam menggunakan sistem	Apakah formulir Pendaftaran Peserta Didik Baru mudah digunakan atau dioperasikan?
			Apakah informasi yang ditampilkan oleh formulir

			Pendaftaran Peserta Didik Baru jelas dan mudah dipahami?
3	<i>Accuracy</i>	Berfokus pada kesesuaian hasil <i>input</i> dengan <i>output</i> yang seharusnya.	Apakah dengan adanya formulir Pendaftaran Peserta Didik Baru dapat mempermudah dalam mengisi formulir PPDB secara online?

Tabel 7.9 mendefinisikan perancangan pengujian UAT yang akan dilakukan kepada aktor calon siswa. Terdapat 3 kriteria yang akan digunakan pada pengujian ini diantaranya *functional correctness and completeness*, *Usability*, dan *Accuracy*. Kriteria tersebut dipilih karena dinilai dapat menguji kebutuhan pengguna. Lalu terdapat 4 pertanyaan yang akan dinilai oleh aktor dengan menggunakan skala likert.

Setelah menentukan kriteria dan pernyataan yang akan dinilai pada pengujian UAT, langkah berikutnya membuat *Acceptance Test Plan* (ATP) berisi tugas – tugas yang akan dilakukan oleh masing-masing aktor, perancangan ATP tersebut dideskripsikan pada tabel 7.10.

Tabel 7.10 Perancangan *Acceptance Test Plan* (ATP)

No	Deskripsi Tugas
Aktor Admin	
1	Menambah data siswa SMA Sejahtera 1 Depok
2	Menambah data guru SMA Sejahtera 1 Depok
3	Memverifikasi calon siswa Pendaftaran Peserta Didik Baru
4	Mencetak data Pendaftaran Peserta Didik Baru
5	Memasukkan batas waktu input detail nilai
Aktor Guru	
1	Memasukkan detail nilai secara manual
2	Memasukkan detail nilai menggunakan fungsi import data nilai
Aktor Calon Siswa	
1	Mengisi Formulir Pendaftaran Peserta Didik Baru
2	Mencetak Formulir Pendaftaran Peserta Didik Baru
3	Melihat Pengumuman Pendaftaran Peserta Didik Baru

Tabel 7.10 merupakan deskripsi dari perancangan ATP yang tugasnya akan dijalankan oleh masing-masing aktor. Terdapat 3 tugas yang harus dijalankan oleh aktor admin, 2 tugas oleh aktor guru, dan 3 tugas oleh aktor calon siswa. Tugas tersebut dirancang agar aktor dapat memberikan penilaian terhadap pertanyaan UAT yang telah dibuat sebelumnya.

7.3.2 Hasil Pengujian UAT

Pada sub bab ini merupakan hasil pengujian *User Acceptance Testing* (UAT) berdasarkan perancangan pengujian sebelumnya. Hasil pengujian UAT terdiri dari pengujian UAT melakukan verifikasi pendaftaran kepada 1 guru BPTIK yaitu sebagai aktor admin, pengujian UAT memasukkan detail nilai kepada 1 guru yaitu sebagai aktor guru, dan pengujian UAT mengisi formulir pendaftaran peserta didik baru kepada 1 calon siswa yaitu sebagai aktor calon siswa. Tabel 7.11 mendeskripsikan hasil pengujian UAT untuk aktor admin.

Tabel 7.11 Hasil pengujian UAT terhadap aktor admin

No	Pertanyaan	Penilaian				
		Sangat Setuju	Setuju	Netral	Tidak Setuju	Sangat Tidak Setuju
1	Apakah Sistem Informasi Manajemen Akademik Sekolah dapat diakses dengan cepat?	1	0	0	0	0
2	Apakah Sistem Informasi Manajemen Akademik Sekolah mudah digunakan atau dioperasikan?	0	1	0	0	0
3	Mudah untuk berinteraksi dengan Sistem Informasi Manajemen Akademik Sekolah	0	1	0	0	0
4	Apakah dengan adanya Sistem Manajemen Akademik Sekolah dapat mempermudah admin dalam melakukan verifikasi pendaftaran?	1	0	0	0	0

Pada tabel 7.11 dideksripsikan form penilaian hasil pengujian UAT terhadap aktor admin. Terdapat 4 poin pertanyaan yang diujikan kepada responden. Tabel 7.12 mendeskripsikan rekapitulasi hasil UAT terhadap aktor admin.



Tabel 7.12 Rekapitulasi hasil UAT terhadap aktor admin

No	Jawaban	Jumlah Jawaban
1	Sangat Setuju	2
2	Setuju	2
3	Netral	0
4	Tidak Setuju	0
5	Sangat Tidak Setuju	0

Dengan 4 pertanyaan yang diujikan pada pengujian UAT terhadap aktor admin yang dideskripsikan pada tabel 7.12, jawaban “Sangat Setuju” dipilih oleh responden dengan jumlah 2 kali, lalu dengan jawaban “Setuju” sebanyak 2 kali, dan responden tidak memilih sama sekali jawaban “Netral”, “Tidak Setuju”, serta “Sangat Tidak Setuju” pada pertanyaan yang diajukan. Selanjutnya dilakukan pengujian UAT terhadap aktor guru. Hasil pengujian UAT tersebut dideskripsikan pada tabel 7.13.

Tabel 7.13 Hasil pengujian UAT terhadap aktor guru

No	Pertanyaan	Penilaian				
		Sangat Setuju	Setuju	Netral	Tidak Setuju	Sangat Tidak Setuju
1	Apakah Sistem Informasi Manajemen Akademik Sekolah dapat diakses dengan cepat?	0	1	0	0	0
2	Apakah Sistem Informasi Manajemen Akademik Sekolah mudah digunakan atau dioperasikan?	1	0	0	0	0
3	Mudah untuk berinteraksi dengan Sistem Informasi Manajemen Akademik Sekolah	1	0	0	0	0
4	Apakah dengan adanya Sistem Informasi Manajemen Akademik Sekolah dapat mempermudah guru dalam memasukkan detail nilai?	0	1	0	0	0



Pada tabel 7.13 dideskripsikan form penilaian hasil pengujian UAT terhadap aktor guru. Terdapat 4 poin pertanyaan yang diujikan kepada responden. Tabel 7.14 mendeskripsikan rekapitulasi hasil UAT terhadap aktor guru.

Tabel 7.14 Rekapitulasi hasil UAT terhadap aktor admin

No	Jawaban	Jumlah Jawaban
1	Sangat Setuju	2
2	Setuju	2
3	Netral	0
4	Tidak Setuju	0
5	Sangat Tidak Setuju	0

Dengan 4 pertanyaan yang diujikan pada pengujian UAT terhadap aktor admin yang dideskripsikan pada tabel 7.14, jawaban “Sangat Setuju” dipilih oleh responden dengan jumlah 2 kali, lalu dengan jawaban “Setuju” sebanyak 2 kali, dan responden tidak memilih sama sekali jawaban “Netral”, “Tidak Setuju”, serta “Sangat Tidak Setuju” pada pertanyaan yang diajukan. Dan yang terakhir, pengujian UAT dilakukan terhadap aktor calon siswa. Hasil pengujian UAT tersebut dideskripsikan pada tabel 7.15.

Tabel 7.15 Hasil pengujian UAT terhadap aktor calon siswa

No	Pertanyaan	Penilaian				
		Sangat Setuju	Setuju	Netral	Tidak Setuju	Sangat Tidak Setuju
1	Apakah formulir Pendaftaran Peserta Didik Baru dapat diakses dengan cepat?	1	0	0	0	0
2	Apakah formulir Pendaftaran Peserta Didik Baru mudah digunakan atau dioperasikan?	0	0	1	0	0
3	Apakah informasi yang ditampilkan oleh formulir Pendaftaran Peserta Didik Baru jelas dan mudah dipahami?	1	0	0	0	0
4	Apakah dengan adanya formulir Pendaftaran Peserta Didik Baru dapat mempermudah dalam	0	1	0	0	0

	mengisi formulir PPDB secara online?					
--	--------------------------------------	--	--	--	--	--

Pada tabel 7.15 dideskripsikan form penilaian hasil pengujian UAT terhadap aktor calon siswa. Terdapat 4 poin pertanyaan yang diujikan kepada responden. Tabel 7.16 mendeskripsikan rekapitulasi hasil pengujian UAT pada aktor calon siswa.

Tabel 7.16 Rekapitulasi hasil UAT terhadap aktor calon siswa

No	Jawaban	Jumlah Jawaban
1	Sangat Setuju	2
2	Setuju	1
3	Netral	1
4	Tidak Setuju	0
5	Sangat Tidak Setuju	0

Dengan 4 pertanyaan yang diujikan pada pengujian UAT terhadap aktor calon siswa yang dideskripsikan pada tabel 7.16, jawaban "Sangat Setuju" merupakan paling banyak dipilih oleh responden dengan jumlah 3 kali dipilih, lalu diikuti dengan jawaban "Setuju" sebanyak 1 kali, lalu jawaban "Netral" sebanyak 1 kali, dan responden tidak memilih sama sekali jawaban "Tidak Setuju" serta "Sangat Tidak Setuju" pada pertanyaan yang diajukan.

7.3.3 Analisis Hasil Pengujian UAT

Setelah dilakukan pengujian pada perancangan UAT, tahapan selanjutnya adalah melakukan analisis serta perhitungan terhadap hasil pengujian UAT yang telah didapatkan dengan menggunakan skala *likert*. Hasil perhitungan UAT terhadap aktor admin adalah:

1. Aktor admin

1.1. Total Nilai

$$\text{Jawaban Sangat Setuju (A)} = 5 \times 2 = 10$$

$$\text{Jawaban Setuju (B)} = 4 \times 2 = 8$$

$$\text{Jawaban Netral (C)} = 3 \times 0 = 0$$

$$\text{Jawaban Tidak Setuju (D)} = 2 \times 0 = 0$$

$$\text{Jawaban Sangat Tidak Setuju (E)} = 1 \times 0 = 0$$

$$\text{Total Nilai} = 10 + 8 + 0 + 0 + 0 = 18$$

1.2. Nilai Y

$$\text{Nilai Y} = 5 \times 1 \times 4 = 20$$

1.3. Indeks Persentase

$$\text{Indeks persentase} = (18/20) \times 100\% = 90\%$$

Berdasarkan perhitungan yang telah dilakukan, aktor admin memiliki hasil penilaian sebesar 90% terhadap sistem informasi manajemen akademik sekolah. Berdasarkan skala likert yang telah dipaparkan sebelumnya, indeks persentase tersebut termasuk ke dalam kategori "Sangat Setuju". Sehingga dapat disimpulkan aktor admin sangat setuju dengan adanya sistem informasi ini. Kemudian terdapat perhitungan UAT terhadap aktor guru, dan berikut hasil perhitungannya:

2. Aktor guru

2.1. Total Nilai

$$\text{Jawaban Sangat Setuju (A)} = 5 \times 2 = 10$$

$$\text{Jawaban Setuju (B)} = 4 \times 2 = 8$$

$$\text{Jawaban Netral (C)} = 3 \times 0 = 0$$

$$\text{Jawaban Tidak Setuju (D)} = 2 \times 0 = 0$$

$$\text{Jawaban Sangat Tidak Setuju (E)} = 1 \times 0 = 0$$

$$\text{Total Nilai} = 10 + 8 + 0 + 0 + 0 = 18$$

2.2. Nilai Y

$$\text{Nilai Y} = 5 \times 1 \times 4 = 20$$

2.3. Indeks Persentase

$$\text{Indeks persentase} = (18/20) \times 100\% = 90\%$$

Berdasarkan perhitungan yang telah dilakukan, aktor guru memiliki hasil penilaian sebesar 90% terhadap sistem informasi manajemen akademik sekolah. Berdasarkan skala likert yang telah dipaparkan sebelumnya, indeks persentase tersebut termasuk ke dalam kategori "Sangat Setuju". Sehingga dapat disimpulkan aktor guru sangat setuju dengan adanya sistem informasi ini. Kemudian terdapat perhitungan UAT terhadap aktor calon siswa, dan berikut hasil perhitungannya:

3. Aktor calon siswa

3.1. Total Nilai

$$\text{Jawaban Sangat Setuju (A)} = 5 \times 2 = 10$$

$$\text{Jawaban Setuju (B)} = 4 \times 1 = 4$$

$$\text{Jawaban Netral (C)} = 3 \times 1 = 3$$

$$\text{Jawaban Tidak Setuju (D)} = 2 \times 0 = 0$$

$$\text{Jawaban Sangat Tidak Setuju (E)} = 1 \times 0 = 0$$

$$\text{Total Nilai} = 10 + 4 + 3 + 0 + 0 = 17$$

3.2. Nilai Y

$$\text{Nilai Y} = 5 \times 1 \times 4 = 20$$

3.3. Indeks Persentase

$$\text{Indeks persentase} = (17/20) \times 100\% = 85\%$$

Berdasarkan perhitungan yang telah dilakukan, aktor calon siswa memiliki hasil penilaian sebesar 85% terhadap sistem informasi manajemen akademik sekolah. Berdasarkan skala likert yang telah dipaparkan sebelumnya, indeks persentase tersebut termasuk ke dalam kategori “Sangat Setuju”. Sehingga dapat disimpulkan aktor calon siswa sangat setuju dengan adanya sistem informasi ini.



BAB 8 PENUTUP

8.1 Kesimpulan

Kesimpulan yang didapat berdasarkan penelitian yang dilakukan bahwa Sistem Informasi Manajemen Akademik Sekolah pada SMA Sejahtera 1 Depok dapat dikembangkan dengan melakukan beberapa kegiatan pengembangan sistem sebagai berikut:

1. Hasil analisis kebutuhan berupa dokumentasi yang berisi 2 hasil pemodelan proses bisnis saat ini, analisis proses bisnis saat ini, 2 hasil pemodelan proses bisnis usulan, identifikasi *stakeholder*, identifikasi fitur, kebutuhan fungsional dan nonfungsional, *use case diagram*, dan *use case scenario*. Hasil analisis kebutuhan memuat 5 fitur, 5 kebutuhan fungsional dan 1 kebutuhan nonfungsional yang merepresentasikan kemampuan sistem secara umum. Hasil analisis kebutuhan juga memuat informasi mengenai tujuan penggunaan sistem dan urutan aktivitas yang dilakukan untuk mencapai tujuan tersebut. Informasi tersebut di dokumentasikan pada 6 *use case*. Setiap *use case* akan dijelaskan urutan aktivitas utama dan aktivitas alternative yang didokumentasikan pada spesifikasi *use case*.
2. Hasil perancangan sistem yang dilakukan berdasarkan hasil analisis kebutuhan terdiri dari perancangan dan pemodelan arsitektur sistem, *Activity Diagram*, *Sequence Diagram*, *Class Diagram*, *Physical Data Model*, dan perancangan antarmuka. Hasil perancangan juga memuat pemodelan objek yang seperti *class diagram* yaitu kelas perancangan untuk *controller*, kelas perancangan untuk *model*, dan kelas perancangan untuk *view*. *Class Diagram* terdiri dari 7 *class controller*, 3 *class model*, 8 *class view*. Hasil perancangan sistem juga menghasilkan beberapa pemodelan interaksi objek yang divisualisasikan ke dalam *sequence diagram*, *Physical Data Model* yang ebrisi tabel-table sebagai media penyimpanan data sistem informasi, dan beberapa rancangan antarmuka pengguna.
3. Hasil implementasi berdasarkan hasil perancangan sistem berupa Sistem Informasi Manajemen Akademik berbasis situs web dengan kerangka kerja *CodeIgniter* dengan menggunakan bahasa pemrograman PHP, HTML, CSS, dan Javascript dengan basis data MYSQL. Sistem Informasi Manajemen Akademik memberikan fitur-fitur sesuai dengan hasil analisis kebutuhan. Hasil implementasi juga menghasilkan artefak sistem yang didokumentasikan pada struktur artefak sistem, beberapa implementasi algoritme, dan beberapa implementasi antarmuka pengguna.
4. Hasil pengujian validasi pada beberapa fungsi yang berkaitan mengisi formulir Pendaftaran Peserta Didik Baru dan Memasukkan Detail Nilai dinyatakan valid. Selanjutnya hasil pengujian *User Acceptance Test* digunakan untuk mengetahui apakah sistem yang telah dikembangkan telah memenuhi kriteria untuk diterima pengguna atau tidak yang dilakukan kepada 3 jenis aktor yang telah didefinisikan dengan masing-masing jenis aktor memiliki 1 orang

responden. Hasilnya menunjukkan bahwa tingkat penerimaan pengguna pada aktor admin dan aktor guru mendapatkan persentase penerimaan sebesar 90%, lalu aktor calon siswa sebesar 85%. Ketiga persentase tersebut termasuk kedalam kategori “Sangat Setuju”. Sehingga dapat disimpulkan bahwa ketiga aktor tersebut sangat setuju dengan adanya sistem informasi manajemen akademik sekolah ini.

8.2 Saran

Berdasarkan penelitian yang telah dilakukan, terdapat beberapa saran untuk proses pengembangan selanjutnya yaitu, perlu dilakukan penerapan lapangan Sistem Informasi Manajemen Akademik Sekolah pada studi kasus Sekolah Menengah Atas Sejahtera 1 Depok. Kemudian perlu dilakukan kegiatan evaluasi terhadap penerapan dari Sistem Informasi Manajemen Akademik Sekolah sehingga memungkinkan hasil pengembangan selanjutnya yang lebih baik.



DAFTAR REFERENSI

- 1keydata,2017.*Physical Data Model*. [Online] Tersedia di: <<https://www.1keydata.com/datawarehousing/physical-data-model.html>> [Diakses 20 Januari 2019]
- Alsammarry, Y., Sirisuthi, C., & Duangcharthom, S. 2016. Development of Educational Management System in Small Primary School. *International Education Studies*, 9(12), 244.
- Becker, J., Kugeler, M. & Rosemann, M.,2013. *Process Management: A Guide for the Design of Business Processes*. Berlin: Springer-Verlag.
- Bittner, Kurt, Ian Spence.,2003. *Use Case Modelling*. Boston: Pearson Education, Inc.
- Bizagi, 2014. *BPMN by Example-Bizagi Suite*.
- Booch, G. James, R. Ivar, J, 2005. *The Unified Modeling Language User Guide*. Second Edition. United State: Addison Wesley Professional.
- Cunningham, W., 2001. *Principles Behind the Agile Manifesto*. Tersedia di <<http://agilemanifesto.org/principles.html>> [Diakses 31 Agustus 2018]
- Depdiknas. 2004. *Kerangka Dasar Kurikulum 2004*, Jakarta.
- Hambling, B., & Goethem, P. V. 2013. *User Acceptance Testing*. UK: BCS Learning & Development.
- Mishra, A. & Dubey, D., 2013. *A Comparative Study of Different Software Development Life Cycle Models in Different Scenarios*. International Journal of Advance Research in Computer Science and Management Studies, p. 6.
- Naik, K. dan Tripathy, P., 2008. *Software Testing and Quality Assurance*. Hoboken: John Wiley & Sons, Inc..
- Object Management Group (OMG), 2011. *OMG Object Management Group*. [Online] tersedia di: <<http://www.omg.org/spec/BPMN/2.0/PDF/>> [Diakses 22 Januari 2019]
- O'Brien & Marakas. 2010. *Management Information Systems*. Eighth Edition. New York: McGraw-Hill/Irwin.
- O'Brien & Marakas. 2013. *Management Information Systems*. Sixteenth Edition. New York: McGraw-Hill/Irwin.
- Prasetyo, S., 2015. *Analisis dan Perancangan Sistem Informasi Sekolah Berbasis Web Di SMK Muhammadiyah 2 Muntilan*. Universitas Negeri Yogyakarta. Daerah Istimewa Yogyakarta.
- Pressman, R. S., 2010. *Software Engineering - A practitioner's approach*. 7th Edition. Boston: McGraw-Hill.

- Pressman, R. S., 2015. *Software Engineering - A practitioner's approach*. 8th Edition. Boston: McGraw-Hill.
- Przybylek, A., 2011. *Bridging the gap between business process models and use-case models*. Gdańsk, Gdańsk University of Technology.
- Rosa AS dan M.Shalahuddin. 2015. *Rekayasa Perangkat Lunak Terstruktur Dan Berorientasi Objek*. Bandung : INFORMATIKA.
- Rumbaugh, J., Jacobson, I. & Booch, G., 2005. *The Unified Modeling Language reference manual*. 2nd Edition. Boston: Addison-Wesley.
- Setiyawan, A. (2013). Pembuatan Sistem Informasi Akademik Berbasis Web. *Indonesian Journal On Networking And Security - ISSN : 2302-5700*.
- Sommerville, Ian. 2011. *Software Engineering (Rekayasa Perangkat Lunak)*. Jakarta: Erlangga.
- Supriyanto, Aji. (2005). *Pengantar Teknologi Informasi*. Jakarta: Penerbit Salemba Infotek.
- Suryandani, F. ,Basori & Maryono,D., 2017. *Pengembangan Sistem Informasi Akademik Berbasis Web Sebagai Sistem Pengolahan Nilai Siswa di SMK Negeri 1 Kudus*. Universitas Sebelas Maret. Jawa Tengah.
- Syaiful Sagala. 2010. *Konsep dan Makna Pembelajaran*. Bandung: Alfabeta.
- Turban, Efraim, Rainer, Kelly, Potter, Richard, 2010. *Introduction to Information Technology*. 9th Edition. New York: John Wiley & sons.
- Weske, M., 2007. *Business Process Management Concept, Languages, Architectures*. New York: Springer-Verlag Berlin Heidelberg.