

PENGEMBANGAN SISTEM INFORMASI LEMBAGA BIMBINGAN BELAJAR AL-HASYIMI

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Muhammad Jauhar Fuadi

NIM: 145150400111005



**PROGRAM STUDI SISTEM INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI LEMBAGA BIMBINGAN BELAJAR AL-HASYIMI

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer

Disusun Oleh :
Muhammad Jauhar Fuadi
NIM: 145150400111005

Skripsi ini telah diuji dan dinyatakan lulus pada
30 Juli 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II



Retno Indah Rokhmawati, S.Pd., M.Pd.
NIK: 201609 9009172 001

M. Chandra Saputra, S.Kom., M.Eng.
NIK: 201609 860106 1 001

Mengetahui
Ketua Jurusan Sistem Informasi



Dr. Eng. Hectian Tolle, S.T., M.T.
NIP: 19740823 200012 1 001



PERNYATAAN ORISINALITAS

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 pustaka.

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, 14 Juli 2018



Muhammad Jauhar Fuadi

NIM: 145150400111005



DAFTAR RIWAYAT HIDUP





IDENTITAS

Nama: Muhammad Jauhar Fuadi
 TTL: Tulungagung, 29 Agustus 1996
 Alamat: Ds. Punjul Kec. Karangrejo Kab. Tulungagung
 Agama: Islam
 Jenis Kelamin: Laki-Laki
 Pekerjaan: Pelajar



KONTAK

No. Handphone: 085646352142
 E-mail: jauharfuad54@gmail.com
 Facebook: Muhammad Jauhar Fuadi @jauhar54
 Line: muhammad_jauhar_fuadi
 Instagram:



HARDSKILL

- Photoshop
- Coreldraw
- HTML/CSS
- Java Script
- PHP
- SQL



SOFTSKILL

- Kreativitas
- Komunikasi
- Kepemimpinan
- Kepedulian
- Kerja Tim

RIWAYAT PENDIDIKAN

SEKOLAH	TAHUN
SDN Punjul 1	2001 - 2007
MTsN Karangrejo	2007 - 2010
MAN 1 Tulungagung	2010 - 2013
Universitas Brawijaya	2014 - Sekarang

PENGALAMAN ORGANISASI

ORGANISASI	JABATAN	TAHUN
OSPAM (Organisasi Pesantren Mahasiswa Al-Hikam Malang)	Dept. Bahasa	2015 - 2016
Unit Teknologi Informasi Al-Hikam	Ketua	2015 - Sekarang
POSKESTREN	Div. Gizi	2015 - 2016
KBH Al-Hikam	Div. Humas	2015 - 2016
KBH Al-Hikam	Div. Pendidikan	2016 - Sekarang

PENGALAMAN KEPANTIAAN

ACARA	JABATAN	TAHUN
Pekan Olahraga Santri	Div. PDD	2014
Dienstalls FILKOM	Div. Acara	2015
FILKOM Go to School	Co. Daerah	2015
Pekan Olahraga Santri	Div. Acara	2015
PASION FILKOM	Div. PDD	2016
HM Cup	Div. PDD	2016
Penerimaan Santri Baru	Div. Sekretaris	2016
HM Cup	Div. PDD	2016
Masa Orientasi Santri	Ketua	2016
Penerimaan Santri Baru	Pengelolaan Data	2017

PRESTASI

PRESTASI	ACARA	TAHUN
Juara harapan 2	I-Fest 2.0 FILKOM	2015
Finalis	E-Fest FILKOM	2015

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kehadirat Allah SWT, karena berkat rahmat dan hidayah-Nya, penulis dapat menyelesaikan laporan skripsi yang berjudul “Pengembangan Sistem Informasi Lembaga Bimbingan Belajar Al-Hasyimi”. Selanjutnya shalawat serta salam semoga tetap tercurahkan kepada Nabi Muhammad SAW yang telah menuntun kita dari zaman jahiliah sampai ke zaman yang terang benderang seperti sekarang ini.

Dalam penyusunan dan penulisan laporan ini penulis mendapatkan banyak bantuan dan dukungan dari berbagai pihak. Penulis ingin mengucapkan terimakasih yang sebesar - besarnya kepada :

1. Kedua orang tua saya, Istianah yang telah mendidik saya dari kecil hingga dewasa seperti saat ini. Ibu saya, Istianah yang tidak pernah berhenti mendo'akan dan memberikan semangat kepada saya untuk terus giat belajar. Ayah saya, Nahrowi yang tidak pernah berhenti bekerja keras dan selalu memberikan motivasi untuk saya agar menjadi pribadi yang lebih baik. Serta adik saya, Salwa Asyifa Salsabila yang saya sayangi.
2. Ibu Retno Indah Rokhmawati, S.Pd., M.Pd., selaku dosen Pembimbing 1 yang selalu sabar dalam membimbing, mengarahkan, memberikan motivasi dan saran kepada penulis selama proses pengerjaan laporan skripsi.
3. Bapak Mochammad Chandra Saputra, S.Kom., M.Eng., selaku dosen Pembimbing 2 yang selalu sabar membimbing, mengarahkan, memberi saran serta motivasi kepada penulis selama penyusunan laporan skripsi.
4. Ustadz Ismu Aditya selaku ketua Lembaga Bimbingan Belajar Al-Hasyimi serta jajarannya yang telah mengizinkan penulis untuk melakukan penelitian di Lembaga Bimbingan Belajar Al-Hasyimi. Serta selalu mendukung penulis dalam pengerjaan laporan skripsi.
5. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
6. Bapak Herman Tolle, Dr. Eng., S.T, M.T selaku Ketua Jurusan Sistem Informasi Universitas Brawijaya Malang.
7. Bapak Suprpto, S.T, M.T selaku Ketua Program Studi Sistem Informasi Universitas Brawijaya Malang, serta seluruh Dosen Program Studi Sistem Informasi Universitas Brawijaya Malang yang telah mendidik dan membimbing penulis selama proses perkuliahan.
8. Almarhum KH. Ahmad Hasyim Muzadi selaku bapak pengasuh dan pendiri Pesantren Mahasiswa Al-Hikam Malang, serta Ustadz Mohammad Nafi', Ustadz Hilman Wajdi, Ustadz Nur Kholis, Ustadz Abdul Hadi, Ustadz Muzammil, dan para asatidz lainnya yang selalu memberikan nasihat dan motivasi tentang bagaimana menjalani kehidupan ini dengan berlandaskan islam rahmatan lil 'alamin.
9. Sahabat santri Pesantren Mahasiswa Al-Hikam Malang, khususnya angkatan Ashamuda yang selalu memberikan semangat kepada penulis.
10. Sahabat G10S, Iqbal, Wendi, Mashuda, Caraka, Dhimas, Andry, Simson, Rajiv, Ian, Santo, Jawara, Vicky, Wahyu, Yoga, Yogi, Ferli, Aulia, Nozomi, Sasa, Aisyatul, dan Instan yang menjadi sahabat bermain baik dalam suka dan duka.

11. Jawara, Bayu, Stefanus, Adit serta teman-teman bimbingan Ibu Retno dan Bapak Chandra yang senantiasa mau saya ajak diskusi terkait pengerjaan laporan skripsi.
12. Keluarga besar Sistem Informasi Universitas Brawijaya angkatan 2014 yang menemani selama masa perkuliahan.

Penulis menyadari bahwa dalam penyusunan laporan skripsi ini masih banyak kekurangan baik *format* laporan maupun isinya. Untuk itu penulis sangat mengharapkan kritik dan saran yang membangun demi perbaikan selanjutnya. Semoga laporan skripsi ini dapat bermanfaat bagi pembaca dan penulis, amin.

Malang, 14 Juli 2018

Penulis

Jauharfuadi54@gmail.com



ABSTRAK

Muhammad Jauhar Fuadi, Pengembangan Sistem Informasi Lembaga Bimbingan Belajar Al-Hasyimi

Pembimbing: Retno Indah Rokhmawati, S.Pd., M.Pd. dan Mochammad Chandra Saputra, S.Kom., M.Eng.

Lembaga Bimbingan Belajar (LBB) Al-Hasyimi merupakan lembaga pendidikan swasta yang menyediakan bimbingan belajar membaca Al-Qur'an dan Iqro' baik secara privat maupun kelompok. Dalam mengelola hasil belajar siswa, LBB Al-Hasyimi masih menggunakan cara konvensional. Setiap tutor harus mencatat hasil belajar siswanya dalam buku yang disebut dengan buku prestasi siswa. Kemudian buku ini akan direkap oleh sekretaris setiap satu bulan sekali. Selain itu tutor juga harus melakukan presensi setelah melakukan bimbingan. Dengan memanfaatkan teknologi informasi, permasalahan tersebut dapat dipermudah dengan menggunakan Sistem Informasi Lembaga Bimbingan Al-Hasyimi berbasis *web* yang dikembangkan dengan metode *waterfall*. Metode pengembangan ini meliputi analisis kebutuhan, perancangan, implementasi, dan pengujian. Pada tahap analisis kebutuhan dilakukan pemodelan proses bisnis *to-be* yang dibuat berdasarkan hasil wawancara dengan *stakeholder* dan analisis permasalahan. Hasil analisis kebutuhan divisualisasikan dalam diagram *use case*. Hasil perancangan sistem menghasilkan *sequence diagram*, *class diagram*, *physical data model*, *pseudocode algorithm*, *user interface*, dan *testing design*. Implementasi dari sistem informasi ini menggunakan *framework CodeIgniter* dengan menerapkan pola *Model-View-Controller*. Hasil *Validation Testing* yang dilakukan pada 6 kasus uji memperoleh presentasi 100% *valid*. Kemudian hasil *User Acceptance Testing (UAT)* yang dilakukan pada beberapa fitur utama, diantaranya menambahkan hasil belajar siswa, melihat hasil belajar, melihat presensi tutor, dan melakukan komplain memperoleh presentase rata-rata lebih dari 80%. Dengan hasil pengujian *UAT* tersebut, maka dapat disimpulkan bahwa Sistem Informasi Lembaga Bimbingan Belajar Al-Hasyimi dapat diterima oleh pengguna.

Kata kunci: *sistem informasi, waterfall, codeigniter, user acceptance testing, web.*

ABSTRACT

Muhammad Jauhar Fuadi, Pengembangan Sistem Informasi Lembaga Bimbingan Belajar Al-Hasyimi

Pembimbing: Retno Indah Rokhmawati, S.Pd., M.Pd. dan Mochammad Chandra Saputra, S.Kom., M.Eng.

Lembaga Bimbingan Belajar (LBB) Al-Hasyimi is a private educational institution that providing study guidance in reading the Holy Koran and Iqro' both by private and by group. In managing student's study result, LBB Al-Hasyimi still uses a conventional method. Every mentor has to note their student's study result in the book which called by "Buku Prestasi Siswa". Then this book would be recapped by secretary once in every month. In the other hand, the mentor had to do checking presence after guiding too. By using information technology, the problem can be simplified by using Lembaga Bimbingan Belajar Al-Hasyimi Information System based on website which developed by using waterfall method. This development method covered requirement analysis, design, implementation and testing. On the requirement analysis produced business process to-be that created based on the result of stakeholder interview and problem analysis. The result of the requirement analysis visualized in use case diagram. The result of the system design produced sequence diagram, class diagram, physical data model, pseudocode algorithm, user interface, and testing design. On its implementation, this system developed by using CodeIgniter framework by applying the Model-View-Controller pattern. The result of Validation Testing on 6 test case got as many as 100% valid. Then the result of User Acceptance Testing (UAT) on some features, there are adding a study result, seeing a study result, seeing a tentor absence, and complaining got average percentage more than 80%. With the result of UAT revealed that Lembaga Bimbingan Belajar Al-Hasyimi Information System already accepted by user.

Keywords: information system, waterfall, codeigniter, user acceptance testing, web.

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	v
ABSTRAK.....	vii
ABSTRACT	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR.....	xvii
DAFTAR LAMPIRAN	xix
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 2 LANDASAN KEPUSTAKAAN	Error! Bookmark not defined.
2.1 Kajian Pustaka	Error! Bookmark not defined.
2.2 Sistem Informasi	Error! Bookmark not defined.
2.3 Website	Error! Bookmark not defined.
2.4 Proses Bisnis.....	Error! Bookmark not defined.
2.4.1 <i>Business Process Model and Notation</i>	Error! Bookmark not defined.
2.5 <i>Unifield Modeling Language</i>	Error! Bookmark not defined.
2.5.1 <i>Indentifikasi Stakeholder</i>	Error! Bookmark not defined.
2.5.2 Analisis Permasalahan.....	Error! Bookmark not defined.
2.5.3 Analisis Kebutuhan.....	Error! Bookmark not defined.
2.5.4 Identifikasi Fitur	Error! Bookmark not defined.
2.5.5 Spesifikasi Kebutuhan	Error! Bookmark not defined.
2.5.6 <i>Pemodelan Use Case</i>	Error! Bookmark not defined.



2.5.7 Use Case Diagram	Error! Bookmark not defined.
2.5.8 Activity Diagram	Error! Bookmark not defined.
2.5.9 Sequence Diagram	Error! Bookmark not defined.
2.5.10 Class Diagram	Error! Bookmark not defined.
2.6 Software Process Models	Error! Bookmark not defined.
2.6.1 Waterfall	Error! Bookmark not defined.
2.7 Black-box Testing	Error! Bookmark not defined.
2.7.1 Validation Testing	Error! Bookmark not defined.
2.7.2 Compatibility Testing	Error! Bookmark not defined.
2.7.3 User Acceptance Testing	Error! Bookmark not defined.
BAB 3 METODOLOGI PENELITIAN	Error! Bookmark not defined.
3.1 Studi Literatur	Error! Bookmark not defined.
3.2 Analisis Kebutuhan	Error! Bookmark not defined.
3.3 Perancangan Sistem	Error! Bookmark not defined.
3.4 Implementasi Sistem	Error! Bookmark not defined.
3.5 Pengujian dan Analisis	Error! Bookmark not defined.
3.6 Pengambilan Kesimpulan dan Saran	Error! Bookmark not defined.
BAB 4 ANALISIS KEBUTUHAN	Error! Bookmark not defined.
4.1 Identifikasi Pemangku Kepentingan	Error! Bookmark not defined.
4.2 Analisis Proses Bisnis	Error! Bookmark not defined.
4.2.1 Identifikasi Proses Bisnis <i>As-Is</i>	Error! Bookmark not defined.
4.2.2 Analisis Permasalahan	Error! Bookmark not defined.
4.2.3 Pemodelan Proses Bisnis <i>To-be</i>	Error! Bookmark not defined.
4.3 Analisis Perbaikan Proses Bisnis	Error! Bookmark not defined.
4.4 Analisis Kebutuhan <i>Stakeholder</i> dan Pengguna	Error! Bookmark not defined.
4.5 Identifikasi Aktor	Error! Bookmark not defined.
4.6 Identifikasi Fitur	Error! Bookmark not defined.
4.7 Spesifikasi Kebutuhan	Error! Bookmark not defined.
4.8 Pemodelan <i>Use Case</i>	Error! Bookmark not defined.
4.9 <i>Use Case Scenario</i>	Error! Bookmark not defined.



- 4.9.1 *Use Case Scenario* Menambah Pengguna **Error! Bookmark not defined.**
- 4.9.2 *Use Case Scenario* Mengedit Data Pengguna **Error! Bookmark not defined.**
- 4.9.3 *Use Case Scenario* Menghapus Pengguna **Error! Bookmark not defined.**
- 4.9.4 *Use Case Scenario* Melihat List Pengguna **Error! Bookmark not defined.**
- 4.9.5 *Use Case Scenario* Menambahkan Hasil Belajar **Error! Bookmark not defined.**
- 4.9.6 *Use Case Scenario* Melihat Hasil Belajar **Error! Bookmark not defined.**
- 4.9.7 *Use Case Scenario* Menerima Notifikasi **Error! Bookmark not defined.**
- 4.9.8 *Use Case Scenario* Melihat Presensi Tentor **Error! Bookmark not defined.**
- 4.9.9 *Use Case Scenario* Mengirim Komplain **Error! Bookmark not defined.**
- 4.9.10 *Use Case Scenario* Mengelola Komplain **Error! Bookmark not defined.**
- 4.9.11 *Use Case Scenario* Melihat Feedback **Error! Bookmark not defined.**
- 4.9.12 *Use Case Scenario* Membuat Pengumuman **Error! Bookmark not defined.**
- 4.9.13 *Use Case Scenario* Melihat Pengumuman **Error! Bookmark not defined.**
- 4.9.14 *Use Case Scenario* Membuat Biodata **Error! Bookmark not defined.**
- 4.9.15 *Use Case Scenario* Mengedit Biodata **Error! Bookmark not defined.**
- 4.9.16 *Use Case Scenario* Mengubah Password **Error! Bookmark not defined.**
- 4.9.17 *Use Case Scenario* Login.....**Error! Bookmark not defined.**
- 4.10 *Activity Diagram***Error! Bookmark not defined.**
- 4.10.1 *Activity Diagram* Menambahkan Data Hasil Belajar.....**Error! Bookmark not defined.**

- 4.10.2 *Activity Diagram* Melihat Hasil Belajar Siswa **Error! Bookmark not defined.**
- 4.10.3 *Activity Diagram* Melihat Presensi Tentor **Error! Bookmark not defined.**
- 4.10.4 *Activity Diagram* Melakukan Komplain **Error! Bookmark not defined.**

BAB 5 PERANCANGAN.....Error! Bookmark not defined.

- 5.1 Perancangan *Sequence Diagram* **Error! Bookmark not defined.**
 - 5.1.1 *Sequence Diagram* Menambahkan Hasil Belajar **Error! Bookmark not defined.**
 - 5.1.2 *Sequence Diagram* Melihat Hasil Belajar Siswa **Error! Bookmark not defined.**
 - 5.1.3 *Sequence Diagram* Melihat Presensi Tentor **Error! Bookmark not defined.**
 - 5.1.4 *Sequence Diagram* Melakukan Komplain **Error! Bookmark not defined.**
- 5.2 Pemodelan *Object*..... **Error! Bookmark not defined.**
 - 5.2.1 Diagram Kelas Analisis..... **Error! Bookmark not defined.**
 - 5.2.2 Diagram Kelas Perancangan..... **Error! Bookmark not defined.**
- 5.3 Perancangan *Database* **Error! Bookmark not defined.**
- 5.4 Perancangan Algoritma **Error! Bookmark not defined.**
 - 5.4.1 *Pseudocode* Menambahkan Hasil Belajar **Error! Bookmark not defined.**
 - 5.4.2 *Pseudocode* Melihat Hasil Belajar ... **Error! Bookmark not defined.**
 - 5.4.3 *Pseudocode* Melihat Presensi Tentor **Error! Bookmark not defined.**
 - 5.4.4 *Pseudocode* Melakukan Komplain .. **Error! Bookmark not defined.**
- 5.5 Perancangan *User Interface*..... **Error! Bookmark not defined.**
 - 5.5.1 *User Interface* Menambahkan Hasil Belajar **Error! Bookmark not defined.**
 - 5.5.2 *User Interface* Hasil Belajar Siswa ... **Error! Bookmark not defined.**
 - 5.5.3 *User Interface* Presensi Tentor..... **Error! Bookmark not defined.**
 - 5.5.4 *User Interface Form* Komplain **Error! Bookmark not defined.**
- 5.6 Perancangan Pengujian **Error! Bookmark not defined.**
 - 5.6.1 Validation Testing..... **Error! Bookmark not defined.**



5.6.2 *Compatibility Testing*Error! Bookmark not defined.

5.6.3 *User Acceptance Testing*Error! Bookmark not defined.

BAB 6 IMPLEMENTASIError! Bookmark not defined.

6.1 Implementasi AlgoritmaError! Bookmark not defined.

6.1.1 Menambahkan Hasil Belajar Siswa .Error! Bookmark not defined.

6.1.2 Melihat Hasil Belajar SiswaError! Bookmark not defined.

6.1.3 Melihat Presensi TentorError! Bookmark not defined.

6.1.4 Melakukan KomplainError! Bookmark not defined.

6.2 Implementasi *User Interface*Error! Bookmark not defined.

6.2.1 Menambahkan Hasil BelajarError! Bookmark not defined.

6.2.2 Melihat Hasil BelajarError! Bookmark not defined.

6.2.3 Melihat Presensi TentorError! Bookmark not defined.

6.2.4 *Form* KomplainError! Bookmark not defined.

BAB 7 PENGUJIANError! Bookmark not defined.

7.1 Validation TestingError! Bookmark not defined.

7.2 *Compatibility Testing*Error! Bookmark not defined.

7.3 *User Acceptance Testing*Error! Bookmark not defined.

7.3.1 Analisis Hasil *UAT Menambahkan Hasil Belajar*Error! Bookmark not defined.

7.3.2 Analisis Hasil *UAT Melihat Hasil Belajar*Error! Bookmark not defined.

7.3.3 Analisis Hasil *UAT Melihat Presensi tentor*Error! Bookmark not defined.

7.3.4 Analisis Hasil *UAT Melakukan Komplain*Error! Bookmark not defined.

BAB 8 KESIMPULAN.....Error! Bookmark not defined.

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

8.2 SaranError! Bookmark not defined.

DAFTAR PUSTAKA.....Error! Bookmark not defined.

LAMPIRAN A DOKUMENTASI WAWANCARAError! Bookmark not defined.

LAMPIRAN B DOKUMENTASI VALIDASI FITUR DAN ANALISIS WAKTUError! Bookmark not defined.

LAMPIRAN C DOKUMENTASI PENGUJIAN *UAT*Error! Bookmark not defined.



DAFTAR TABEL

Table 2.1 <i>BPMN Task Description</i>	Error! Bookmark not defined.
Table 2.2 <i>BPMN Task Description (lanjutan)</i>	Error! Bookmark not defined.
Table 2.3 <i>BPMN Flow Object Description</i>	Error! Bookmark not defined.
Table 2.4 <i>BPMN Data Object Description</i>	Error! Bookmark not defined.
Table 2.5 <i>BPMN Event Description</i>	Error! Bookmark not defined.
Table 2.6 <i>BPMN Gateway Description</i>	Error! Bookmark not defined.
Table 4.1 Identifikasi Pemangku Kepentingan.....	Error! Bookmark not defined.
Table 4.2 Analisis Permasalahan.....	Error! Bookmark not defined.
Table 4.3 Identifikasi Fitur (lanjutan).....	Error! Bookmark not defined.
Table 4.40 Spesifikasi Kebutuhan	Error! Bookmark not defined.
Table 4.5 Kebutuhan Fungsional.....	Error! Bookmark not defined.
Table 4.6 <i>Use Case Scenario</i> Menambah Pengguna ..	Error! Bookmark not defined.
Table 4.7 <i>Use Case Scenario</i> Mengedit Data Pengguna	Error! Bookmark not defined.
Table 4.8 <i>Use Case Scenario</i> Menghapus Pengguna.	Error! Bookmark not defined.
Table 4.9 <i>Use Case Scenario</i> Menghapus Pengguna (lanjutan)	Error! Bookmark not defined.
Table 4.10 <i>Use Case Scenario</i> Menghapus Pengguna	Error! Bookmark not defined.
Table 4.11 <i>Use Case Scenario</i> Menambahkan Data Hasil Belajar	Error! Bookmark not defined.
Table 4.12 <i>Use Case Scenario</i> Melihat Hasil Belajar Siswa	Error! Bookmark not defined.
Table 4.13 <i>Use Case Scenario</i> Melihat Hasil Belajar Siswa (lanjutan).....	Error! Bookmark not defined.
Table 4.14 <i>Use Case Scenario</i> Menerima notifikasi ..	Error! Bookmark not defined.
Table 4.15 <i>Use Case Scenario</i> Melihat Presensi Tentor	Error! Bookmark not defined.
Table 4.16 <i>Use Case Scenario</i> Melihat Presensi Tentor (lanjutan)	Error! Bookmark not defined.
Table 4.17 <i>Use Case Scenario</i> Mengirim Komplain...	Error! Bookmark not defined.

Table 4.18 *Use Case Scenario* Mengirim Komplain (lanjutan)**Error! Bookmark not defined.**

Table 4.19 *Use Case Scenario* Mengelola Komplain .**Error! Bookmark not defined.**

Table 4.20 *Use Case Scenario* Mengelola Komplain .**Error! Bookmark not defined.**

Table 4.21 *Use Case Scenario* Membuat Pengumuman**Error! Bookmark not defined.**

Table 4.22 *Use Case Scenario* Membuat Pengumuman (lanjutan)**Error! Bookmark not defined.**

Table 4.23 *Use Case Scenario* Melihat Pengumuman**Error! Bookmark not defined.**

Table 4.24 *Use Case Scenario* Membuat Biodata**Error! Bookmark not defined.**

Table 4.25 *Use Case Scenario* Membuat Biodata (lanjutan)**Error! Bookmark not defined.**

Table 4.26 *Use Case Scenario* Mengedit Biodata.....**Error! Bookmark not defined.**

Table 4.27 *Use Case Scenario* Mengubah Password.**Error! Bookmark not defined.**

Table 4.28 *Use Case Scenario* Login**Error! Bookmark not defined.**

Table 5.1 Penjelasan Fungsi Tabel**Error! Bookmark not defined.**

Table 5.2 Pseudocode Menambahkan Hasil Belajar .**Error! Bookmark not defined.**

Table 5.3 Pseudocode Melihat Hasil Belajar**Error! Bookmark not defined.**

Table 5.4 Pseudocode Melihat Presensi Tentor**Error! Bookmark not defined.**

Table 5.5 Pseudocode Melakukan Komplain**Error! Bookmark not defined.**

Table 5.6 Kasus Uji Menambahkan Hasil Belajar Siswa: Basic Flow**Error! Bookmark not defined.**

Table 5.7 Kasus Uji Menambahkan Hasil Belajar Siswa: Alt1**Error! Bookmark not defined.**

Table 5.8 Kasus Uji Melihat Hasil Belajar Siswa: Basic Flow**Error! Bookmark not defined.**

Table 5.9 Kasus Uji Melihat Presensi Tentor: Basic Flow**Error! Bookmark not defined.**

Table 5.10 Kasus Uji Melakukan Komplain: Basic Flow**Error! Bookmark not defined.**

Table 5.11 Kasus Uji Melakukan Komplain: Alt1.....**Error! Bookmark not defined.**

Table 5.12 Kategori Tingkat Permasalahan *Compatibility***Error! Bookmark not defined.**

Table 5.13 Komponen - Komponen yang Diujikan.....**Error! Bookmark not defined.**

Table 5.14 Kasus Uji *UAT* Menambahkan Hasil Belajar **Error! Bookmark not defined.**

Table 5.15 Kasus Uji *UAT* Melihat Hasil Belajar Siswa **Error! Bookmark not defined.**

Table 5.16 Kasus Uji *UAT* Melihat Presensi Tentor ... **Error! Bookmark not defined.**

Table 5.17 Kasus Uji *UAT* Melakukan Komplain **Error! Bookmark not defined.**

Table 5.18 Pertanyaan Kasus Uji Menambahkan Hasil Belajar oleh Tentor **Error! Bookmark not defined.**

Table 5.19 Pertanyaan Kasus Uji Melihat Hasil Belajar oleh Siswa dan Orang Tua **Error! Bookmark not defined.**

Table 5.20 Pertanyaan Kasus Uji Melihat Presensi Tentor oleh Sekertaris **Error! Bookmark not defined.**

Table 5.21 Pertanyaan Kasus Uji Melakukan komplain oleh Siswa dan Orang Tua **Error! Bookmark not defined.**

Table 5.22 Rancangan Pertanyaan Responden **Error! Bookmark not defined.**

Table 5.23 Keterangan Bobot Nilai **Error! Bookmark not defined.**

Table 5.24 Keterangan Presentase Nilai **Error! Bookmark not defined.**

Table 6.1 Implementasi Menambahkan Hasil Belajar **Error! Bookmark not defined.**

Table 6.2 Implementasi Melihat Hasil Belajar **Error! Bookmark not defined.**

Table 6.3 Implementasi Melihat Presensi Tentor **Error! Bookmark not defined.**

Table 6.4 Implementasi Melakukan Komplain **Error! Bookmark not defined.**

Table 7.1 Hasil *Validation Testing* **Error! Bookmark not defined.**

Table 7.2 Hasil *Validation Testing* (lanjutan) **Error! Bookmark not defined.**

Table 7.3 Hasil *Compatibility Testing* **Error! Bookmark not defined.**

Table 7.4 Hasil Kuisisioner Kasus Uji Menambahkan Hasil Belajar oleh Tentor **Error! Bookmark not defined.**

Table 7.5 Hasil Kuisisioner Kasus Uji Menambahkan Hasil Belajar oleh Siswa dan Orang Tua **Error! Bookmark not defined.**

Table 7.6 Hasil Kuisisioner Kasus Uji Melihat Presensi Tentor oleh Sekertaris.. **Error! Bookmark not defined.**

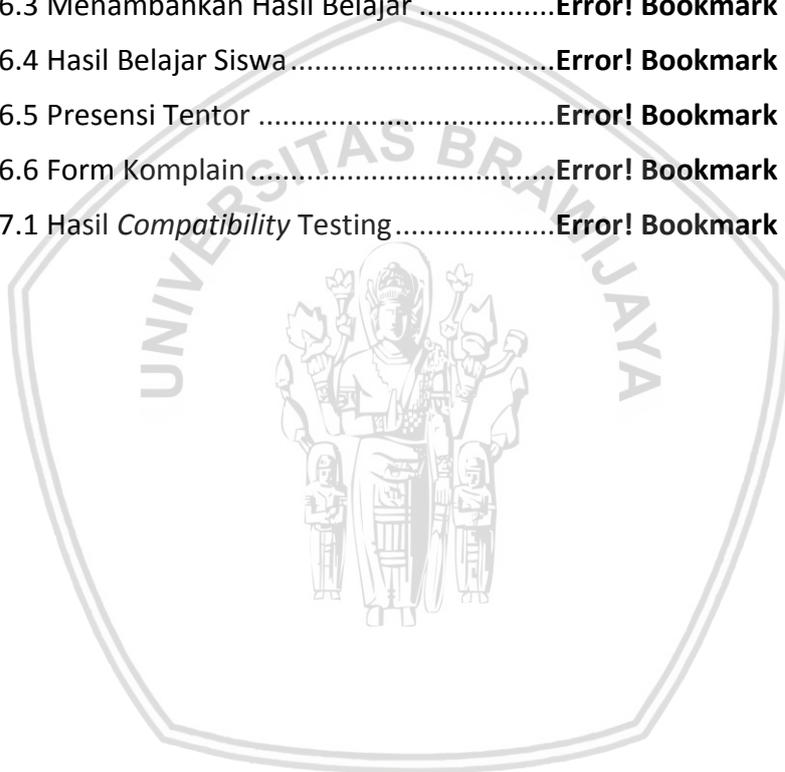
Table 7.7 Hasil Kuisisioner Kasus Uji Melakukan Komplain oleh Siswa dan Orang Tua **Error! Bookmark not defined.**



DAFTAR GAMBAR

Gambar 2.1 Sistem Informasi.....	Error! Bookmark not defined.
Gambar 2.2 Alur Diagram Model Waterfall	Error! Bookmark not defined.
Gambar 3.1 Alur Metodologi Penelitian	Error! Bookmark not defined.
Gambar 4.1 Bisnis Proses As-Is Mengelola Hasil Belajar Siswa	Error! Bookmark not defined.
Gambar 4.2 Bisnis Proses As-Is Presensi Tentor	Error! Bookmark not defined.
Gambar 4.3 Bisnis Proses As-Is Komplain Orang Tua Siswa	Error! Bookmark not defined.
Gambar 4.4 Proses Bisnis To-Be Mengelola Data Hasil Belajar	Error! Bookmark not defined.
Gambar 4.5 Proses Bisnis To-Be Presensi Tentor	Error! Bookmark not defined.
Gambar 4.6 Proses Bisnis To-Be Komplain Orang Tua Siswa	Error! Bookmark not defined.
Gambar 4.7 Contoh pembuatan kode kebutuhan.....	Error! Bookmark not defined.
Gambar 4.8 Kodifikasi Fitur	Error! Bookmark not defined.
Gambar 4.9 Diagram <i>Use Case</i>	Error! Bookmark not defined.
Gambar 4.10 Activity Diagram Menambahkan data hasil belajar	Error! Bookmark not defined.
Gambar 4.11 Activity Diagram Melihat data hasil belajar	Error! Bookmark not defined.
Gambar 4.12 Activity Diagram Memberi komplain ..	Error! Bookmark not defined.
Gambar 5.1 Squence Diagram Menambahkan Hasil Belajar	Error! Bookmark not defined.
Gambar 5.2 Sequence Diagram Melihat Hasil Belajar	Error! Bookmark not defined.
Gambar 5.3 Sequence Diagram Melihat Presensi Tentor	Error! Bookmark not defined.
Gambar 5.4 Sequence Diagram Melakukan Komplain	Error! Bookmark not defined.
Gambar 5.5 Diagram Kelas Analisis.....	Error! Bookmark not defined.
Gambar 5.6 Diagram Kelas Model.....	Error! Bookmark not defined.
Gambar 5.7 Diagram Kelas Controller	Error! Bookmark not defined.
Gambar 5.8 Physical Data Model.....	Error! Bookmark not defined.

Gambar 5.9 User Interface Hasil Belajar Siswa**Error! Bookmark not defined.**
 Gambar 5.10 User Interface Form Hasil Belajar**Error! Bookmark not defined.**
 Gambar 5.11 User Interface Hasil Belajar Siswa.....**Error! Bookmark not defined.**
 Gambar 5.12 User Interface Presensi Tentor**Error! Bookmark not defined.**
 Gambar 5.13 User Interface Form Komplain**Error! Bookmark not defined.**
 Gambar 5.14 Kodifikasi Validation Testing**Error! Bookmark not defined.**
 Gambar 6.1 Menambahkan Hasil Belajar**Error! Bookmark not defined.**
 Gambar 6.2 Menambahkan Hasil Belajar**Error! Bookmark not defined.**
 Gambar 6.3 Menambahkan Hasil Belajar**Error! Bookmark not defined.**
 Gambar 6.4 Hasil Belajar Siswa.....**Error! Bookmark not defined.**
 Gambar 6.5 Presensi Tentor**Error! Bookmark not defined.**
 Gambar 6.6 Form Komplain**Error! Bookmark not defined.**
 Gambar 7.1 Hasil *Compatibility* Testing**Error! Bookmark not defined.**



DAFTAR LAMPIRAN

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

A.1 Wawancara dengan Ketua LBB Al-Hasyimi**Error! Bookmark not defined.**

A.2 Wawancara dengan Ketua LBB Al-Hasyimi**Error! Bookmark not defined.**

A.3 Wawancara dengan Sekretaris LBB Al-Hasyimi**Error! Bookmark not defined.**

A.4 Wawancara dengan Tentor LBB Al-Hasyimi**Error! Bookmark not defined.**

A.5 Wawancara dengan Siswa LBB Al-Hasyimi**Error! Bookmark not defined.**

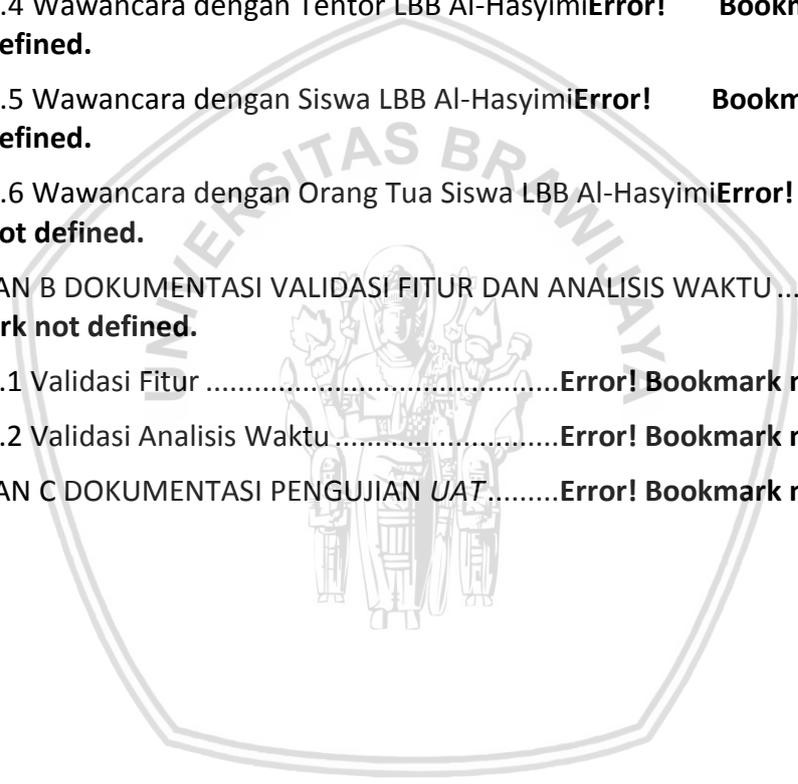
A.6 Wawancara dengan Orang Tua Siswa LBB Al-Hasyimi**Error! Bookmark not defined.**

LAMPIRAN B DOKUMENTASI VALIDASI FITUR DAN ANALISIS WAKTU**Error! Bookmark not defined.**

B.1 Validasi Fitur**Error! Bookmark not defined.**

B.2 Validasi Analisis Waktu**Error! Bookmark not defined.**

LAMPIRAN C DOKUMENTASI PENGUJIAN UAT**Error! Bookmark not defined.**





BAB 1 PENDAHULUAN

1.1 Latar belakang

Lembaga Bimbingan Belajar (LBB) Al-Hasyimi merupakan sebuah lembaga yang didirikan oleh santri Pesantren Mahasiswa Al-Hikam Malang. LBB Al-Hasyimi ini bertujuan untuk menyalurkan kemampuan para santri khususnya dalam bidang agama untuk dapat membagikan ilmunya kepada Masyarakat. Untuk saat ini LBB Al-Hasyimi menyediakan bimbingan secara privat maupun kelompok yang terdiri dari bimbingan baca Al-Quran dan Iqro'. Siswa – siswa yang terdaftar pada LBB Al-Hasyimi terdiri dari berbagai usia, mulai dari anak – anak hingga orang dewasa. Saat ini siswa LBB Al-Hasyimi sudah menyebar di seluruh kota Malang baik dari tingkat SD hingga Universitas.

Berdasarkan hasil wawancara yang dilakukan kepada Saudara Ismu Aditya selaku Ketua LBB Al-Hasyimi pada lampiran A.1 dan lampiran A.2. LBB Al-Hasyimi memiliki kendala dalam proses pengelolaan hasil belajar siswa. Dalam proses mencatat hasil belajar siswa, tutor masih menggunakan cara konvensional dengan cara mencatatnya pada buku prestasi siswa. Kemudian setiap satu bulan sekali, tutor harus menyerahkan data buku prestasi siswa tersebut kepada bagian sekretaris yang bertugas merekap data tersebut. Data tersebut untuk selanjutnya akan dimasukkan ke Microsoft Excel untuk direkap. Selain itu, setiap kali setelah melakukan bimbingan, tutor harus melakukan presensi ke Kantor LBB Al-Hasyimi.

Dari uraian masalah di atas, maka dibutuhkan sistem informasi yang dapat membantu LBB-Al-Hasyimi dalam proses mengelola data hasil belajar siswa serta proses presensi tutor. Dengan adanya bantuan sistem informasi ini, tutor dapat menyampaikan informasi terkait hasil belajar siswa dengan lebih mudah dan cepat. Selain itu, ketika tutor memasukkan dan mengirim data hasil belajar, secara otomatis tutor sudah melakukan presensi. Sistem informasi ini juga terdapat fitur komplain yang dapat dilakukan oleh siswa maupun orang tua.

Waterfall merupakan model pengembangan perangkat lunak yang menggunakan pendekatan *sequential* dalam proses pengembangannya (Pressman, 2001). Metode ini sangat sederhana dan memiliki tahap – tahap yang sistematis serta dokumentasi yang baik. Sehingga hal ini dapat membantu dalam proses pengembangan pada sistem informasi ini. Selain itu metode ini sangat cocok ketika kebutuhan dari sistem sudah diketahui di awal, sehingga dapat mempermudah dalam proses implementasi (Alshamrani & Bahattab, 2015). Maka dari itu, metode ini akan diterapkan pada penelitian ini.

Pada tahap akhir pengembangan, dilakukan pengujian terhadap sistem informasi dengan menggunakan 3 metode pengujian diantaranya *validation testing*, *compatibility testing*, dan *user acceptance testing*. *Validation testing* bertujuan untuk mengetahui apakah sistem yang dikembangkan sudah memenuhi persyaratan fungsional sehingga dapat dijalankan dengan baik dan sesuai harapan pengguna (Pressman, 2010). *Compatibility testing* bertujuan untuk mengetahui apakah sistem yang dikembangkan dengan menggunakan bahasa atau *script*

tertentu dapat berjalan dengan baik ketika dijalankan di berbagai jenis dan versi *browser* (Pressman, 2010). *User acceptance testing* bertujuan untuk memastikan apakah sistem yang dikembangkan dapat diterima oleh pengguna (Hambling & Goethem, 2013).

Penelitian sebelumnya oleh Jannah, Astuti, dan Maharani (2015) yang berjudul “Rancang Bangun Sistem Informasi Bimbingan Belajar Berbasis Web (Studi Kasus: Lembaga Bimbingan Belajar Tadica)” mengemukakan bahwa sistem pembelajaran dapat diakses baik oleh admin maupun siswa, sehingga proses bimbingan dapat berjalan dengan lebih cepat, tepat dan akurat. Adapun modul yang terdiri dari fungsi – fungsi diantaranya pendafaran *online*, penyimpanan data akademik, unduh materi atau soal, percakapan dengan siswa maupun admin melalui chat room dan dapat melakukan ujian online. Adapun fitur yang belum diterapkan pada penelitian tersebut yang akan diterapkan pada penelitian ini, yaitu penambahan user tutor atau guru dan orang tua siswa, manajemen aktifitas tutor serta penanganan komplain dari siswa maupun orang tua terkait kepengajaran tutor.

1.2 Rumusan masalah

Berdasarkan permasalahan yang dijabarkan pada latar belakang penelitian ini, maka didapatkan rumusan masalah sebagai berikut:

1. Bagaimana hasil pemodelan proses bisnis pada Sistem Informasi Bimbingan Belajar Al-Hasyimi?
2. Bagaimana hasil analisis kebutuhan pada Sistem Informasi Bimbingan Belajar Al-Hasyimi?
3. Bagaimana rancangan dan implementasi Sistem Informasi Bimbingan Belajar Al-Hasyimi?
4. Bagaimana hasil pengujian pada Sistem Informasi Lembaga Bimbingan Belajar Al-Hasyimi?

1.3 Tujuan

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Memodelkan proses bisnis pada Sistem Informasi Lembaga Bimbingan Belajar Al-Hasyimi menggunakan Business Process Model and Notation (BPMN).
2. Melakukan analisis kebutuhan pada Sistem Informasi Lembaga Bimbingan Belajar Al-Hasyimi.
3. Merancang dan mengimplementasikan sebuah Sistem Informasi Bimbingan Belajar Al-Hasyimi berbasis *website*.
4. Menguji kebutuhan fungsionalitas pada Sistem Informasi Bimbingan Belajar Al-Hasyimi.

1.4 Manfaat

Adapun manfaat yang diharapkan dalam penelitian ini adalah sebagai berikut:

1. Bagi Lembaga Bimbingan Belajar (LBB) Al-Hasyimi
Dapat memberikan kemudahan dalam mengelola hasil belajar siswa, presensi tentor, serta melayani komplain baik dari siswa maupun orang tua.
2. Bagi Siswa dan Orang Tua
Dapat memberikan kemudahan dalam menerima informasi hasil belajar dan melakukan komplain kepada LBB Al-Hasyimi.
3. Bagi Penulis
Dapat menambah pengetahuan dan pengalaman penulis dalam melakukan pengembangan Sistem Informasi LBB Al-Hasyimi ini.
4. Bagi Pembaca
Dapat menambah wawasan serta dapat menjadi referensi dalam melakukan penelitian khususnya yang memiliki topic dan studi kasus yang sama.

1.5 Batasan masalah

Agar permasalahan pada penelitian ini tidak meluas, peneliti membatasi pada hal-hal berikut:

1. Pengerjaan skripsi ini hanya berfokus pada analisis kebutuhan, perancangan, dan implementasi sistem serta pengujian sistem.
2. Sistem informasi dikembangkan menggunakan aplikasi berbasis *website* dengan bahasa pemrograman *PHP* menggunakan *framework Codeigniter* dan menggunakan *DBMS MySQL*, serta menggunakan *framework Bootstrap* untuk mempercantik tampilan.
3. Pengujian pada sistem ini menggunakan metode *validation testing*, *compatibility testing*, dan *User Acceptance Testing* dengan fokus pada kebutuhan sistem.

1.6 Sistematika pembahasan

Penulisan laporan skripsi ini terbagi menjadi delapan bab utama. Berikut ini merupakan pemaparan secara singkat inti pembahasan berserta bagian-bagian yang dibahas dari tiap masing-masing bab:

BAB I Pendahuluan

Berisi latar belakang dilakukan penelitian, hal-hal yang menjadi permasalahan dalam penelitian, tujuan dilakukannya penelitian, manfaat dilakukannya penelitian, batasan penelitian yang dilakukan, serta sistematika penulisan penelitian.

BAB II Landasan Kepustakaan

Berisi dasar-dasar teori yang akan menjadi landasan dalam pengembangan sistem informasi bimbingan belajar LBB Al-Hasyimi. Teori-teori tersebut diambil dari berbagai sumber yang terpercaya.

BAB III Metodologi

Berisi tentang tahap-tahapan penelitian serta ulasan singkat metode dan analisis penelitian yang terdiri dari Alur Penelitian.

BAB IV Analisis Kebutuhan

Berisi tentang pembahasan analisis kebutuhan baik fungsional maupun non-fungsional. Analisis kebutuhan dimulai dari identifikasi *stakeholder*, pemodelan proses bisnis, analisis permasalahan, identifikasi aktor, analisis kebutuhan, hingga pemodelan *UML*.

BAB V Perancangan

Berisi tentang perancangan sistem yang dibuat berdasarkan pada analisis kebutuhan pada bab sebelumnya. Perancangan ini terdiri dari pembuatan *sequence diagram*, *class diagram*, *physical data model*, algoritma *pseudocode*, dan *user interface*.

BAB VI Implementasi

Berisi tentang implementasi sistem yang dikembangkan berdasarkan perancangan yang dilakukan pada bab sebelumnya.

BAB VII Pengujian

Berisi tentang hasil pengujian sistem dengan menggunakan metode pengujian *validation testing*, *compatibility testing*, dan *User Acceptance Testing*.

BAB VIII Penutup

Berisi tentang kesimpulan dan saran dari penelitian yang telah dilakukan.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

TADICA (Taman Pendidikan Cendekiawan Muda) adalah lembaga bimbingan belajar yang berada di Samarinda Seberang, Kalimantan Timur. Lembaga ini memiliki beberapa program yang ditawarkan, yaitu paket baca, tulis dan menghitung, bimbingan belajar tingkat SD dan SMP, Persiapan ujian nasional SD dan SMP, serta JARITMATIKA. Pada penelitian yang dilakukan oleh Jannah, Astuti, dan Maharani (2015) yang berjudul “Rancang Bangun Sistem Informasi Bimbingan Belajar Berbasis Web (Studi Kasus: Lembaga Bimbingan Belajar TADICA)” mengemukakan bahwa dengan adanya sistem informasi bimbingan belajar berbasis web ini, dapat meningkatkan keefektifan siswa dan guru dalam proses belajar mengajar, sehingga proses bimbingan dapat berjalan dengan lebih cepat, tepat dan akurat.

Adapun modul yang terdiri dari fungsi – fungsi yang diimplementasikan pada sistem yang mereka kembangkan diantaranya pendaftaran *online*, penyimpanan data akademik, unduh materi atau soal, percakapan dengan siswa maupun admin melalui *chat room* dan dapat melakukan ujian *online*. Fungsi pendaftaran online berguna untuk siswa yang menginginkan program bimbingan belajar, sehingga ia dapat mendaftar secara *online*. Fungsi penyimpanan data berguna untuk memudahkan admin dalam mengelola data siswa yang terdaftar. Fungsi unduh materi atau soal berguna untuk memberikan fasilitas siswa yang ingin mempelajari materi ataupun soal yang telah dan akan diajarkan oleh guru. Fungsi chat room berguna untuk komunikasi antara siswa dan admin. Dan terakhir fungsi ujian online berguna bagi siswa yang ingin menguji kemampuannya.

Dari fungsi – fungsi yang dijabarkan di atas, memiliki beberapa kesamaan dan tujuan dengan fungsi – fungsi yang dibutuhkan pada sistem informasi Lembaga Bimbingan Belajar Al-Hasyimi. Beberapa fungsi diantaranya adalah penyimpanan akademik. Maka dari itu penelitian yang dilakukan oleh Jannah, Astuti, dan Maharani (2015) tersebut menjadi kajian pustaka dalam pengembangan sistem informasi Lembaga Bimbingan Belajar Al-Hasyimi ini.

Penelitian sebelumnya yang dilakukan oleh Mahalakshmi dan Sundararajan (2013) yang berjudul “*Traditional SDLC Vs Scrum Methodology – A Comparative Study*” mengungkapkan beberapa keuntungan menggunakan model pengembangan waterfall jika dibandingkan model pengembangan Scrum. Diantaranya adalah kebutuhan sudah jelas diawal, sehingga setiap fase pengembangan akan dilakukan secara berurutan dengan perkiraan waktu yang dapat ditentukan. Selain itu, model waterfall mudah untuk diimplementasikan dan cocok untuk pengembangan yang berorientasi pada proyek. *Waterfall* juga memberikan hasil dokumentasi yang lengkap dibanding dengan *Scrum*.

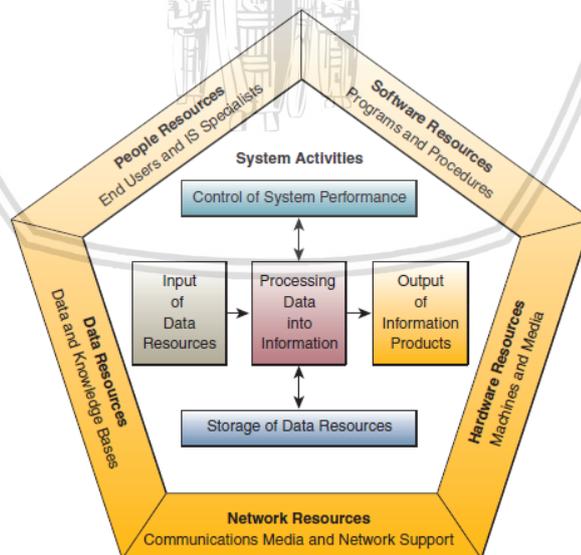
Kemudian penelitian tentang “Pemodelan *Use Case (UML)*: Evaluasi Terhadap Beberapa Kesalahan Dalam Praktik” yang dilakukan oleh Kurniawan. Dimana penelitian ini bertujuan untuk mengevaluasi beberapa kesalahan dalam

pembuatan *Use Case scenario* dan *Use Case diagram* yang sering terjadi baik dalam konteks pembelajaran di kampus maupun implementasi dalam dunia kerja. Penelitian ini berhasil mengidentifikasi 11 jenis kesalahan pembuatan *Use Case scenario* dan 7 kesalahan pembuatan *Use Case diagram*. Maka dari itu, penulis menggunakan penelitian ini untuk dijadikan acuan dalam pembuatan *Use Case scenario* dan *Use Case diagram* dalam pengembangan sistem informasi ini.

Selanjutnya penelitian yang dilakukan oleh Melnik, Read, dan Maurer yang berjudul "*Suitability of FIT User Acceptance Tests for Specifying Functional Requirements: Developer Perspective*" menyatakan bahwa metode *User Acceptance Testing (UAT)* ini diperlukan untuk memastikan kepuasan pelanggan dengan produk yang kita buat. UAT juga berfungsi sebagai regression testing, untuk memastikan bahwa fungsi kerja harus berjalan sesuai dengan yang diharapkan. Dengan ini maka spesifikasi persyaratan akan secara langsung mempengaruhi control kualitas.

2.2 Sistem Informasi

Menurut Obrien dan Marakas (2010) sistem adalah kumpulan elemen yang memiliki keterkaitan dan bekerja sama dalam menggapai tujuan sedangkan informasi adalah data yang sudah diproses sehingga bernilai manfaat bagi pengguna akhir. Maka dari itu yang dimaksud sistem informasi adalah sebuah pengorganisasian seperangkat manusia, komunikasi, data, jaringan, peraturan, prosedur, software dan hardware yang mana akan melakukan proses menyimpan, menerima, merubah dan menyebarkan informasi dalam sebuah organisasi (Obrien dan Marakas, 2010).



Gambar 2.1 Sistem Informasi

Sumber: O'Brien & Marakas (2010)



Di era modern ini, sistem informasi menjadi alat yang sangat penting bagi organisasi untuk meningkatkan efisiensi dan produktivitas yang lebih tinggi dalam proses bisnis sebuah organisasi (Laudon & Laudon, 2014). Ditengah persaingan bisnis yang semakin kompetitif, penting bagi organisasi ataupun perusahaan untuk memiliki sistem informasi sebagai salah satu strategi dalam memenangkan persaingan. Menurut O'Brien & Marakas (2010) terdapat tiga peran utama yang dimiliki sistem informasi dalam sebuah bisnis pada organisasi, diantaranya:

1. Sebagai pendukung dalam menjalankan proses dan operasi bisnis. Hampir semua toko retail sudah menggunakan sistem informasi berbasis komputer untuk membantu pegawainya dalam mencatat pembelian pelanggan, pengadaan produk, evaluasi penjualan, dan masih banyak lagi.
2. Sebagai pendukung dalam pengambilan keputusan dalam bisnis. Dengan adanya sistem informasi, membantu manajer dalam melakukan pengambilan keputusan yang tepat.

Sebagai strategi untuk memenangkan persaingan. Informasi yang sangat cepat dan mudah didapat, sangat berguna bagi sebuah organisasi untuk memanfaatkannya dalam meningkatkan kualitas proses bisnis untuk dapat bersaing dengan *competitor*.

2.3 Website

Website merupakan sekumpulan halaman web yang terhubung satu sama lain (Agung, 2000). Pertama ditemukan pada tahun 1980. Pada awalnya website digunakan hanya untuk melakukan pertukaran informasi dan memperbarui informasi untuk dibagikan kepada para peneliti di Prancis dan Swiss. Namun di era saat ini fungsi website sudah mulai meluas. Dermawan dan Permana (2013) menyebutkan terdapat lima fungsi website saat ini. Diantaranya sebagai media promosi, pemasaran, informasi, pendidikan, dan komunikasi.

Jenis – jenis website juga semakin beragam seiring berkembangnya teknologi serta kebutuhan manusia. Dermawan dan Permana (2013) kembali menyebutkan beberapa jenis website yang populer saat ini. Diantaranya website search engine, portal berita, blog, *networking*, forum, *multimedia streaming*, *e-commerce*, dan *e-learning*. Website banyak dimanfaatkan oleh organisasi atau perusahaan untuk mendukung proses bisnis mereka.

2.4 Proses Bisnis

Proses bisnis adalah kumpulan aktifitas yang saling berkoordinasi dalam sebuah organisasi untuk melakukan *input* dan menghasilkan *output* sesuai dengan tujuan (Weske, 2012). Aktifitas proses bisnis menjadi hal yang penting pada sebuah organisasi. Efektifitas dan efisiensi proses bisnis menjadi faktor penting dalam sebuah organisasi dalam menghadapi *competitor*. Teknologi informasi menjadi salah satu aspek penting yang harus dikembangkan dalam organisasi. Organisasi harus dapat manangani kecepatan arus informasi. Semakin cepat menerima informasi dan menyebarkan informasi, berguna untuk organisasi dalam mencapai

tujuannya. Dalam pemodelan proses bisnis suatu organisasi atau perusahaan, terdapat 2 macam yaitu pemodelan proses bisnis *as-is* dan pemodelan proses bisnis *to-be*.

Pemodelan proses bisnis *as-is* bertujuan untuk menggambarkan tentang bagaimana kondisi proses bisnis suatu organisasi atau perusahaan saat ini secara actual, serta untuk meningkatkan kekurangan – kekurangan yang masih terdapat pada proses bisnis saat ini (Becker, et al., 2013). Sedangkan pemodelan proses bisnis *to-be* bertujuan untuk menggambarkan dan mengidentifikasi proses bisnis sebelumnya untuk ditingkatkan lagi berdasarkan analisis yang telah dilakukan (Becker, et al., 2013).

2.4.1 Business Process Model and Notation

Business Process Model and Notation (BPMN) merupakan sebuah standar model proses bisnis yang disajikan menggunakan notasi untuk proses bisnis yang spesifik dengan berdasarkan bagan teknik bagan alur tradisional (Rosing, et al., 2015). Dengan adanya *BPMN* ini, sebuah organisasi dapat dengan mudah menggambarkan proses bisnisnya dengan model notasi kepada semua pihak atau stakeholder. Pada *BPMN* terdapat tentang siapa yang membuat dan menyaring proses. Kemudian pengembang akan mengimplementasikan proses bisnis yang ada. Proses bisnis akan terus dikelola dan dimonitor oleh manajer bisnis. Dengan demikian *BPMN* menjembatani jarak komunikasi yang terjadi antara desainer proses bisnis dan pengembang yang akan mengimplementasikan proses bisnis. Berikut adalah beberapa notasi yang digunakan pada *BPMN*:

1. Activity

Notasi ini berkaitan dengan sebuah representasi kegiatan atau aktivitas dalam proses bisnis. Pada Tabel 2.1 menjelaskan apa saja yang termasuk dalam notasi *activity*.

Table 2.1 BPMN Task Description

No	Name	Symbol	Description
1	None		Tidak ada tipe aktivitas yang terindikasi.
2	User Task		Aktivitas dimana manusia atau pengguna sudah berinteraksi langsung dengan perangkat lunak.
3	Manual Task		Aktivitas dimana manusia masih mengerjakan sesuatu secara manual atau tanpa bantuan sistem atau aplikasi.



Table 2.2 *BPMN Task Description (lanjutan)*

4	<i>Service Task</i>		Aktivitas otomatis yang dilakukan oleh sistem atau aplikasi atau bisa disebut <i>web service</i> .
5	<i>Recive Task</i>		Menunggu atau menerima pesan datang dari pihak eksternal yang masih ada hubungannya dengan proses tersebut.
6	<i>Send</i>		Mengirim pesan kepada pihak eksternal yang masih ada hubungannya dengan proses tersebut.
7	<i>Script</i>		Sebuah <i>script</i> yang dijalankan oleh mesin ketika aktivitas siap untuk dimulai.
8	<i>Business Rule</i>		Aktivitas yang menyediakan <i>input</i> dan mendapatkan kalkulasi <i>output</i> dari <i>business rule engine</i> .
9	<i>Sub-process</i>		Aktivitas dimana didalamnya terdapat aktivitas lain. Ini berguna untuk proses dekomposisi atau proses umum organisasi.

Sumber: Rosing et al. (2015)

2. Flow Object

Notasi ini berguna untuk menghubungkan antara notasi pada BPMN dengan notasi *flow object*. Pada Tabel 2.2 menjelaskan tentang jenis – jenis notasi *flow object*.

Table 2.3 BPMN Flow Object Description

No	Name	Symbol	Description
1	Sequence Flow		Berfungsi untuk menghubungkan antar aktivitas dalam satu proses atau diagram koreografi.
2	Message Flow		Berfungsi untuk mengirimkan pesan antar aktivitas yang berada dalam proses yang terpisah.
3	Association		Berfungsi untuk menghubungkan asosiasi teks dengan artifak lainnya.
4	Data Association		Berfungsi untuk menghubungkan aktivitas dalam penyimpanan data dalam sebuah <i>database</i> .

Sumber: Rosing et al. (2015)

3. Data Object

Notasi ini digunakan untuk mewakili sebuah data baik dalam bentuk dokumen, atau yang lainnya. Pada Tabel 2.3 menjelaskan tentang jenis – jenis *data object* dalam *BPMN*.

Table 2.4 BPMN Data Object Description

No	Name	Symbol	Description
1	Data Object		<i>Data object</i> digunakan untuk melakukan aktivitas input dan output serta dapat bersifat singular objek maupun kolektif objek.
2	Data Input		Data eksternal yang dimasukkan pada sebuah proses berupa parameter.
3	Data Output		Hasil data yang dimasukkan pada proses berupa parameter
4	Data Store		Tempat untuk menyimpan sebuah data.

Sumber: Rosing et al. (2015)

4. Event

Notasi ini digunakan untuk memicu terjadinya suatu aktivitas baik di awal, di tengah maupun di akhir. Pada Tabel 2.4 menjelaskan tentang jenis – jenis *event* pada *BPMN*.



Table 2.5 BPMN Event Description

No	Name	Symbol	Description
1	Start Event		Notasi yang menandakan dimana dimulainya sebuah proses
2	Intermediate and Boundary event		Notasi yang memicu sebuah aktivitas dimana notasi ini berada diantara <i>start event</i> dan <i>end event</i> .
3	End Event		Notasi ini menandakan bahwa proses sudah berakhir.
4	Message (receive)	 	Menerima pesan untuk memulai sebuah proses.
5	Message (send)	 	Mengirim pesan di tengah atau di akhir sebuah jalur proses.
6	Timer (catch)	 	Menunjukkan berapa lama waktu yang harus ditunggu untuk memulai sebuah aktivitas.

Sumber: Rosing et al. (2015)

5. Gateways

Notasi ini digunakan untuk menentukan alur berjalannya sebuah proses bisnis sesuai dengan kondisi tertentu. Pada Tabel 2.5 menjelaskan tentang jenis – jenis *gateway* pada BPMN.

Table 2.6 BPMN Gateway Description

No	Name	Symbol	Description
1	Gateway		Digunakan untuk mengontrol alur proses baik secara <i>converge</i> maupun <i>diverge</i> .
	Exclusive Gateway		Digunakan untuk menentukan hanya satu cabang saja yang dipakai, sesuai dengan kondisi.
3	Inclusive Gateway		Memungkinkan hanya satu cabang saja yang akan diaktifkan, dengan syarat aktivitas sebelumnya harus sudah selesai dijalankan.



Table 2.5 BPMN Gateway Description (lanjutan)

4	Parallel Gateway		Dapat menjalankan cabang secara bersamaan.
---	------------------	---	--

Sumber: Rosing et al. (2015)

2.5 Unified Modeling Language

Unified Modeling Language (UML) adalah sebuah bahasa pemodelan standar untuk memvisualisasikan, merancang dan mengimplementasikan artefak dari analisa dan pengembangan perangkat lunak. *UML* juga digunakan untuk mendeskripsikan dan mendokumentasikan proses bisnis.

UML dapat diterapkan pada beragram domain aplikasi, seperti perbankan, keuangan, internet, *e-commerce*, dll). Selain itu juga dapat digunakan dengan semua metode pengembangan perangkat keras dan komponen utama untuk berbagai *platform* implementasi, seperti J2EE dan .NET.

UML memiliki beberapa fungsi dan kegunaan, diantaranya:

1. Memberikan panduan mengenai urutan aktivitas sistem.
2. Menentukan artefak mana yang perlu dikembangkan.
3. Mengarahkan tugas masing – masing pengembang dan tim secara keseluruhan.

Menawarkan kriteria untuk memantau dan mengukur produk dan aktivitas proyek. Dengan adanya *UML* akan mempermudah pengembang dalam mengembangkan dan mengkomunikasikan analisis kebutuhan sistem kepada *stakeholder*.

2.5.1 Identifikasi Stakeholder

Sebelum membuat sebuah *Use Case*, identifikasi *stakeholder* harus dilakukan terlebih dahulu untuk mengetahui siapa sajakah pihak yang akan berpengaruh dengan adanya sistem informasi ini. Dengan adanya sistem yang dikembangkan berharap dapat menyelesaikan permasalahan yang terdapat pada *stakeholder*. Oleh karena itu, tim pengembang harus memiliki visi yang sama dengan *stakeholder* dalam mengembangkan sistem atau produk tertentu (Bitner & Spence, 2002).

Dengan membangun komunikasi dengan *stakeholder*, maka tim pengembang dapat memahami dengan baik tentang permasalahan yang ingin dipcahkan bersama baik oleh *stakeholder* maupun tim pengembang. Hal ini sangat berguna sebagai pondasi agar tidak terjadi masalah yang besar antara tim pengembang dan *stakeholder*. *Stakeholder* juga akan membantu tim pengembang dalam menentukan kebutuhan dan fitur – fitur pada produk yang dikembangkan.

Untuk mengidentifikasi bagaimana tipe – tipe tiap *stakeholder*, maka perlu dideskripsikan secara jelas. Ketika jumlah *stakeholder* sangat besar, tidak mungkin jika dilibatkan semua dalam sebuah proyek. Maka dari itu ditentukan beberapa

tipe sebagai perwakilan tiap *stakeholder*. Bittner & Spence mendefinisikan tipe *stakeholder* menjadi 3 bagian, yaitu *name*, *brief description*, dan *stakeholder representative*. *Name* merupakan nama tipe dari tiap *stakeholder*. *Brief description* merupakan deskripsi singkat mengenai tanggungjawab apa sajakah yang dimiliki *stakeholder* terhadap proyek. *Stakeholder representative* merupakan ringkasan bagaimana *stakeholder* akan direpresentasikan dalam proyek. Identifikasi tipe – tipe *stakeholder* tersebut akan ditampilkan pada tabel seperti berikut.

Table 2.6 Format Identifikasi Stakeholder

Name	Brief Description	Stakeholder Representative
Berisikan nama tipe dari tiap <i>stakeholder</i> .	Berisikan deskripsi singkat mengenai tanggungjawab apa sajakah yang dimiliki <i>stakeholder</i> terhadap proyek.	Berisikan ringkasan bagaimana <i>stakeholder</i> akan direpresentasikan dalam proyek.

Sumber: Bittner & Spence (2002)

2.5.2 Analisis Permasalahan

Untuk memahami sebuah permasalahan yang dialami oleh *stakeholder*, maka dibutuhkan analisis permasalahan. Dengan mengetahui permasalahan – permasalahan yang terdapat atau yang dialami oleh *stakeholder*, maka tim pengembang akan dapat mengidentifikasi solusi – solusi yang nantinya menjadi sebuah fitur untuk menyelesaikan masalah yang ada.

Agar analisis permasalahan dapat dipahami dengan baik oleh tiap *stakeholder*, maka dibutuhkan sebuah *problem statement* (Bittner & Spence, 2002). Untuk mendokumentasikan analisis permasalahan tersebut, Bittner & Spence membuat sebuah format *problem statement* seperti pada Tabel 2.7 berikut.

Table 2.7 Format Dokumentasi Analisis Permasalahan

The problem of	Berisikan tentang pendeskripsian masalah yang dialami saat ini.
Affect	Berisikan pengaruh apa yang akan terjadi pada <i>stakeholder</i> .
The Impact of which	Berisikan tentang dampak apa saja yang akan terjadi pada <i>stakeholder</i> .
A Successful solution would	Berisikan uraian solusi dari permasalahan yang ada.

Sumber; Bittner & Spence (2003)



2.5.3 Analisis Kebutuhan

Dalam melakukan analisis kebutuhan, melibatkan beberapa jenis orang atau yang disebut *stakeholder*. Sebagai seorang *stakeholder* harus memiliki pengaruh besar terhadap sistem yang dikembangkan. Analisis permasalahan juga digunakan sebagai acuan dalam menentukan kebutuhan – kebutuhan *stakeholder*. Dalam melakukan analisis kebutuhan, Sommerville membaginya menjadi 4 tahap sebagai berikut:

1. *Requirements discovery*, proses terjadinya interaksi dengan *stakeholder* untuk mengetahui apa yang menjadi kebutuhan mereka. Selain itu juga terdapat proses observasi dan studi dokumentasi untuk mengetahui lebih lanjut terkait kebutuhan *stakeholder*.
2. *Requirements classification and organization*, proses mengumpulkan hasil dari penemuan kebutuhan untuk yang masih belum terstruktur untuk dikelompokkan dan diidentifikasi.
3. *Requirements prioritization and negotiation*, proses dalam memprioritaskan kebutuhan mana yang harus didahulukan. Hal ini disebabkan adanya beberapa kebutuhan yang berbeda dari tiap tipe *stakeholder*.
4. *Requirements specification*, proses mem-*break down* kebutuhan – kebutuhan *stakeholder* menjadi lebih spesifik.

Dalam memprioritaskan sebuah kebutuhan, terdapat sebuah teknik dalam melakukan prioritas kebutuhan. Salah satu teknik yang dapat digunakan dalam proses prioritas kebutuhan adalah aturan MoSCow. Dalam aturan MoSCow ini, Slepton membaginya menjadi 4 jenis prioritas, yaitu:

1. Mo (Must Have)

Kebutuhan yang harus diprioritaskan pertama kali untuk segera diselesaikan agar sistem yang dikembangkan dapat berjalan dengan baik. Prioritas ini harus dipenuhi karena jika kebutuhan utama tidak terpenuhi maka akan berpengaruh kepada kebutuhan yang lain.

2. S (Should Have)

Prioritas ini memiliki arti bahwa kebutuhan tersebut bukanlah yang utama, namun memiliki nilai untuk *stakeholder*.

3. C (Could Have)

Prioritas ini memiliki arti bahwa kebutuhan tersebut dapat meningkatkan kepuasan *stakeholder*, namun jika tidak diimplementasikan juga tidak pengaruh terhadap proyek.

4. Won't Have (W)

Prioritas ini memiliki arti bahwa kebutuhan dapat diimplementasikan bahkan setelah selesainya proyek.

2.5.4 Identifikasi Fitur

Sebuah fasilitas atau layanan yang disediakan oleh sebuah sistem untuk memenuhi kebutuhan pengguna disebut fitur (Leffingwell & Widrig, 2003). Dengan adanya fitur, maka pengembang dapat mendeskripsikan kebutuhan pengguna dengan lebih ringkas dan tidak terlalu detail. Fitur yang dibuat harus memiliki ringkasan yang umum sehingga dapat dipahami dengan mudah. Fitur dapat berupa kebutuhan fungsional maupun non-fungsional. Pada Tabel 2.8 merupakan *framework* penulisan fitur yang terdiri dari *feature code*, *name*, *description*, dan *priority*.

Table 2.8 Format Identifikasi Fitur

<i>Feature Code</i>	<i>Name</i>	<i>Description</i>	<i>Priority</i>
Berisikan kode tiap fitur.	Berisikan nama fitur.	Berisikan deskripsi fitur.	Berisikan prioritas fitur dengan aturan MoSCoW.

Sumber: Bittner & Spence (2003)

2.5.5 Spesifikasi Kebutuhan

Spesifikasi kebutuhan merupakan proses mendokumentasikan kebutuhan pengguna dan sistem secara mendetail (Sommerville, 2011). Sommerville menambahkan, hal ini dilakukan agar kebutuhan pengguna dan sistem dapat dipahami dengan jelas, mudah, konsisten, dan tidak ambigu. Kebutuhan pengguna pada sistem dibagi menjadi dua jenis, yaitu kebutuhan fungsional dan kebutuhan non-fungsional.

Kebutuhan fungsional merupakan kebutuhan yang melayani pengguna berdasarkan masukan dari pengguna dan bagaimana sistem merespon hal tersebut (Sommerville, 2011). Sedangkan kebutuhan non-fungsional merupakan aspek yang berhubungan dengan *usability*, *performance*, *supportability*, dan *reliability* dari sistem yang dikembangkan (Leffingwell & Widrig, 2003).

2.5.6 Pemodelan Use Case

Pemodelan *Use Case* menggambarkan bagaimana interaksi antara aktor dengan sistem (Bittner & Spence, 2002). Model *Use Case* juga menggambarkan bagaimana cara yang memungkinkan dalam penggunaan sistem. Dasar dalam pemodelan *Use Case*, Bittner & Spence membaginya menjadi beberapa bagian, yaitu *actor*, *Use Cases*, *connection actors and Use Cases*, *brief description*, *Use Case descriptions*, *the flow of events*, *the basic flow*, *the alternative flows*, *preconditions*, dan *postcondition*.

Actor merupakan pengguna yang berinteraksi secara langsung dengan sistem. Sistem dapat berupa orang atau sistem lain. *Use Case* merupakan deskripsi tentang apa yang dilakukan aktor terhadap sistem. *Connecting actor and Use Case* merupakan komunikasi antara aktor dan sistem yang direpresentasikan dalam bentuk *Use Case*. *Brief description* merupakan pemaparan singkat tentang adanya

Use Case tersebut. *Use Case description* merupakan deskripsi dari *Use Case* tersebut. *The flow of event* merupakan kolaborasi sistem dan aktor dalam memberikan nilai pada *Use Case*. *The basic flow* merupakan langkah – langkah normal dalam menjalankan *Use Case*. *The alternative flow* merupakan langkah – langkah alternatif jika *basic flow* tidak berhasil dijalankan. *Precondition* merupakan kondisi awal sebelum *Use Case* dimulai. Terakhir *postcondition* merupakan kondisi akhir dari *Use Case*. Dalam pemodelan *Use Case scenario* akan dibuat seperti *framework* pada Tabel 2.9 berikut.

Table 2.9 Format Use Case Scenario

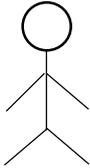
Item	Description
Use Case Code	Berisikan kode <i>Use Case</i>
Use Case Name	Berisikan nama <i>Use Case</i>
Actor	Berisikan nama aktor
Brief Description	Berisikan deskripsi singkat tentang <i>Use Case</i>
Precondition	Berisikan kondisi awal <i>Use Case</i>
Basic Flow	Berisikan langkah – langkah normal dalam menjalankan <i>Use Case</i>
Alternative Flow	Berisikan langkah – langkah alternatif jika <i>basic flow</i> gagal
Postondition	Berisikan kondisi akhir iUse Case

Sumber: Bittner & Spence (2002)

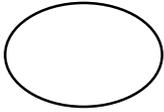
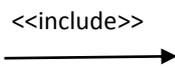
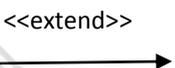
2.5.7 Use Case Diagram

Diagram *Use Case* digunakan untuk menggambarkan satu set tindakan dari sebuah atau beberapa sistem yang berinteraksi dengan pengguna sistem atau aktor. *Use Case diagram* juga digunakan untuk mengetahui apa saja fungsi yang terdapat di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi – fungsi tersebut (Rosa & Shalahudin, 2016). Setiap kasus penggunaan harus memberikan beberapa hasil yang layak dan dapat diamati kepada aktor maupun *stakeholder* lain dari sistem tersebut. Berikut pada Tabel 2.10 merupakan notasi pada *Use Case diagram*.

Table 2.10 Notasi pada Use Case Diagram

No	Elemen	Fungsi	Notasi
1	<i>Actor</i>	Menggambarkan tokoh atau pengguna yang mempengaruhi sistem, dan berada di luar sistem. Aktor biasanya ditempatkan di luar <i>subject boundary</i> .	

Tabel 2.10 Notasi pada *Use Case Diagram* (lanjutan)

No	Elemen	Fungsi	Notasi
2	<i>Use Case</i>	Mewakili fungsionalitas sebuah sistem, dan diletakkan di dalam <i>boundary sistem</i> .	
3	<i>Association Relationship</i>	Sebagai penghubung antara aktor dengan <i>Use Case</i> .	
4	<i>Include Relationship</i>	Menunjukkan inclusion fungsionalitas dari sebuah <i>Use Case</i> dengan <i>Use Case</i> yang lain. Arah panah dari <i>base Use Case</i> ke <i>included Use Case</i> .	
5	<i>Extend Relationship</i>	Menunjukkan keadaan yang hanya berjalan di bawah kondisi tertentu seperti menggerakkan alarm.	
6	<i>Generalization Relationship</i>	Menunjukkan generalisasi dari <i>Use Case</i> khusus ke <i>Use Case</i> yang lebih umum.	

Sumber: Rosa & Shalahuddin (2016)

2.5.8 Activity Diagram

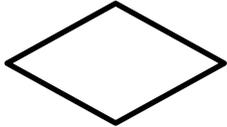
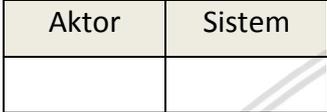
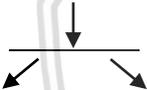
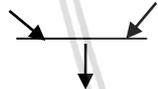
Activity diagram menunjukkan sebuah alur aktifitas yang dilakukan dalam proses bisnis dimana seluruh aktifitas akan dieksekusi dari awal hingga akhir (Rumbaugh, Jacobson, & Booch, 2004). Untuk memulai eksekusi akan dimulai dari node start dan akan berakhir hingga node end. Dalam prosesnya dapat terjadi percabangan dan penggabungan aktifitas. Berikut notasi yang biasa digunakan dalam *activity diagram* dijelaskan pada Tabel 2.11:

Table 2.11 Notasi pada *Activity Diagram*

Simbol	Nama	Deskripsi
	<i>Activity</i>	Berisi tentang aktifitas atau pekerjaan yang sedang dilakukan.
	<i>Start</i>	Sebagai tanda awal aktifitas akan mulai dijalankan.



Tabel 2.11 Notasi pada *Activity Diagram* (lanjutan)

Simbol	Nama	Deskripsi
	<i>End</i>	Sebagai tanda akhir aktifitas telah dijalankan semua.
	<i>Decision</i>	Percabangan apabila terjadi lebih dari satu pilihan pada aktifitas berikutnya.
	<i>Swimlane</i>	Pemisah antar organisasi atau aktor dalam suatu sistem.
	<i>Fork</i>	Digunakan ketika ada aktifitas yang akan dijalankan secara bersamaan.
	<i>Join</i>	Digunakan ketika ada aktifitas yang akan dijalankan secara bersama.

Sumber: Rosa & Shalahuddin (2016)

2.5.9 *Sequence Diagram*

Sequence diagram menunjukkan sebuah interaksi yang dilakukan oleh bagan dua dimensi (Rumbaugh, Jacobson, & Booch, 2004). Dua dimensi tersebut adalah dimensi vertikal dan dimensi horizontal. Dimana pada dimensi vertikal merepresentasikan sebuah waktu dan dimensi horizontal merepresentasikan sebuah obyek. Notasi yang ada pada *sequence diagram* beserta fungsi dan notasinya akan dijelaskan pada Tabel 2.12 berikut:

Table 2.12 Notasi Pada *Sequence Diagram*

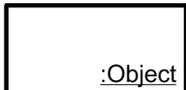
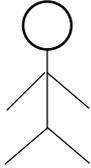
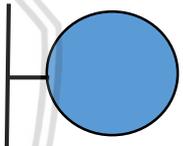
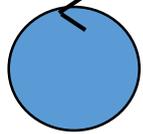
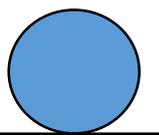
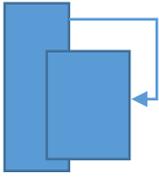
No.	Elemen	Fungsi	Notasi
1	<i>Object</i>	Digambarkan sebagai sebuah kelas dengan nama object didalamnya yang berinteraksi menggunakan pesan	



Table 2.12 Notasi Pada *Sequence Diagram* (lanjutan)

No.	Elemen	Fungsi	Notasi
2	<i>Actor</i>	Untuk berkomunikasi dengan object	
3	<i>Lifeline</i>	Mengidentifikasi keberadaan sebuah object dalam basis waktu. Notasi untuk lifeline adalah garis putus-putus vertikal yang ditarik dari sebuah object	
4	<i>Activation</i>	Mengidentifikasi sebuah object yang akan melakukan sebuah aksi.	
5	<i>Boundary</i>	Terletak diantara sistem dengan dunia sekelilingnya, semua <i>form</i> , laporan – laporan, antarmuka perangkat keras, seperti printer scanner, dan antarmuka ke sistem lain termasuk dalam kategori	
6	<i>Control</i>	Berhubungan dengan fungsionalitas seperti pemanfaatan sumber daya, pemrosesan distribusi atau penampilan kesalahan.	
7	<i>Entity</i>	Untuk menangani informasi yang mungkin akan disimpan secara permanen.	
8	<i>Message</i>	Untuk mengidentifikasi komunikasi antar object	
9	<i>Self-Message</i>	Untuk mengidentifikasi komunikasi yang kembali ke dalam sebuah object itu sendiri	

Sumber: Rosa & Shalahuddin (2016)



2.5.10 Class Diagram

Kelas merupakan deskripsi konsep dari domain aplikasi atau solusi aplikasi (Rumbaugh, Jacobson, & Booch, 2004). Kelas merupakan karakteristik khusus dari pengembangan perangkat lunak yang berorientasi obyek. Kelas menggambarkan atribut sebuah sistem serta metode – metode yang terkandung didalamnya.

Diagram kelas menggambarkan struktur dan deskripsi sebuah kelas, package dan obyek serta containment, inheritance, association dan lain – lain. Sebuah Kelas pasti memiliki atribut dan metode. Atribut dan metode memiliki salah satu sifat berikut:

- a. *Private*, tidak dapat dipanggil oleh kelas yang berada diluar dari kelas yang bersangkutan.
- b. *Protected*, hanya dipanggil dari kelas yang memiliki hubungan yang berkaitan dengan child dari kelas yang mewarisinya.
- c. *Public*, dapat dipanggil oleh dan dari kelas manapun dan kapanpun.

Hubungan antar kelas:

- a. Asosiasi, yaitu hubungan statis antar kelas yang menggambarkan bahwa kelas tersebut memiliki atribut yang ada di kelas lain atau kelas tersebut harus mengetahui eksistensi kelas lain.
- b. Agregasi, yaitu hubungan yang menyatakan bagian.

Pewarisan, yaitu hubungan hirarki antar kelas, dimana kelas tersebut diturunkan dari kelas lainnya serta ikut mewarisi atribut dan metode kelas asal. Sehingga ia disebut anak kelas. Kebalikan dari pewarisan adalah generalisasi.

2.6 Software Process Models

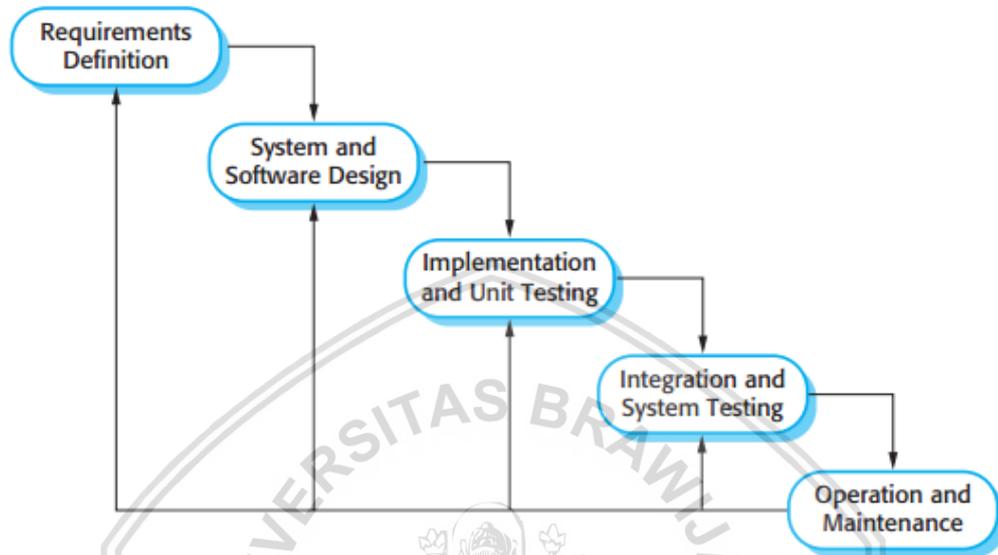
Software process model secara sederhana akan merepresentasikan proses dalam pengembangan perangkat lunak dimana setiap proses dibagi menjadi bagian – bagian yang spesifik dan setiap bagian mengandung informasi tentang proses yang dikerjakan (Sommerville, 2011). Model pada proses pengembangan perangkat lunak beragam, namun setidaknya setiap model pasti terdapat proses spesifikasi, desain, implementasi, validasi, dan evolusi. Selain itu proses pengembangan perangkat lunak juga terdapat roles, product, pre- dan post-condition. Roles merepresentasikan tanggungjawab setiap orang yang terlibat dalam pengembangan perangkat lunak.

Product adalah hasil dari aktifitas pengembangan. Pre- dan post-condition adalah statemen sebelum dan sesudah ketika produk akan digunakan. Salah satu model pengembangan perangkat lunak yang sering digunakan adalah waterfall. Selain itu ada incremental, reused-oriented, RUP, agile, dll.

2.6.1 Waterfall

Waterfall merupakan model pengembangan perangkat lunak yang menggunakan pendekatan sequential dalam proses pengembangannya

(Pressman, 2001). Pada dasarnya proses pengembangan perangkat lunak terdiri dari spesifikasi, pengembangan, validasi, dan evolusi, namun pada model ini dipecah lagi menjadi fase proses yang terpisah diantaranya analisis kebutuhan, desain dan implementasi, pengujian, dan perawatan (Sommerville, 2011). Berikut adalah model waterfall yang ditunjukkan pada Gambar 2.2.



Gambar 2.2 Alur Diagram Model Waterfall

Sumber: Sommerville (2011)

Penjelasan mengenai tahap – tahap dalam pengembangan *waterfall* adalah sebagai berikut:

1. Mendefinisikan Kebutuhan

Pengumpulan informasi dari *stakeholder* terkait kebutuhan dari sistem yang dikembangkan. Hal ini dapat dilakukan dengan melakukan komunikasi dengan *stakeholder* terkait. Spesifikasi kebutuhan harus didokumentasikan dengan baik agar mempermudah pada tahap selanjutnya.

2. Perancangan Sistem

Pada tahap ini sudah diketahui dengan jelas terkait kebutuhan pengguna. Selanjutnya dilakukan perancangan terkait struktur data, antarmuka pengguna, hingga prosedur pengkodean program. Tahap ini dijadikan acuan pada tahap implementasi.

3. Implementasi Sistem

Pada tahap ini akan terjadi proses pentranslasian dari perancangan menjadi kode program. Implementasi yang dibuat harus konsisten dengan perancangan yang sudah dibuat sebelumnya.

4. Pengujian Sistem

Setelah program selesai dibuat, selanjutnya program atau sistem yang kita bangun harus diuji baik secara fungsional maupun non-fungsional. Hal ini berguna untuk meminimalisir terjadinya kesalahan. Sehingga produk dapat dihasilkan sesuai dengan harapan.

5. Pendukung dan Pemeliharaan

Tidak menutup kemungkinan, sistem yang sudah selesai dikembangkan akan terjadi *error* atau pembaharuan pada suatu saat. Maka dari itu dibutuhkan *maintenance*.

Metode pengembangan *waterfall* merupakan salah satu metode pengembangan yang banyak digunakan dalam berbagai proyek. Hal ini dikarenakan metode ini memiliki beberapa kelebihan dibanding metode yang lainnya. Alshamrani & Bahattab dalam penelitiannya menyatakan beberapa kelebihan dari metode *waterfall* diantaranya:

- Metode ini mudah untuk dimengerti serta diimplementasikan bahkan untuk pemula.
- Metode ini sudah sering digunakan dan dikenal oleh banyak kalangan, yang artinya metode ini sudah terbukti kualitasnya.
- Pendefinisian dan perancangan yang jelas di awal akan memudahkan pengembang dalam proses implementasi.
- Metode ini cocok untuk tim yang bahkan baru masuk dalam proyek.
- Meminimalisir perencanaan yang berlebihan.
- Metode ini menuntut untuk disiplin dalam pengerjaannya pada tiap tahap.

2.7 Black-box Testing

Black-box testing adalah model pengujian perangkat lunak yang fokus pada pengujian spesifikasi kebutuhan sistem yang telah dirancang diawal, tanpa harus melakukan pengujian pada kode program (Sommerville, 2011). Pengujian ini bertujuan untuk memastikan apakah perangkat lunak yang kita buat telah memenuhi spesifikasi kebutuhan fungsional yang diharapkan stakeholder. Skenario pengujian dibutuhkan untuk memandu penguji dalam melaksanakan pengujian ini. *Black-box testing* bukan sebagai alternatif dari *white-box testing*, melainkan sebagai pelengkap. Pada pengujian *black-box* ini, peneliti menggunakan metode *Validation Testing*, *Compatibility Testing*, dan *User Acceptance Testing*.

2.7.1 Validation Testing

Validation testing yaitu teknik pengujian yang dilakukan untuk memastikan apakah sebuah sistem yang dikembangkan dapat berjalan dengan baik sesuai dengan harapan (Pressman, 2010). Dalam pengujian ini, hanya dilakukan pada kebutuhan

fungsional sistem. Sebuah perancangan pengujian dibutuhkan sebelum melakukan pengujian ini. Sebuah prosedur pengujian dispesifikan pada kasus uji yang didesain untuk memastikan semua kebutuhan fungsional dapat berjalan dengan baik. Dalam kasus uji terdapat tujuan dari pengujian, langkah – langkah pengujian, hasil yang diharapkan serta hasil dari pengujian itu sendiri.

2.7.2 Compatibility Testing

Untuk mengetahui apakah sistem yang dikembangkan dapat dijalankan pada beberapa lingkungan yang berbeda maka dibutuhkan pengujian yang disebut dengan *compatibility testing*. Pengujian jenis ini juga bertujuan untuk memastikan apakah *script* yang digunakan untuk mengembangkan suatu sistem informasi dapat bekerja dengan baik pada konfigurasi lingkungan yang berbeda (Pressman, 2010). *Browser* yang berbeda biasanya menghasilkan hasil yang sedikit berbeda, terlepas dari tingkat standarisasi HTML dalam WebApp.

2.7.3 User Acceptance Testing

User Acceptance Testing (UAT) adalah pengujian yang dilakukan oleh pengembang untuk mengetahui sejauh mana sistem dapat memenuhi kriteria kebutuhan pengguna, sehingga sistem yang kita buat benar – benar akan digunakan (Hambling, 2013). *UAT* akan memverifikasi apakah solusi yang dibuat sudah sesuai dengan yang diharapkan oleh pengguna. *UAT* pada pengembangan berorientasi obyek akan berpedoman dengan *Use Case* yang sudah dibuat. Terdapat lima jenis *UAT* yang digunakan pada saat pengujian (Panduan *User Acceptance Testing*, 2017):

1. *Black-box Testing*
2. *Regulation Acceptance Testing*
3. *Alpha & Beta Testing*
4. *Contract Acceptance Testing*
5. *Operational Acceptance Testing*

Pada penelitian ini akan menggunakan *UAT* jenis *black-box testing*. Karena jenis ini merupakan jenis pengujian yang berfokus pada kebutuhan fungsional sistem. Jenis *black-box testing* akan menganalisis fungsi tanpa harus memperhatikan struktur kode program. Karena yang terpenting dalam pengujian ini adalah sistem dapat memenuhi kebutuhan pengguna dan pengguna merasa terselesaikan masalahnya dengan adanya fungsi pada sistem ini.

2.7.3.1 Skala Linkert

Untuk menghubungkan sikap dan persepsi responden yang diperiksa dan dianalisis, maka salah satu cara yang dapat digunakan untuk menganalisis permasalahan tersebut yaitu dengan Skala *Likert* (McLeod, et al., 2011). Teknik ini dilakukan dengan cara meminta responden untuk mengisi kuisioner dengan penentuan skala yang sesuai dengan kriteria jawaban. Penilaian dari responden adalah sejauh mana mereka setuju dengan pertanyaan yang diajukan. Cara perhitungan *Skala Likert* dengan memberikan setiap jawaban bobot yang berbeda.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

TADICA (Taman Pendidikan Cendekiawan Muda) adalah lembaga bimbingan belajar yang berada di Samarinda Seberang, Kalimantan Timur. Lembaga ini memiliki beberapa program yang ditawarkan, yaitu paket baca, tulis dan menghitung, bimbingan belajar tingkat SD dan SMP, Persiapan ujian nasional SD dan SMP, serta JARITMATIKA. Pada penelitian yang dilakukan oleh Jannah, Astuti, dan Maharani (2015) yang berjudul “Rancang Bangun Sistem Informasi Bimbingan Belajar Berbasis Web (Studi Kasus: Lembaga Bimbingan Belajar TADICA)” mengemukakan bahwa dengan adanya sistem informasi bimbingan belajar berbasis web ini, dapat meningkatkan keefektifan siswa dan guru dalam proses belajar mengajar, sehingga proses bimbingan dapat berjalan dengan lebih cepat, tepat dan akurat.

Adapun modul yang terdiri dari fungsi – fungsi yang diimplementasikan pada sistem yang mereka kembangkan diantaranya pendaftaran *online*, penyimpanan data akademik, unduh materi atau soal, percakapan dengan siswa maupun admin melalui *chat room* dan dapat melakukan ujian *online*. Fungsi pendaftaran online berguna untuk siswa yang menginginkan program bimbingan belajar, sehingga ia dapat mendaftar secara *online*. Fungsi penyimpanan data berguna untuk memudahkan admin dalam mengelola data siswa yang terdaftar. Fungsi unduh materi atau soal berguna untuk memberikan fasilitas siswa yang ingin mempelajari materi ataupun soal yang telah dan akan diajarkan oleh guru. Fungsi chat room berguna untuk komunikasi antara siswa dan admin. Dan terakhir fungsi ujian online berguna bagi siswa yang ingin menguji kemampuannya.

Dari fungsi – fungsi yang dijabarkan di atas, memiliki beberapa kesamaan dan tujuan dengan fungsi – fungsi yang dibutuhkan pada sistem informasi Lembaga Bimbingan Belajar Al-Hasyimi. Beberapa fungsi diantaranya adalah penyimpanan akademik. Maka dari itu penelitian yang dilakukan oleh Jannah, Astuti, dan Maharani (2015) tersebut menjadi kajian pustaka dalam pengembangan sistem informasi Lembaga Bimbingan Belajar Al-Hasyimi ini.

Penelitian sebelumnya yang dilakukan oleh Mahalakshmi dan Sundararajan (2013) yang berjudul “*Traditional SDLC Vs Scrum Methodology – A Comparative Study*” mengungkapkan beberapa keuntungan menggunakan model pengembangan waterfall jika dibandingkan model pengembangan Scrum. Diantaranya adalah kebutuhan sudah jelas diawal, sehingga setiap fase pengembangan akan dilakukan secara berurutan dengan perkiraan waktu yang dapat ditentukan. Selain itu, model waterfall mudah untuk diimplementasikan dan cocok untuk pengembangan yang berorientasi pada proyek. *Waterfall* juga memberikan hasil dokumentasi yang lengkap dibanding dengan *Scrum*.

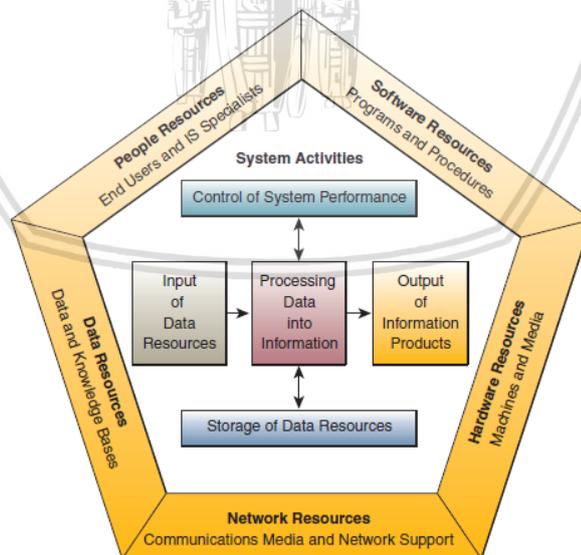
Kemudian penelitian tentang “Pemodelan *Use Case (UML)*: Evaluasi Terhadap Beberapa Kesalahan Dalam Praktik” yang dilakukan oleh Kurniawan. Dimana penelitian ini bertujuan untuk mengevaluasi beberapa kesalahan dalam

pembuatan *Use Case scenario* dan *Use Case diagram* yang sering terjadi baik dalam konteks pembelajaran di kampus maupun implementasi dalam dunia kerja. Penelitian ini berhasil mengidentifikasi 11 jenis kesalahan pembuatan *Use Case scenario* dan 7 kesalahan pembuatan *Use Case diagram*. Maka dari itu, penulis menggunakan penelitian ini untuk dijadikan acuan dalam pembuatan *Use Case scenario* dan *Use Case diagram* dalam pengembangan sistem informasi ini.

Selanjutnya penelitian yang dilakukan oleh Melnik, Read, dan Maurer yang berjudul "*Suitability of FIT User Acceptance Tests for Specifying Functional Requirements: Developer Perspective*" menyatakan bahwa metode *User Acceptance Testing (UAT)* ini diperlukan untuk memastikan kepuasan pelanggan dengan produk yang kita buat. UAT juga berfungsi sebagai regression testing, untuk memastikan bahwa fungsi kerja harus berjalan sesuai dengan yang diharapkan. Dengan ini maka spesifikasi persyaratan akan secara langsung mempengaruhi control kualitas.

2.2 Sistem Informasi

Menurut Obrien dan Marakas (2010) sistem adalah kumpulan elemen yang memiliki keterkaitan dan bekerja sama dalam menggapai tujuan sedangkan informasi adalah data yang sudah diproses sehingga bernilai manfaat bagi pengguna akhir. Maka dari itu yang dimaksud sistem informasi adalah sebuah pengorganisasian seperangkat manusia, komunikasi, data, jaringan, peraturan, prosedur, software dan hardware yang mana akan melakukan proses menyimpan, menerima, merubah dan menyebarkan informasi dalam sebuah organisasi (Obrien dan Marakas, 2010).



Gambar 2.1 Sistem Informasi

Sumber: O'Brien & Marakas (2010)



Di era modern ini, sistem informasi menjadi alat yang sangat penting bagi organisasi untuk meningkatkan efisiensi dan produktivitas yang lebih tinggi dalam proses bisnis sebuah organisasi (Laudon & Laudon, 2014). Ditengah persaingan bisnis yang semakin kompetitif, penting bagi organisasi ataupun perusahaan untuk memiliki sistem informasi sebagai salah satu strategi dalam memenangkan persaingan. Menurut O'Brien & Marakas (2010) terdapat tiga peran utama yang dimiliki sistem informasi dalam sebuah bisnis pada organisasi, diantaranya:

1. Sebagai pendukung dalam menjalankan proses dan operasi bisnis. Hampir semua toko retail sudah menggunakan sistem informasi berbasis komputer untuk membantu pegawainya dalam mencatat pembelian pelanggan, pengadaan produk, evaluasi penjualan, dan masih banyak lagi.
2. Sebagai pendukung dalam pengambilan keputusan dalam bisnis. Dengan adanya sistem informasi, membantu manajer dalam melakukan pengambilan keputusan yang tepat.

Sebagai strategi untuk memenangkan persaingan. Informasi yang sangat cepat dan mudah didapat, sangat berguna bagi sebuah organisasi untuk memanfaatkannya dalam meningkatkan kualitas proses bisnis untuk dapat bersaing dengan *competitor*.

2.3 Website

Website merupakan sekumpulan halaman web yang terhubung satu sama lain (Agung, 2000). Pertama ditemukan pada tahun 1980. Pada awalnya website digunakan hanya untuk melakukan pertukaran informasi dan memperbarui informasi untuk dibagikan kepada para peneliti di Prancis dan Swiss. Namun di era saat ini fungsi website sudah mulai meluas. Dermawan dan Permana (2013) menyebutkan terdapat lima fungsi website saat ini. Diantaranya sebagai media promosi, pemasaran, informasi, pendidikan, dan komunikasi.

Jenis – jenis website juga semakin beragam seiring berkembangnya teknologi serta kebutuhan manusia. Dermawan dan Permana (2013) kembali menyebutkan beberapa jenis website yang populer saat ini. Diantaranya website search engine, portal berita, blog, *networking*, forum, *multimedia streaming*, *e-commerce*, dan *e-learning*. Website banyak dimanfaatkan oleh organisasi atau perusahaan untuk mendukung proses bisnis mereka.

2.4 Proses Bisnis

Proses bisnis adalah kumpulan aktifitas yang saling berkoordinasi dalam sebuah organisasi untuk melakukan *input* dan menghasilkan *output* sesuai dengan tujuan (Weske, 2012). Aktifitas proses bisnis menjadi hal yang penting pada sebuah organisasi. Efektifitas dan efisiensi proses bisnis menjadi faktor penting dalam sebuah organisasi dalam menghadapi *competitor*. Teknologi informasi menjadi salah satu aspek penting yang harus dikembangkan dalam organisasi. Organisasi harus dapat manangani kecepatan arus informasi. Semakin cepat menerima informasi dan menyebarkan informasi, berguna untuk organisasi dalam mencapai

tujuannya. Dalam pemodelan proses bisnis suatu organisasi atau perusahaan, terdapat 2 macam yaitu pemodelan proses bisnis *as-is* dan pemodelan proses bisnis *to-be*.

Pemodelan proses bisnis *as-is* bertujuan untuk menggambarkan tentang bagaimana kondisi proses bisnis suatu organisasi atau perusahaan saat ini secara actual, serta untuk meningkatkan kekurangan – kekurangan yang masih terdapat pada proses bisnis saat ini (Becker, et al., 2013). Sedangkan pemodelan proses bisnis *to-be* bertujuan untuk menggambarkan dan mengidentifikasi proses bisnis sebelumnya untuk ditingkatkan lagi berdasarkan analisis yang telah dilakukan (Becker, et al., 2013).

2.4.1 Business Process Model and Notation

Business Process Model and Notation (BPMN) merupakan sebuah standar model proses bisnis yang disajikan menggunakan notasi untuk proses bisnis yang spesifik dengan berdasarkan bagan teknik bagan alur tradisional (Rosing, et al., 2015). Dengan adanya *BPMN* ini, sebuah organisasi dapat dengan mudah menggambarkan proses bisnisnya dengan model notasi kepada semua pihak atau stakeholder. Pada *BPMN* terdapat tentang siapa yang membuat dan menyaring proses. Kemudian pengembang akan mengimplementasikan proses bisnis yang ada. Proses bisnis akan terus dikelola dan dimonitor oleh manajer bisnis. Dengan demikian *BPMN* menjembatani jarak komunikasi yang terjadi antara desainer proses bisnis dan pengembang yang akan mengimplementasikan proses bisnis. Berikut adalah beberapa notasi yang digunakan pada *BPMN*:

1. *Activity*

Notasi ini berkaitan dengan sebuah representasi kegiatan atau aktivitas dalam proses bisnis. Pada Tabel 2.1 menjelaskan apa saja yang termasuk dalam notasi *activity*.

Table 2.1 BPMN Task Description

No	Name	Symbol	Description
1	<i>None</i>		Tidak ada tipe aktivitas yang terindikasi.
2	<i>User Task</i>		Aktivitas dimana manusia atau pengguna sudah berinteraksi langsung dengan perangkat lunak.
3	<i>Manual Task</i>		Aktivitas dimana manusia masih mengerjakan sesuatu secara manual atau tanpa bantuan sistem atau aplikasi.



Table 2.2 *BPMN Task Description (lanjutan)*

4	<i>Service Task</i>		Aktivitas otomatis yang dilakukan oleh sistem atau aplikasi atau bisa disebut <i>web service</i> .
5	<i>Recive Task</i>		Menunggu atau menerima pesan datang dari pihak eksternal yang masih ada hubungannya dengan proses tersebut.
6	<i>Send</i>		Mengirim pesan kepada pihak eksternal yang masih ada hubungannya dengan proses tersebut.
7	<i>Script</i>		Sebuah <i>script</i> yang dijalankan oleh mesin ketika aktivitas siap untuk dimulai.
8	<i>Business Rule</i>		Aktivitas yang menyediakan <i>input</i> dan mendapatkan kalkulasi <i>output</i> dari <i>business rule engine</i> .
9	<i>Sub-process</i>		Aktivitas dimana didalamnya terdapat aktivitas lain. Ini berguna untuk proses dekomposisi atau proses umum organisasi.

Sumber: Rosing et al. (2015)

2. Flow Object

Notasi ini berguna untuk menghubungkan antara notasi pada BPMN dengan notasi *flow object*. Pada Tabel 2.2 menjelaskan tentang jenis – jenis notasi *flow object*.

Table 2.3 BPMN Flow Object Description

No	Name	Symbol	Description
1	<i>Sequence Flow</i>		Berfungsi untuk menghubungkan antar aktivitas dalam satu proses atau diagram koreografi.
2	<i>Message Flow</i>		Berfungsi untuk mengirimkan pesan antar aktivitas yang berada dalam proses yang terpisah.
3	<i>Association</i>		Berfungsi untuk menghubungkan asosiasi teks dengan artifak lainnya.
4	<i>Data Association</i>		Berfungsi untuk menghubungkan aktivitas dalam penyimpanan data dalam sebuah <i>database</i> .

Sumber: Rosing et al. (2015)

3. Data Object

Notasi ini digunakan untuk mewakili sebuah data baik dalam bentuk dokumen, atau yang lainnya. Pada Tabel 2.3 menjelaskan tentang jenis – jenis *data object* dalam *BPMN*.

Table 2.4 BPMN Data Object Description

No	Name	Symbol	Description
1	<i>Data Object</i>		<i>Data object</i> digunakan untuk melakukan aktivitas input dan output serta dapat bersifat singular objek maupun kolektif objek.
2	<i>Data Input</i>		Data eksternal yang dimasukkan pada sebuah proses berupa parameter.
3	<i>Data Output</i>		Hasil data yang dimasukkan pada proses berupa parameter
4	<i>Data Store</i>		Tempat untuk menyimpan sebuah data.

Sumber: Rosing et al. (2015)

4. Event

Notasi ini digunakan untuk memicu terjadinya suatu aktivitas baik di awal, di tengah maupun di akhir. Pada Tabel 2.4 menjelaskan tentang jenis – jenis *event* pada *BPMN*.



Table 2.5 BPMN Event Description

No	Name	Symbol	Description
1	Start Event		Notasi yang menandakan dimana dimulainya sebuah proses
2	Intermediate and Boundary event		Notasi yang memicu sebuah aktivitas dimana notasi ini berada diantara <i>start event</i> dan <i>end event</i> .
3	End Event		Notasi ini menandakan bahwa proses sudah berakhir.
4	Message (receive)	 	Menerima pesan untuk memulai sebuah proses.
5	Message (send)	 	Mengirim pesan di tengah atau di akhir sebuah jalur proses.
6	Timer (catch)	 	Menunjukkan berapa lama waktu yang harus ditunggu untuk memulai sebuah aktivitas.

Sumber: Rosing et al. (2015)

5. Gateways

Notasi ini digunakan untuk menentukan alur berjalannya sebuah proses bisnis sesuai dengan kondisi tertentu. Pada Tabel 2.5 menjelaskan tentang jenis – jenis *gateway* pada BPMN.

Table 2.6 BPMN Gateway Description

No	Name	Symbol	Description
1	Gateway		Digunakan untuk mengontrol alur proses baik secara <i>converge</i> maupun <i>diverge</i> .
	Exclusive Gateway		Digunakan untuk menentukan hanya satu cabang saja yang dipakai, sesuai dengan kondisi.
3	Inclusive Gateway		Memungkinkan hanya satu cabang saja yang akan diaktifkan, dengan syarat aktivitas sebelumnya harus sudah selesai dijalankan.



Table 2.5 BPMN Gateway Description (lanjutan)

4	<i>Parallel Gateway</i>		Dapat menjalankan cabang secara bersamaan.
---	-------------------------	---	--

Sumber: Rosing et al. (2015)

2.5 Unified Modeling Language

Unified Modeling Language (UML) adalah sebuah bahasa pemodelan standar untuk memvisualisasikan, merancang dan mengimplementasikan artefak dari analisa dan pengembangan perangkat lunak. *UML* juga digunakan untuk mendeskripsikan dan mendokumentasikan proses bisnis.

UML dapat diterapkan pada beragram domain aplikasi, seperti perbankan, keuangan, internet, *e-commerce*, dll). Selain itu juga dapat digunakan dengan semua metode pengembangan perangkat keras dan komponen utama untuk berbagai *platform* implementasi, seperti J2EE dan .NET.

UML memiliki beberapa fungsi dan kegunaan, diantaranya:

1. Memberikan panduan mengenai urutan aktivitas sistem.
2. Menentukan artefak mana yang perlu dikembangkan.
3. Mengarahkan tugas masing – masing pengembang dan tim secara keseluruhan.

Menawarkan kriteria untuk memantau dan mengukur produk dan aktivitas proyek. Dengan adanya *UML* akan mempermudah pengembang dalam mengembangkan dan mengkomunikasikan analisis kebutuhan sistem kepada *stakeholder*.

2.5.1 Identifikasi Stakeholder

Sebelum membuat sebuah *Use Case*, identifikasi *stakeholder* harus dilakukan terlebih dahulu untuk mengetahui siapa sajakah pihak yang akan berpengaruh dengan adanya sistem informasi ini. Dengan adanya sistem yang dikembangkan berharap dapat menyelesaikan permasalahan yang terdapat pada *stakeholder*. Oleh karena itu, tim pengembang harus memiliki visi yang sama dengan *stakeholder* dalam mengembangkan sistem atau produk tertentu (Bitner & Spence, 2002).

Dengan membangun komunikasi dengan *stakeholder*, maka tim pengembang dapat memahami dengan baik tentang permasalahan yang ingin dipcahkan bersama baik oleh *stakeholder* maupun tim pengembang. Hal ini sangat berguna sebagai pondasi agar tidak terjadi masalah yang besar antara tim pengembang dan *stakeholder*. *Stakeholder* juga akan membantu tim pengembang dalam menentukan kebutuhan dan fitur – fitur pada produk yang dikembangkan.

Untuk mengidentifikasi bagaimana tipe – tipe tiap *stakeholder*, maka perlu dideskripsikan secara jelas. Ketika jumlah *stakeholder* sangat besar, tidak mungkin jika dilibatkan semua dalam sebuah proyek. Maka dari itu ditentukan beberapa

tipe sebagai perwakilan tiap *stakeholder*. Bittner & Spence mendefinisikan tipe *stakeholder* menjadi 3 bagian, yaitu *name*, *brief description*, dan *stakeholder representative*. *Name* merupakan nama tipe dari tiap *stakeholder*. *Brief description* merupakan deskripsi singkat mengenai tanggungjawab apa sajakah yang dimiliki *stakeholder* terhadap proyek. *Stakeholder representative* merupakan ringkasan bagaimana *stakeholder* akan direpresentasikan dalam proyek. Identifikasi tipe – tipe *stakeholder* tersebut akan ditampilkan pada tabel seperti berikut.

Table 2.6 Format Identifikasi Stakeholder

Name	Brief Description	Stakeholder Representative
Berisikan nama tipe dari tiap <i>stakeholder</i> .	Berisikan deskripsi singkat mengenai tanggungjawab apa sajakah yang dimiliki <i>stakeholder</i> terhadap proyek.	Berisikan ringkasan bagaimana <i>stakeholder</i> akan direpresentasikan dalam proyek.

Sumber: Bittner & Spence (2002)

2.5.2 Analisis Permasalahan

Untuk memahami sebuah permasalahan yang dialami oleh *stakeholder*, maka dibutuhkan analisis permasalahan. Dengan mengetahui permasalahan – permasalahan yang terdapat atau yang dialami oleh *stakeholder*, maka tim pengembang akan dapat mengidentifikasi solusi – solusi yang nantinya menjadi sebuah fitur untuk menyelesaikan masalah yang ada.

Agar analisis permasalahan dapat dipahami dengan baik oleh tiap *stakeholder*, maka dibutuhkan sebuah *problem statement* (Bittner & Spence, 2002). Untuk mendokumentasikan analisis permasalahan tersebut, Bittner & Spence membuat sebuah format *problem statement* seperti pada Tabel 2.7 berikut.

Table 2.7 Format Dokumentasi Analisis Permasalahan

The problem of	Berisikan tentang pendeskripsian masalah yang dialami saat ini.
Affect	Berisikan pengaruh apa yang akan terjadi pada <i>stakeholder</i> .
The Impact of which	Berisikan tentang dampak apa saja yang akan terjadi pada <i>stakeholder</i> .
A Successful solution would	Berisikan uraian solusi dari permasalahan yang ada.

Sumber; Bittner & Spence (2003)



2.5.3 Analisis Kebutuhan

Dalam melakukan analisis kebutuhan, melibatkan beberapa jenis orang atau yang disebut *stakeholder*. Sebagai seorang *stakeholder* harus memiliki pengaruh besar terhadap sistem yang dikembangkan. Analisis permasalahan juga digunakan sebagai acuan dalam menentukan kebutuhan – kebutuhan *stakeholder*. Dalam melakukan analisis kebutuhan, Sommerville membaginya menjadi 4 tahap sebagai berikut:

1. *Requirements discovery*, proses terjadinya interaksi dengan *stakeholder* untuk mengetahui apa yang menjadi kebutuhan mereka. Selain itu juga terdapat proses observasi dan studi dokumentasi untuk mengetahui lebih lanjut terkait kebutuhan *stakeholder*.
2. *Requirements classification and organization*, proses mengumpulkan hasil dari penemuan kebutuhan untuk yang masih belum terstruktur untuk dikelompokkan dan diidentifikasi.
3. *Requirements prioritization and negotiation*, proses dalam memprioritaskan kebutuhan mana yang harus didahulukan. Hal ini disebabkan adanya beberapa kebutuhan yang berbeda dari tiap tipe *stakeholder*.
4. *Requirements specification*, proses mem-*break down* kebutuhan – kebutuhan *stakeholder* menjadi lebih spesifik.

Dalam memprioritaskan sebuah kebutuhan, terdapat sebuah teknik dalam melakukan prioritas kebutuhan. Salah satu teknik yang dapat digunakan dalam proses prioritas kebutuhan adalah aturan MoSCow. Dalam aturan MoSCow ini, Slepton membaginya menjadi 4 jenis prioritas, yaitu:

1. Mo (Must Have)

Kebutuhan yang harus diprioritaskan pertama kali untuk segera diselesaikan agar sistem yang dikembangkan dapat berjalan dengan baik. Prioritas ini harus dipenuhi karena jika kebutuhan utama tidak terpenuhi maka akan berpengaruh kepada kebutuhan yang lain.

2. S (Should Have)

Prioritas ini memiliki arti bahwa kebutuhan tersebut bukanlah yang utama, namun memiliki nilai untuk *stakeholder*.

3. C (Could Have)

Prioritas ini memiliki arti bahwa kebutuhan tersebut dapat meningkatkan kepuasan *stakeholder*, namun jika tidak diimplementasikan juga tidak pengaruh terhadap proyek.

4. Won't Have (W)

Prioritas ini memiliki arti bahwa kebutuhan dapat diimplementasikan bahkan setelah selesainya proyek.

2.5.4 Identifikasi Fitur

Sebuah fasilitas atau layanan yang disediakan oleh sebuah sistem untuk memenuhi kebutuhan pengguna disebut fitur (Leffingwell & Widrig, 2003). Dengan adanya fitur, maka pengembang dapat mendeskripsikan kebutuhan pengguna dengan lebih ringkas dan tidak terlalu detail. Fitur yang dibuat harus memiliki ringkasan yang umum sehingga dapat dipahami dengan mudah. Fitur dapat berupa kebutuhan fungsional maupun non-fungsional. Pada Tabel 2.8 merupakan *framework* penulisan fitur yang terdiri dari *feature code*, *name*, *description*, dan *priority*.

Table 2.8 Format Identifikasi Fitur

Feature Code	Name	Description	Priority
Berisikan kode tiap fitur.	Berisikan nama fitur.	Berisikan deskripsi fitur.	Berisikan prioritas fitur dengan aturan MoSCoW.

Sumber: Bittner & Spence (2003)

2.5.5 Spesifikasi Kebutuhan

Spesifikasi kebutuhan merupakan proses mendokumentasikan kebutuhan pengguna dan sistem secara mendetail (Sommerville, 2011). Sommerville menambahkan, hal ini dilakukan agar kebutuhan pengguna dan sistem dapat dipahami dengan jelas, mudah, konsisten, dan tidak ambigu. Kebutuhan pengguna pada sistem dibagi menjadi dua jenis, yaitu kebutuhan fungsional dan kebutuhan non-fungsional.

Kebutuhan fungsional merupakan kebutuhan yang melayani pengguna berdasarkan masukan dari pengguna dan bagaimana sistem merespon hal tersebut (Sommerville, 2011). Sedangkan kebutuhan non-fungsional merupakan aspek yang berhubungan dengan *usability*, *performance*, *supportability*, dan *reliability* dari sistem yang dikembangkan (Leffingwell & Widrig, 2003).

2.5.6 Pemodelan Use Case

Pemodelan *Use Case* menggambarkan bagaimana interaksi antara aktor dengan sistem (Bittner & Spence, 2002). Model *Use Case* juga menggambarkan bagaimana cara yang memungkinkan dalam penggunaan sistem. Dasar dalam pemodelan *Use Case*, Bittner & Spence membaginya menjadi beberapa bagian, yaitu *actor*, *Use Cases*, *connection actors and Use Cases*, *brief description*, *Use Case descriptions*, *the flow of events*, *the basic flow*, *the alternative flows*, *preconditions*, dan *postcondition*.

Actor merupakan pengguna yang berinteraksi secara langsung dengan sistem. Sistem dapat berupa orang atau sistem lain. *Use Case* merupakan deskripsi tentang apa yang dilakukan aktor terhadap sistem. *Connecting actor and Use Case* merupakan komunikasi antara aktor dan sistem yang direpresentasikan dalam bentuk *Use Case*. *Brief description* merupakan pemaparan singkat tentang adanya

Use Case tersebut. *Use Case description* merupakan deskripsi dari *Use Case* tersebut. *The flow of event* merupakan kolaborasi sistem dan aktor dalam memberikan nilai pada *Use Case*. *The basic flow* merupakan langkah – langkah normal dalam menjalankan *Use Case*. *The alternative flow* merupakan langkah – langkah alternatif jika *basic flow* tidak berhasil dijalankan. *Precondition* merupakan kondisi awal sebelum *Use Case* dimulai. Terakhir *postcondition* merupakan kondisi akhir dari *Use Case*. Dalam pemodelan *Use Case scenario* akan dibuat seperti *framework* pada Tabel 2.9 berikut.

Table 2.9 Format Use Case Scenario

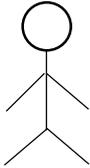
Item	Description
Use Case Code	Berisikan kode <i>Use Case</i>
Use Case Name	Berisikan nama <i>Use Case</i>
Actor	Berisikan nama aktor
Brief Description	Berisikan deskripsi singkat tentang <i>Use Case</i>
Precondition	Berisikan kondisi awal <i>Use Case</i>
Basic Flow	Berisikan langkah – langkah normal dalam menjalankan <i>Use Case</i>
Alternative Flow	Berisikan langkah – langkah alternatif jika <i>basic flow</i> gagal
Postondition	Berisikan kondisi akhir i <i>Use Case</i>

Sumber: Bittner & Spence (2002)

2.5.7 Use Case Diagram

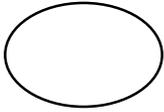
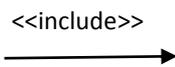
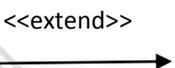
Diagram *Use Case* digunakan untuk menggambarkan satu set tindakan dari sebuah atau beberapa sistem yang berinteraksi dengan pengguna sistem atau aktor. *Use Case diagram* juga digunakan untuk mengetahui apa saja fungsi yang terdapat di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi – fungsi tersebut (Rosa & Shalahudin, 2016). Setiap kasus penggunaan harus memberikan beberapa hasil yang layak dan dapat diamati kepada aktor maupun *stakeholder* lain dari sistem tersebut. Berikut pada Tabel 2.10 merupakan notasi pada *Use Case diagram*.

Table 2.10 Notasi pada Use Case Diagram

No	Elemen	Fungsi	Notasi
1	<i>Actor</i>	Menggambarkan tokoh atau pengguna yang mempengaruhi sistem, dan berada di luar sistem. Aktor biasanya ditempatkan di luar <i>subject boundary</i> .	



Tabel 2.10 Notasi pada *Use Case Diagram* (lanjutan)

No	Elemen	Fungsi	Notasi
2	<i>Use Case</i>	Mewakili fungsionalitas sebuah sistem, dan diletakkan di dalam <i>boundary sistem</i> .	
3	<i>Association Relationship</i>	Sebagai penghubung antara aktor dengan <i>Use Case</i> .	
4	<i>Include Relationship</i>	Menunjukkan inclusion fungsionalitas dari sebuah <i>Use Case</i> dengan <i>Use Case</i> yang lain. Arah panah dari <i>base Use Case</i> ke <i>included Use Case</i> .	
5	<i>Extend Relationship</i>	Menunjukkan keadaan yang hanya berjalan di bawah kondisi tertentu seperti menggerakkan alarm.	
6	<i>Generalization Relationship</i>	Menunjukkan generalisasi dari <i>Use Case</i> khusus ke <i>Use Case</i> yang lebih umum.	

Sumber: Rosa & Shalahuddin (2016)

2.5.8 Activity Diagram

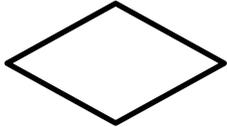
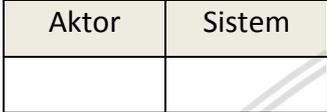
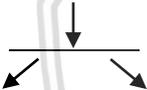
Activity diagram menunjukkan sebuah alur aktifitas yang dilakukan dalam proses bisnis dimana seluruh aktifitas akan dieksekusi dari awal hingga akhir (Rumbaugh, Jacobson, & Booch, 2004). Untuk memulai eksekusi akan dimulai dari node start dan akan berakhir hingga node end. Dalam prosesnya dapat terjadi percabangan dan penggabungan aktifitas. Berikut notasi yang biasa digunakan dalam *activity diagram* dijelaskan pada Tabel 2.11:

Table 2.11 Notasi pada *Activity Diagram*

Simbol	Nama	Deskripsi
	<i>Activity</i>	Berisi tentang aktifitas atau pekerjaan yang sedang dilakukan.
	<i>Start</i>	Sebagai tanda awal aktifitas akan mulai dijalankan.



Tabel 2.11 Notasi pada *Activity Diagram* (lanjutan)

Simbol	Nama	Deskripsi
	<i>End</i>	Sebagai tanda akhir aktifitas telah dijalankan semua.
	<i>Decision</i>	Percabangan apabila terjadi lebih dari satu pilihan pada aktifitas berikutnya.
	<i>Swimlane</i>	Pemisah antar organisasi atau aktor dalam suatu sistem.
	<i>Fork</i>	Digunakan ketika ada aktifitas yang akan dijalankan secara bersamaan.
	<i>Join</i>	Digunakan ketika ada aktifitas yang akan dijalankan secara bersama.

Sumber: Rosa & Shalahuddin (2016)

2.5.9 *Sequence Diagram*

Sequence diagram menunjukkan sebuah interaksi yang dilakukan oleh bagan dua dimensi (Rumbaugh, Jacobson, & Booch, 2004). Dua dimensi tersebut adalah dimensi vertikal dan dimensi horizontal. Dimana pada dimensi vertikal merepresentasikan sebuah waktu dan dimensi horizontal merepresentasikan sebuah obyek. Notasi yang ada pada *sequence diagram* beserta fungsi dan notasinya akan dijelaskan pada Tabel 2.12 berikut:

Table 2.12 Notasi Pada *Sequence Diagram*

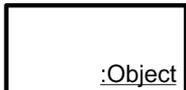
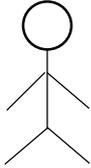
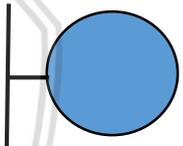
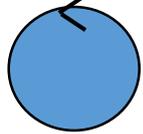
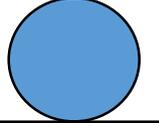
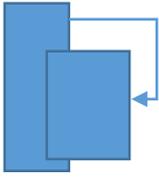
No.	Elemen	Fungsi	Notasi
1	<i>Object</i>	Digambarkan sebagai sebuah kelas dengan nama object didalamnya yang berinteraksi menggunakan pesan	



Table 2.12 Notasi Pada *Sequence Diagram* (lanjutan)

No.	Elemen	Fungsi	Notasi
2	<i>Actor</i>	Untuk berkomunikasi dengan object	
3	<i>Lifeline</i>	Mengidentifikasi keberadaan sebuah object dalam basis waktu. Notasi untuk lifeline adalah garis putus-putus vertikal yang ditarik dari sebuah object	
4	<i>Activation</i>	Mengidentifikasi sebuah object yang akan melakukan sebuah aksi.	
5	<i>Boundary</i>	Terletak diantara sistem dengan dunia sekelilingnya, semua <i>form</i> , laporan – laporan, antarmuka perangkat keras, seperti printer scanner, dan antarmuka ke sistem lain termasuk dalam kategori	
6	<i>Control</i>	Berhubungan dengan fungsionalitas seperti pemanfaatan sumber daya, pemrosesan distribusi atau penampilan kesalahan.	
7	<i>Entity</i>	Untuk menangani informasi yang mungkin akan disimpan secara permanen.	
8	<i>Message</i>	Untuk mengidentifikasi komunikasi antar object	
9	<i>Self-Message</i>	Untuk mengidentifikasi komunikasi yang kembali ke dalam sebuah object itu sendiri	

Sumber: Rosa & Shalahuddin (2016)



2.5.10 Class Diagram

Kelas merupakan deskripsi konsep dari domain aplikasi atau solusi aplikasi (Rumbaugh, Jacobson, & Booch, 2004). Kelas merupakan karakteristik khusus dari pengembangan perangkat lunak yang berorientasi obyek. Kelas menggambarkan atribut sebuah sistem serta metode – metode yang terkandung didalamnya.

Diagram kelas menggambarkan struktur dan deskripsi sebuah kelas, package dan obyek serta containment, inheritance, association dan lain – lain. Sebuah Kelas pasti memiliki atribut dan metode. Atribut dan metode memiliki salah satu sifat berikut:

- a. *Private*, tidak dapat dipanggil oleh kelas yang berada diluar dari kelas yang bersangkutan.
- b. *Protected*, hanya dipanggil dari kelas yang memiliki hubungan yang berkaitan dengan child dari kelas yang mewarisinya.
- c. *Public*, dapat dipanggil oleh dan dari kelas manapun dan kapanpun.

Hubungan antar kelas:

- a. Asosiasi, yaitu hubungan statis antar kelas yang menggambarkan bahwa kelas tersebut memiliki atribut yang ada di kelas lain atau kelas tersebut harus mengetahui eksistensi kelas lain.
- b. Agregasi, yaitu hubungan yang menyatakan bagian.

Pewarisan, yaitu hubungan hirarki antar kelas, dimana kelas tersebut diturunkan dari kelas lainnya serta ikut mewarisi atribut dan metode kelas asal. Sehingga ia disebut anak kelas. Kebalikan dari pewarisan adalah generalisasi.

2.6 Software Process Models

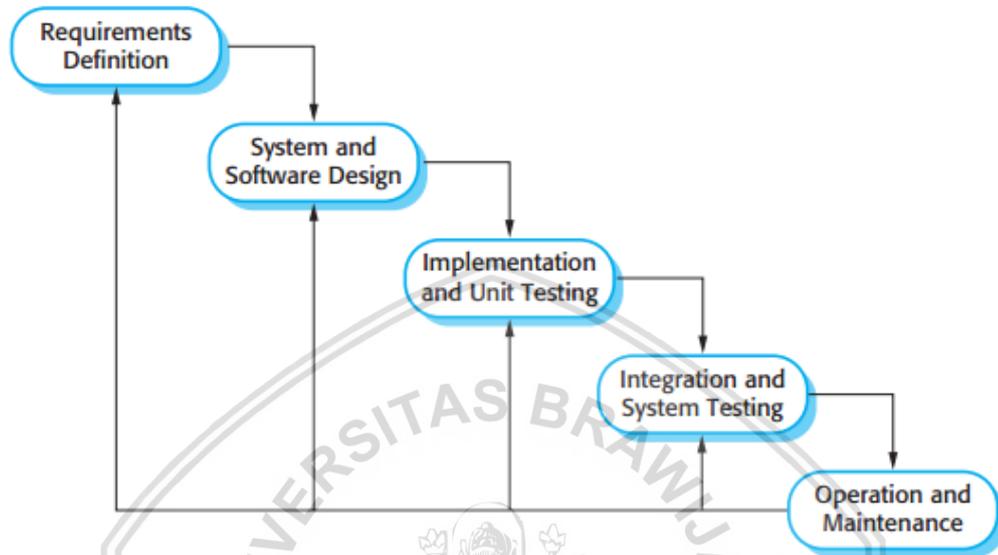
Software process model secara sederhana akan merepresentasikan proses dalam pengembangan perangkat lunak dimana setiap proses dibagi menjadi bagian – bagian yang spesifik dan setiap bagian mengandung informasi tentang proses yang dikerjakan (Sommerville, 2011). Model pada proses pengembangan perangkat lunak beragam, namun setidaknya setiap model pasti terdapat proses spesifikasi, desain, implementasi, validasi, dan evolusi. Selain itu proses pengembangan perangkat lunak juga terdapat roles, product, pre- dan post-condition. Roles merepresentasikan tanggungjawab setiap orang yang terlibat dalam pengembangan perangkat lunak.

Product adalah hasil dari aktifitas pengembangan. Pre- dan post-condition adalah statemen sebelum dan sesudah ketika produk akan digunakan. Salah satu model pengembangan perangkat lunak yang sering digunakan adalah waterfall. Selain itu ada incremental, reused-oriented, RUP, agile, dll.

2.6.1 Waterfall

Waterfall merupakan model pengembangan perangkat lunak yang menggunakan pendekatan sequential dalam proses pengembangannya

(Pressman, 2001). Pada dasarnya proses pengembangan perangkat lunak terdiri dari spesifikasi, pengembangan, validasi, dan evolusi, namun pada model ini dipecah lagi menjadi fase proses yang terpisah diantaranya analisis kebutuhan, desain dan implementasi, pengujian, dan perawatan (Sommerville, 2011). Berikut adalah model waterfall yang ditunjukkan pada Gambar 2.2.



Gambar 2.2 Alur Diagram Model Waterfall

Sumber: Sommerville (2011)

Penjelasan mengenai tahap – tahap dalam pengembangan *waterfall* adalah sebagai berikut:

1. Mendefinisikan Kebutuhan

Pengumpulan informasi dari *stakeholder* terkait kebutuhan dari sistem yang dikembangkan. Hal ini dapat dilakukan dengan melakukan komunikasi dengan *stakeholder* terkait. Spesifikasi kebutuhan harus didokumentasikan dengan baik agar mempermudah pada tahap selanjutnya.

2. Perancangan Sistem

Pada tahap ini sudah diketahui dengan jelas terkait kebutuhan pengguna. Selanjutnya dilakukan perancangan terkait struktur data, antarmuka pengguna, hingga prosedur pengkodean program. Tahap ini dijadikan acuan pada tahap implementasi.

3. Implementasi Sistem

Pada tahap ini akan terjadi proses pentranslasian dari perancangan menjadi kode program. Implementasi yang dibuat harus konsisten dengan perancangan yang sudah dibuat sebelumnya.



4. Pengujian Sistem

Setelah program selesai dibuat, selanjutnya program atau sistem yang kita bangun harus diuji baik secara fungsional maupun non-fungsional. Hal ini berguna untuk meminimalisir terjadinya kesalahan. Sehingga produk dapat dihasilkan sesuai dengan harapan.

5. Pendukung dan Pemeliharaan

Tidak menutup kemungkinan, sistem yang sudah selesai dikembangkan akan terjadi *error* atau pembaharuan pada suatu saat. Maka dari itu dibutuhkan *maintenance*.

Metode pengembangan *waterfall* merupakan salah satu metode pengembangan yang banyak digunakan dalam berbagai proyek. Hal ini dikarenakan metode ini memiliki beberapa kelebihan dibanding metode yang lainnya. Alshamrani & Bahattab dalam penelitiannya menyatakan beberapa kelebihan dari metode *waterfall* diantaranya:

- Metode ini mudah untuk dimengerti serta diimplementasikan bahkan untuk pemula.
- Metode ini sudah sering digunakan dan dikenal oleh banyak kalangan, yang artinya metode ini sudah terbukti kualitasnya.
- Pendefinisian dan perancangan yang jelas di awal akan memudahkan pengembang dalam proses implementasi.
- Metode ini cocok untuk tim yang bahkan baru masuk dalam proyek.
- Meminimalisir perencanaan yang berlebihan.
- Metode ini menuntut untuk disiplin dalam pengerjaannya pada tiap tahap.

2.7 Black-box Testing

Black-box testing adalah model pengujian perangkat lunak yang fokus pada pengujian spesifikasi kebutuhan sistem yang telah dirancang diawal, tanpa harus melakukan pengujian pada kode program (Sommerville, 2011). Pengujian ini bertujuan untuk memastikan apakah perangkat lunak yang kita buat telah memenuhi spesifikasi kebutuhan fungsional yang diharapkan stakeholder. Skenario pengujian dibutuhkan untuk memandu penguji dalam melaksanakan pengujian ini. *Black-box testing* bukan sebagai alternatif dari *white-box testing*, melainkan sebagai pelengkap. Pada pengujian *black-box* ini, peneliti menggunakan metode *Validation Testing*, *Compatibility Testing*, dan *User Acceptance Testing*.

2.7.1 Validation Testing

Validation testing yaitu teknik pengujian yang dilakukan untuk memastikan apakah sebuah sistem yang dikembangkan dapat berjalan dengan baik sesuai dengan harapan (Pressman, 2010). Dalam pengujian ini, hanya dilakukan pada kebutuhan

fungsional sistem. Sebuah perancangan pengujian dibutuhkan sebelum melakukan pengujian ini. Sebuah prosedur pengujian dispesifikan pada kasus uji yang didesain untuk memastikan semua kebutuhan fungsional dapat berjalan dengan baik. Dalam kasus uji terdapat tujuan dari pengujian, langkah – langkah pengujian, hasil yang diharapkan serta hasil dari pengujian itu sendiri.

2.7.2 Compatibility Testing

Untuk mengetahui apakah sistem yang dikembangkan dapat dijalankan pada beberapa lingkungan yang berbeda maka dibutuhkan pengujian yang disebut dengan *compatibility testing*. Pengujian jenis ini juga bertujuan untuk memastikan apakah *script* yang digunakan untuk mengembangkan suatu sistem informasi dapat bekerja dengan baik pada konfigurasi lingkungan yang berbeda (Pressman, 2010). *Browser* yang berbeda biasanya menghasilkan hasil yang sedikit berbeda, terlepas dari tingkat standarisasi HTML dalam WebApp.

2.7.3 User Acceptance Testing

User Acceptance Testing (UAT) adalah pengujian yang dilakukan oleh pengembang untuk mengetahui sejauh mana sistem dapat memenuhi kriteria kebutuhan pengguna, sehingga sistem yang kita buat benar – benar akan digunakan (Hambling, 2013). *UAT* akan memverifikasi apakah solusi yang dibuat sudah sesuai dengan yang diharapkan oleh pengguna. *UAT* pada pengembangan berorientasi obyek akan berpedoman dengan *Use Case* yang sudah dibuat. Terdapat lima jenis *UAT* yang digunakan pada saat pengujian (Panduan *User Acceptance Testing*, 2017):

1. *Black-box Testing*
2. *Regulation Acceptance Testing*
3. *Alpha & Beta Testing*
4. *Contract Acceptance Testing*
5. *Operational Acceptance Testing*

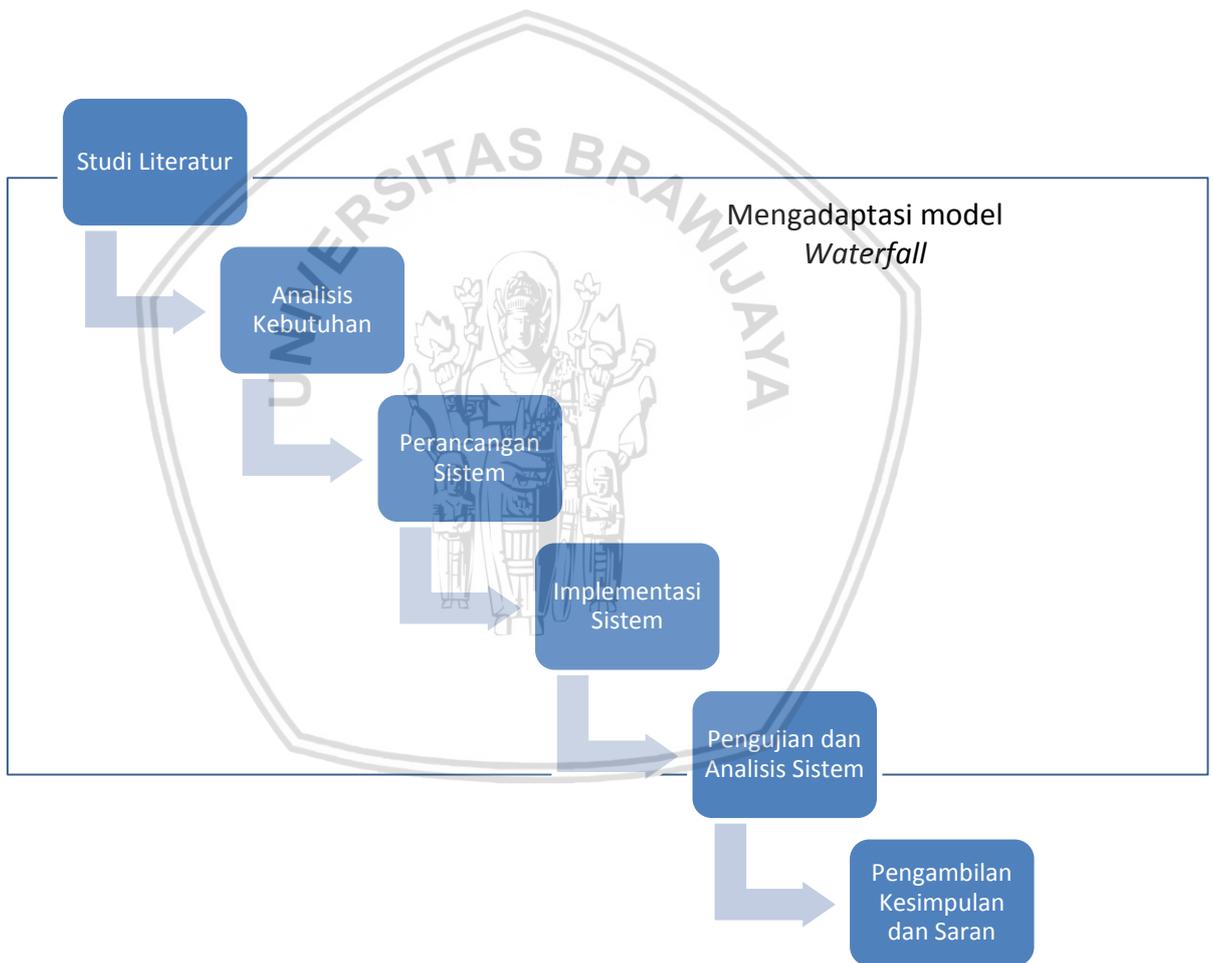
Pada penelitian ini akan menggunakan *UAT* jenis *black-box testing*. Karena jenis ini merupakan jenis pengujian yang berfokus pada kebutuhan fungsional sistem. Jenis *black-box testing* akan menganalisis fungsi tanpa harus memperhatikan struktur kode program. Karena yang terpenting dalam pengujian ini adalah sistem dapat memenuhi kebutuhan pengguna dan pengguna merasa terselesaikan masalahnya dengan adanya fungsi pada sistem ini.

2.7.3.1 Skala Linkert

Untuk menghubungkan sikap dan persepsi responden yang diperiksa dan dianalisis, maka salah satu cara yang dapat digunakan untuk menganalisis permasalahan tersebut yaitu dengan Skala *Likert* (McLeod, et al., 2011). Teknik ini dilakukan dengan cara meminta responden untuk mengisi kuisioner dengan penentuan skala yang sesuai dengan kriteria jawaban. Penilaian dari responden adalah sejauh mana mereka setuju dengan pertanyaan yang diajukan. Cara perhitungan *Skala Likert* dengan memberikan setiap jawaban bobot yang berbeda.

BAB 3 METODOLOGI PENELITIAN

Pada bab ini akan menjelaskan bagaimana tahapan dalam pengembangan sistem informasi Lembaga Bimbingan Belajar Al-Hasyimi. Terdapat enam tahapan, yaitu studi literatur, analisis kebutuhan, perancangan sistem, implementasi sistem, pengujian dan analisis sistem, serta pengambilan kesimpulan dan saran. Berikut adalah gambar diagram alir metodologi yang akan diterapkan.



Gambar 3.1 Alur Metodologi Penelitian



3.1 Studi Literatur

Tahap ini bertujuan untuk mempelajari serta mencari referensi lebih dalam untuk mendukung pengerjaan penelitian ini. Referensi yang diambil diantaranya melalui jurnal baik lokal maupun internasional, buku baik cetak maupun *softfile*, skripsi yang dilakukan sebelumnya, serta artikel dari sumber yang terpercaya. Adapun fokus referensi yang menjadi kebutuhan utama dalam penelitian ini, yaitu perancangan menggunakan pemodelan UML, pengembangan sistem informasi berbasis web, pengembangan berorientasi obyek dengan memanfaatkan design pattern *Model-View-Controller (MVC)*, penggunaan bahasa script *HTML, CSS*, dan bahasa pemrograman *PHP, MySQL* dan *Javascript*, serta pengujian terhadap sistem yang telah dibuat.

3.2 Analisis Kebutuhan

Pada tahap ini dilakukan analisis kebutuhan baik secara fungsional maupun non-fungsional pada sistem yang dikembangkan serta menentukan siapa saja yang menjadi aktor dalam sistem ini. Dalam proses analisis kebutuhan ini dilakukan melalui tiga metode yaitu, wawancara, studi dokumentasi dan observasi. Pada tahap ini seluruh kebutuhan harus benar – benar sudah jelas, sehingga tidak ada lagi perulangan pada tahap pengembangan nanti. Pada tahap ini terjadi pemodelan proses bisnis as-is dan to-be, identifikasi *stakeholder*, identifikasi kebutuhan dan fitur, serta pemodelan *Use Case*.

3.3 Perancangan Sistem

Setelah semua kebutuhan diperoleh, tahap selanjutnya adalah perancangan sistem. Perancangan dilakukan dengan membuat gambaran sistem yang akan dikembangkan dengan menggunakan metode *Unified Model Language (UML)*. Pemodelan yang dibuat diantaranya *Use Case diagram, activity diagram, sequence diagram, diagram class*, desain struktur basis data, perancangan *user interface* dan perancangan pengujian.

3.4 Implementasi Sistem

Tahap implementasi mengacu kepada hasil perancangan yang sudah dibuat sebelumnya. Implementasi sistem menggunakan *framework Codeigniter*, dimana *framework* ini menggunakan *design pattern MVC*. Untuk *back-end* pada sistem, menggunakan bahasa pemrograman *PHP* dan *MySQL* untuk mengelola data. Sedangkan untuk *front-end* pada sistem, menggunakan *HTML, CSS, Javascript*, seta dibantu dengan *framework bootstrap*.

3.5 Pengujian dan Analisis

Pada tahap ini sistem yang sudah jadi akan diuji dengan menggunakan pengujian *black-box*. Pengujian ini berfokus pada pengujian kebutuhan sistem. Apakah sudah sesuai dengan apa yang direncanakan diawal. Dengan adanya

pengujian ini, pengembang akan tahu fungsi – fungsi manakah yang tidak valid, baik dari segi *interface*, struktur data, maupun *performa*.

Kemudian untuk mengetahui apakah sistem yang dikembangkan ini dapat diterima oleh pengguna akhir (*user*) maka perlu dilakukan pengujian *User Acceptance Testing (UAT)*. Pengujian ini bertujuan untuk memvalidasi bahwa solusi yang dibuat dalam sistem telah memenuhi kebutuhan user dan sesuai dengan harapan user. Selain itu juga dilakukan pengujian komabilitas dan validasi.

3.6 Pengambilan Kesimpulan dan Saran

Setelah seluruh tahapan selesai dikerjakan, maka tahap terakhir yaitu pengambilan kesimpulan dan saran. Kesimpulan berisi hasil pengujian dan analisis sistem yang dikembangkan. Sedangkan saran ditujukan kepada pembaca terkait apa saja yang masih kurang dari penelitian ini untuk nantinya dapat dikembangkan lebih lanjut oleh pembaca.



BAB 4 ANALISIS KEBUTUHAN

Pada bab ini akan dijelaskan tentang analisis kebutuhan Sistem Informasi Lembaga Bimbingan Belajar Al-Hasyimi. Analisis kebutuhan terdiri dari identifikasi pemangku kepentingan, pemodelan proses bisnis saat ini dan pemodelan proses bisnis yang diusulkan oleh penulis. Dilanjutkan dengan penjelasan pemangku kepentingan yang terlibat yang kemudian akan dilakukan analisis aktor yang terlibat pada sistem. Setelah itu akan dilakukan analisis kebutuhan baik yang fungsional maupun non-fungsional. Terakhir akan dilakukan pemodelan *Use Case diagram*, *Use Case scenario*, serta *activity diagram*.

4.1 Identifikasi Pemangku Kepentingan

Untuk mengetahui karakteristik dan hubungan tiap pemangku kepentingan dengan sistem maka diperlukan identifikasi pemangku kepentingan. Informasi terkait pemangku kepentingan yang ada di LBB Al-Hasyimi ini diperoleh dari hasil analisis pada hasil wawancara yang telah dilakukan oleh penulis kepada beberapa pihak di LBB Al-Hasyimi.

Penjelasan terkait dengan identifikasi pemangku kepentingan akan dijelaskan pada Tabel 4.1. Hasil identifikasi ini telah dikelompokkan sesuai dengan karakteristik dan hubungan pemangku kepentingan dengan sistem.

Table 4.1 Identifikasi Pemangku Kepentingan

Name	Brief Description	Stakeholder Representative
Pengguna	Individu yang berinteraksi langsung dengan sistem, serta berperan sebagai aktor.	<ul style="list-style-type: none"> • Sekretaris • Tentor • Orang Tua • Siswa
Pengembang	Individu atau organisasi yang mengembangkan sistem informasi dalam penelitian ini.	Penulis
Pihak yang berwenang	Individu atau organisasi yang memiliki wewenang sekaligus mendukung adanya penelitian ini dengan memberikan informasi yang dibutuhkan oleh penulis.	LBB Al-Hasyimi

4.2 Analisis Proses Bisnis

Pada tahap ini akan dilakukan analisis bagaimana kondisi proses bisnis saat ini pada pengelolaan tentor Lembaga Bimbingan Belajar Al-Hasyimi. Hasil dari analisis

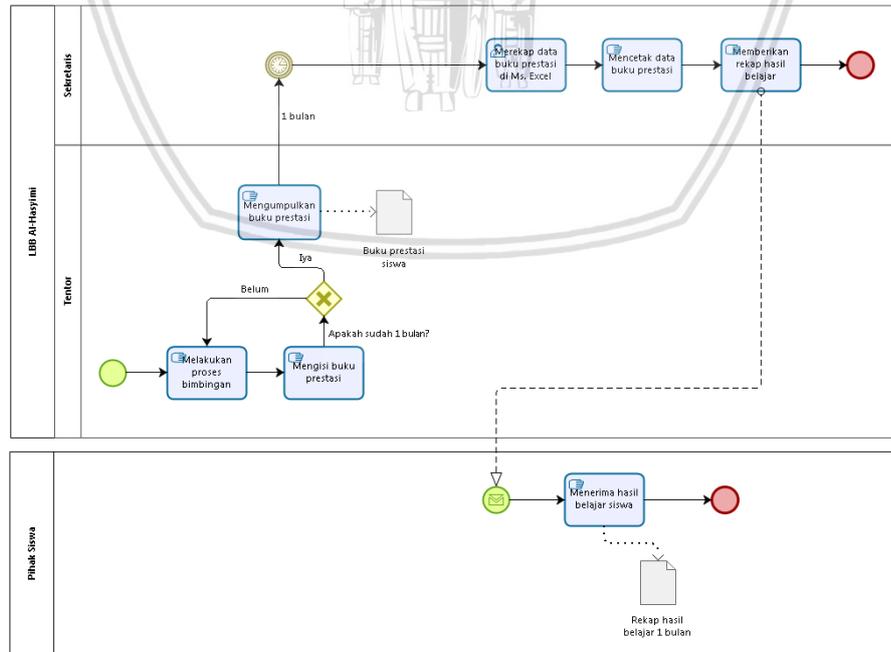
ini akan dimodelkan dalam *Business Process Management and Notation (BPMN)*. Selanjutnya akan dianalisis lagi permasalahan pada proses bisnis saat ini serta dampaknya bagi LBB Al-Hasyimi. Terakhir akan dibuatkan proses bisnis usulan sebagai solusi atas permasalahan tersebut.

4.2.1 Identifikasi Proses Bisnis As-Is

Pada bab ini akan dijelaskan tentang identifikasi proses bisnis as-is yaitu proses bisnis yang sedang berjalan saat ini pada LBB Al-Hasyimi ini. Identifikasi proses bisnis saat ini didapat berdasarkan hasil dari wawancara beberapa pihak LBB Al-Hasyimi, diantaranya ketua, sekretaris, dan salah satu tentor. Dengan hasil wawancara tersebut selain dapat mengetahui bagaimana berjalannya proses bisnis saat ini, dapat diketahui juga beberapa permasalahan yang terjadi pada proses bisnis saat ini. Selanjutnya proses bisnis saat ini akan dibagi menjadi tiga bagian proses bisnis yang akan dijelaskan pada sub-bab berikutnya.

4.2.1.1 Proses Bisnis As-Is Mengelola Hasil Belajar Siswa

Pada Gambar 4.1 menjelaskan proses bisnis as-is tentang pengelolaan data hasil belajar siswa. Untuk saat ini perekapan hasil belajar siswa masih dilakukan dengan cara manual menggunakan buku prestasi. Setiap tentor wajib membawa buku prestasi siswa setiap kali melakukan bimbingan belajar. Setelah selesai belajar. Tmentor mengisi buku prestasi tersebut. Kemudian buku prestasi tersebut diserahkan ke sekretaris setiap satu bulan sekali untuk direkap. Sekretaris memindah data tersebut ke Microsoft Excel untuk direkap dan selanjutnya dicetak untuk diberikan kepada pihak siswa. Pihak siswa merupakan generalisasi dari siswa dan orang tua.

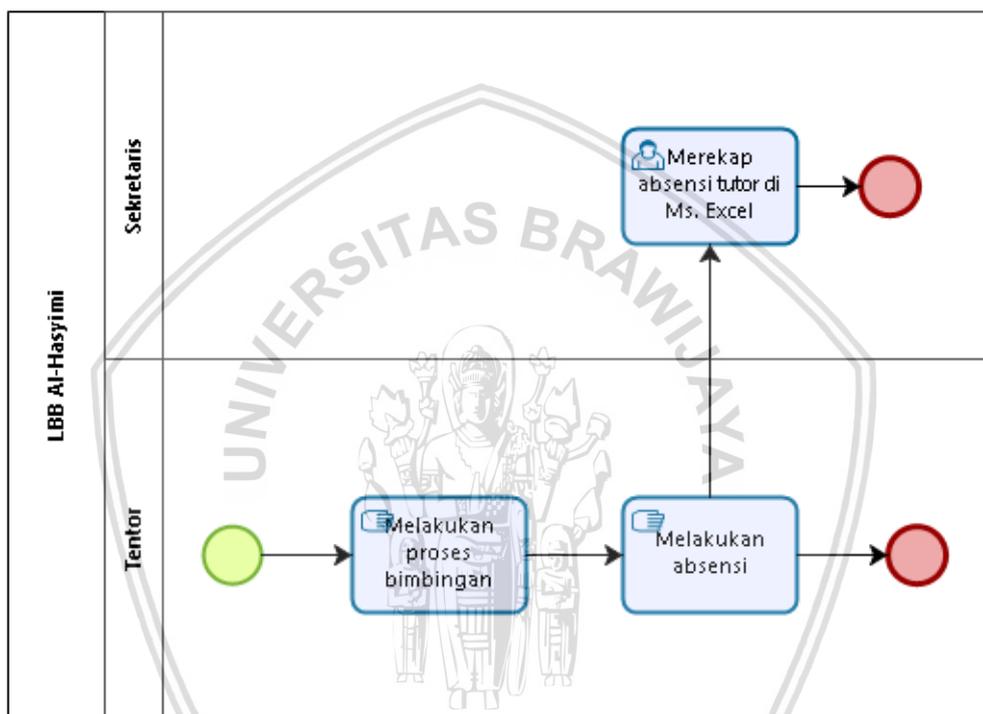


Gambar 4.1 Bisnis Proses As-Is Mengelola Hasil Belajar Siswa



4.2.1.2 Proses Bisnis As-Is Presensi Tentor

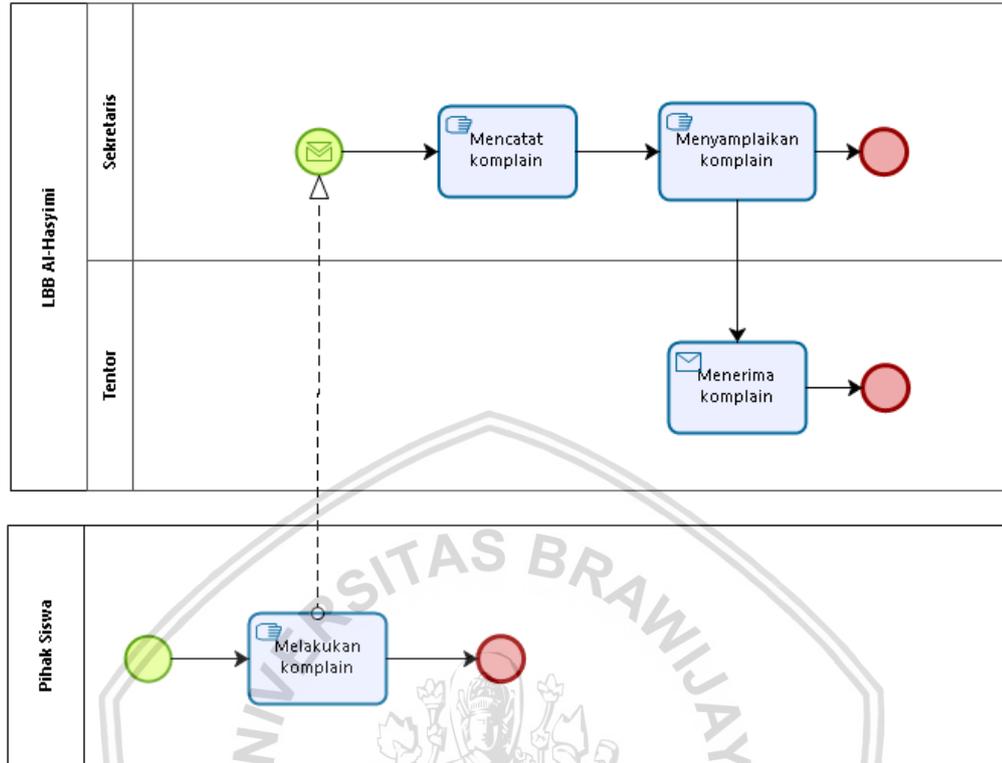
Pada Gambar 4.2 menjelaskan tentang proses bisnis *as-is* presensi tentor. Jadi setelah melakukan bimbingan belajar, selain harus mengisi buku prestasi tersebut, tentor juga harus melakukan presensi dengan cara datang ke Kantor LBB Al-Hasyimi. Presensi dilakukan dengan menggunakan kertas yang sudah tercetak dan terdapat kolom nama tiap tentor serta tanggal pelaksanaan bimbingan tersebut. Setelah data tersebut sudah penuh, kemudian data presensi tersebut akan direkap oleh sekretaris pada Microsoft Excel. Kegiatan presensi ini digunakan untuk melakukan monitoring keaktifan tentor.



Gambar 4.2 Bisnis Proses *As-Is* Presensi Tentor

4.2.1.3 Proses Bisnis *As-Is* Melakukan Komplain

Pada Gambar 4.3 menjelaskan tentang proses bisnis *as-is* bagaimana pihak siswa melakukan komplain. Pihak siswa merupakan generalisasi dari orang tua dan siswa. Hal ini dilakukan karena keduanya melakukan aktifitas yang sama. Jadi ketika pihak siswa memiliki keluhan mengenai kinerja tentor saat proses bimbingan yang kurang baik, pihak siswa melakukan komplain ke pihak LBB Al-Hasyimi dengan cara langsung mendatangi kantor LBB Al-Hasyimi. Kemudian pihak siswa menyampaikan komplain kepada sekretaris. Sekretaris menerima dan mencatat komplain yang disampaikan tadi. Kemudian menyampaikannya kepada tentor yang terkait.



Gambar 4.3 Bisnis Proses As-Is Komplain Orang Tua Siswa

4.2.2 Analisis Permasalahan

Pada bab ini akan menjelaskan permasalahan yang terjadi pada proses bisnis saat ini. Setelah dilakukan analisis terhadap permasalahan yang ada, kemudian akan dianalisis dampak yang muncul dari permasalahan tersebut, terakhir akan dicarikan solusi dari permasalahan yang ada. Pada Tabel 4.2 menjelaskan tentang analisis permasalahan proses bisnis saat ini.

Table 4.2 Analisis Permasalahan

<p>The problem of</p>	<ol style="list-style-type: none"> 1. Pencatatan hasil belajar siswa dilakukan dengan menggunakan buku prestasi yang harus dibawa oleh tentor setiap melakukan bimbingan. 2. Tmentor melaporkan hasil dari buku prestasi sebulan sekali ke sekretaris. 3. Mengumpulkan buku prestasi harus di serahkan ke kantor. 4. Penyimpanan data masih dalam bentuk file excel. 5. Hasil rekap hasil buku prestasi diberikan ke siswa dan orang tua satu bulan sekali dalam bentuk cetakan.
------------------------------	---



Table 4.2 Analisis Permasalahan (lanjutan)

	<ol style="list-style-type: none"> 6. Setelah mengajar tentor harus melakukan presensi, terkadang masih terjadi kelupaan atau ketidak jujuran apalagi tentor harus ke kantor. 7. Jika pihak siswa ada komplain, maka harus langsung menuju kantor. 8. Terkadang tidak ada umpan balik yang diberikan pihak LBB Al-Hasyimi kepada pihak siswa.
Affect	Pelayanan kepada pihak siswa dan efektifitas kerja sekretaris
The Impact of which	<ol style="list-style-type: none"> 1. Pencatatan dalam bentuk kertas (buku prestasi) mudah rusak bahkan hilang. 2. Penumpukan data dapat mempersulit sekretaris dalam penginputan data ke excel dan informasi yang diterima siswa dan orang tua tidak <i>real time</i>. 3. Tantor membutuhkan waktu untuk pengumpulan berkas ke kantor. 4. Data dalam bentuk excel hanya dapat dibuka menggunakan satu perangkat. 5. Tidak dapat memperoleh informasi dengan <i>real time</i>. 6. Ketidak cocokan data dengan kenyataan sebenarnya. 7. Membutuhkan waktu dan biaya untuk melakukan komplain. 8. Kurangnya pelayanan yang baik.
A Successful solution would	<ol style="list-style-type: none"> 1. Menyediakan sistem pelaporan hasil belajar belajar siswa secara <i>real time</i>. 2. Menyediakan sistem presensi secara otomatis saat tentor menambahkan hasil belajar siswa. 3. Menyediakan sistem penyampaian komplain dengan mudah.

Berdasarkan pada hasil analisis masalah pada tabel 4.2, kemudian akan dilakukan pemetakan beberapa proses dalam *BPMN as-is*. Dari hasil pemetakan tersebut maka akan direkomendasikan analisis rekomendasi proses baru untuk menyelesaikan masalah tersebut. Berikut Tabel 4.3 hasil pemetakan dan analisis dari rekomendasi proses baru. Pada tabel tersebut terdapat 7 proses bisnis *as-is* yang akan diperbarui dan ditambah pada proses bisnis *to-be*.

Table 4.3 Hasil Analisis As-Is

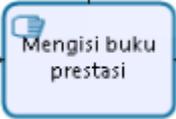
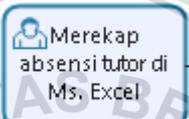
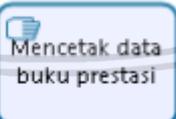
No	Masalah	Proses As-Is	Rekomendasi To-be
1	Pengisian buku prestasi masih menggunakan kertas/buku		Pengisian langsung dilakukan pada sistem dengan menyediakan <i>form</i> buku prestasi pada sistem
2	Tentor harus melakukan presensi setelah mengajar ke Kantor		Presensi langsung otomatis tersimpan di sistem ketika mengisi buku prestasi pada sistem
3	Data presensi direkap dan disimpan masih dalam bentuk Excel.		Data disimpan dan dapat dikelola langsung pada sistem
4	Tentor harus mengumpulkan buku prestasi jika sudah 1 bulan.	 <p>1 bulan</p>	Hasil bimbingan otomatis langsung dikirim dan disimpan oleh sistem setelah tentor mengimputkan data hasil belajar saat itu juga.
5	Hasil evaluasi belajar selama 1 bulan dari buku prestasi tersebut direkap dan dicetak untuk diserahkan ke siswa dan orang tua.		Rekap hasil belajar langsung ditangani oleh sistem dan pengguna bisa langsung melihat secara <i>real time</i> .
6	Jika ada komplain dari pihak siswa, ia harus menuju kantor terlebih dahulu untuk menyampaikan komplain terkait pengajaran tentor.		Menyediakan fitur komplain yang diintegrasikan langsung kepada akun tentor dan sekretaris.



Table 4.3 Hasil Analisis As-Is (lanjutan)

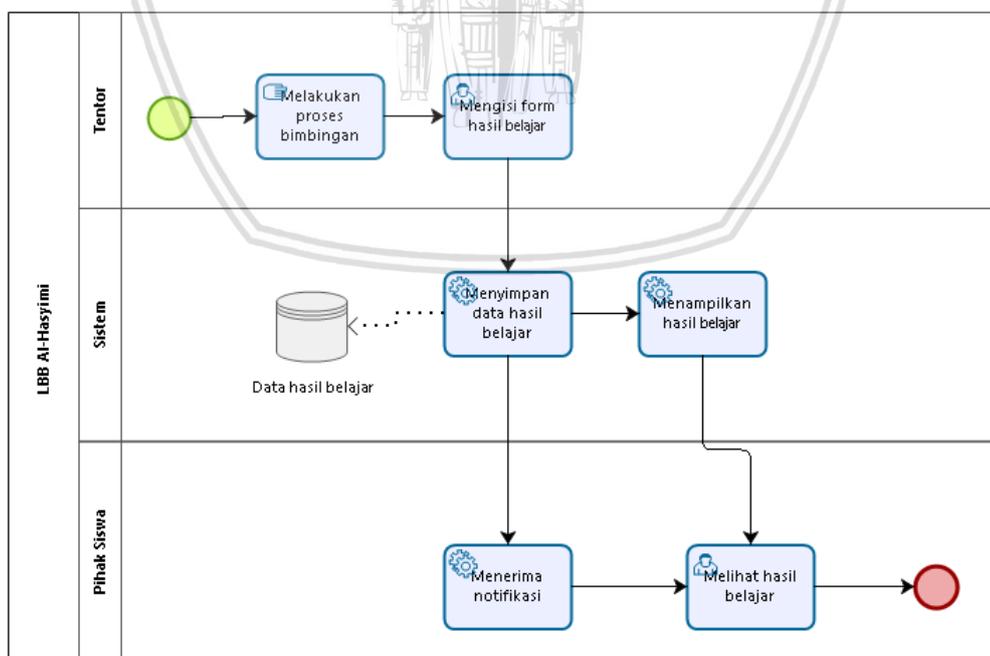
No	Masalah	Proses As-Is	Rekomendasi To-be
7	Komplain dari pihak siswa dicatat pada buku catatan.		Data komplain langsung ditangani oleh sistem.

4.2.3 Pemodelan Proses Bisnis To-be

Berdasarkan hasil analisis pada proses bisnis as-is di atas. Maka akan dibuat pemodelan proses bisnis to-be. Dimana proses bisnis to-be ini akan dapat mengatasi beberapa permasalahan yang masih terdapat pada proses bisnis saat ini. Pada pemodelan proses bisnis to-be ini akan dilakukan penambahan, perubahan bahkan penghilangan beberapa proses pada proses bisnis yang sebelumnya. Berikut adalah gambar sekaligus penjelasan proses bisnis to-be.

4.2.3.1 Proses Bisnis To-be Mengelola Hasil Belajar Siswa

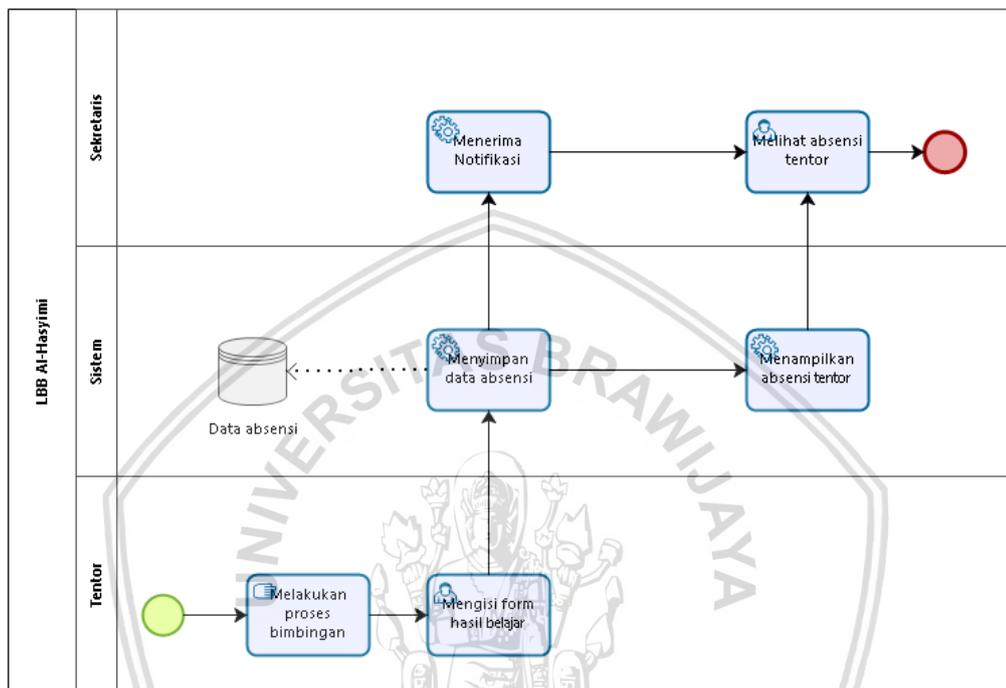
Pada Gambar 4.4 merupakan pemodelan proses bisnis to-be pengolahan hasil belajar siswa. Dimana dengan bantuan sistem informasi, tentor tidak perlu lagi untuk mencatat hasil belajar siswa pada buku prestasi. Tendor dapat menambahkan langsung data hasil belajar siswa pada sistem, kemudian data tersebut akan disimpan oleh sistem dan dapat dilihat secara langsung oleh pihak siswa.



Gambar 4.4 Proses Bisnis To-Be Mengelola Data Hasil Belajar

4.2.3.2 Proses Bisnis *To-be* Presensi Tentor

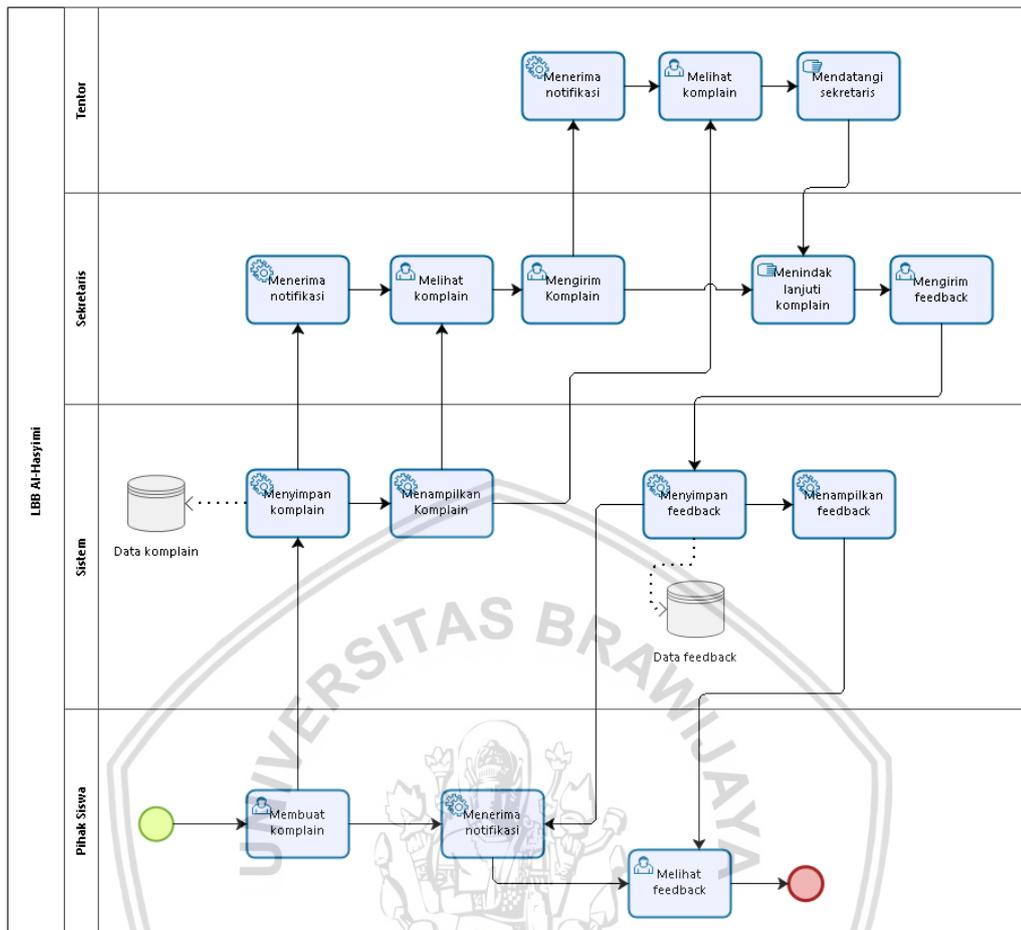
Pada Gambar 4.5 merupakan pemodelan proses bisnis to-be presensi tentor. Dari yang sebelumnya tentor harus melakukan presensi dengan manual, dengan bantuan sistem informasi, tentor akan otomatis terabsen ketika menambahkan data hasil belajar siswa. Kemudian sekretaris akan dapat melihat langsung keaktifan tiap tentor.



Gambar 4.5 Proses Bisnis *To-Be* Presensi Tentor

4.2.3.3 Proses Bisnis *To-be* Melakukan Komplain

Pada Gambar 4.6 merupakan pemodelan proses bisnis to-be melakukan komplain. Dimana pada proses bisnis as-is orang tua harus menuju ke kantor untuk melakukan komplain. Sedangkan pada proses bisnis to-be ini dengan bantuan sistem informasi, pihak siswa dapat langsung mengirimkan komplain melalui sistem. Dimana komplain tersebut akan diterima oleh sekretaris. Dan sekretaris akan mengirimkannya ke tentor yang terkait. Kemudian tentor akan mendatangi sekretaris untuk menindak lanjuti komplain tersebut. Setelah selesai, maka sekretaris akan mengirimkan umpan balik kepada pihak siswa terkait permasalahan tersebut sudah ditindak lanjuti.



Gambar 4.6 Proses Bisnis To-Be Melakukan Komplain

Setelah dilakukan pemodelan proses bisnis to-be selanjutnya akan dilakukan analisis terhadap proses bisnis to-be tersebut dimana proses bisnis to-be ini merupakan jawaban terkait dengan permasalahan yang terjadi pada proses bisnis as-is. Pada Tabel 4.4 merupakan penjelasan hasil analisis to-be.

Table 4.4 Hasil Analisis To-be

No	Masalah	Rekomendasi To-be	Proses To-be
1	Pengisian buku prestasi masih menggunakan kertas/buku	Pengisian langsung dilakukan pada sistem dengan menyediakan <i>form</i> buku prestasi pada sistem	Mengisi form hasil belajar
2	Tentor harus melakukan presensi setelah mengajar ke Kantor	Presensi langsung otomatis tersimpan di sistem ketika mengisi buku prestasi pada sistem	Menyimpan data absensi



Table 4.4 Hasil Analisis To-be (lanjutan)

No	Masalah	Rekomendasi <i>To-be</i>	Proses <i>To-be</i>
3	Sekretaris mengabsen keaktifan tentor masih menggunakan kertas	Presensi langsung otomatis tersimpan di sistem ketika mengisi buku prestasi pada sistem, sehingga sekretaris dapat melihat dan merekap keaktifan tentor melalui sistem tersebut.	
4	Tentor harus mengumpulkan buku prestasi jika sudah 1 bulan	Hasil bimbingan otomatis langsung dikirim dan disimpan oleh sistem setelah tentor menginputkan data hasil belajar saat itu juga.	
5	Data presensi disimpan masih dalam bentuk Excel	Data disimpan dan dapat dikelola langsung pada sistem	
6	Hasil evaluasi belajar selama 1 bulan dari buku prestasi tersebut direkap dan dicetak untuk diserahkan ke siswa dan orang tua.	Rekap hasil belajar langsung ditangani oleh sistem dan pengguna bisa langsung melihat secara <i>real time</i> .	
7	Jika ada komplain dari orang tua, ia harus menuju kantor terlebih dahulu untuk menyampaikan komplain terkait pengajaran tentor, untuk selanjutnya	Menyediakan fitur komplain yang diintegrasikan langsung kepada akun tentor dan sekretaris.	
8	Tidak ada pemberitahuan kalau komplain sudah ditindaklanjuti	Memberi konfirmasi jika komplain sudah diterima oleh tentor	



4.3 Analisis Perbaikan Proses Bisnis

Pada bab ini akan menjelaskan analisis perbaikan proses bisnis untuk mengetahui sejauh mana proses bisnis to-be dapat memberi dampak positif terhadap proses bisnis as-is. Dimana perbandingan ini berdasarkan waktu kerja yang dibutuhkan untuk proses bisnis as-is dan proses bisnis to-be.

Table 4.5 Analisis Perbandingan Proses Bisnis

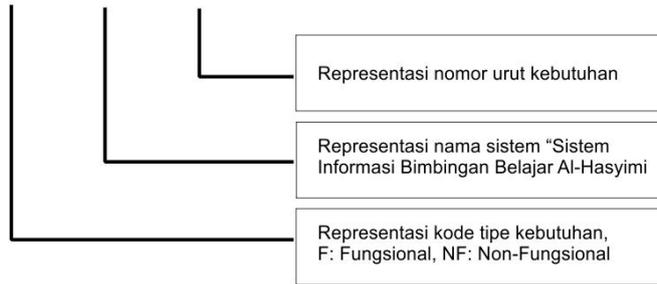
No	Proses bisnis as-is	Proses bisnis to-be	Waktu lama	Waktu baru
1	Tentor mengisi hasil belajar pada buku prestasi	Tentor mengisi hasil belajar pada sistem	5 Menit	3 Menit
2	Tentor menyerahkan buku prestasi kepada sekretaris	Sistem mengirimkan hasil belajar siswa secara otomatis	1 Bulan	3 Detik
3	Tentor melakukan presensi ke sekretaris setelah mengajar	Sistem mengabsen secara otomatis ketika tentor mengirimkan data hasil belajar	30 menit	3 Detik
4	Sekretaris merekap data hasil belajar dan presensi tentor pada excel	Sistem otomatis menyimpan data hasil siswa dan data aktivitas tentor	45 Menit	3 Detik
5	Orang tua melakukan komplain kepada pengurus LBB	Orang tua dapat melakukan komplain langsung melalui sistem	30 Menit	5 Menit

Pada tabel 4.5 merupakan hasil analisis perbandingan proses bisnis setelah dilakukan perbaikan. Data waktu yang dibutuhkan untuk menyelesaikan proses bisnis to-be didapat berdasarkan wawancara dan observasi langsung kepada pihak LBB Al-Hasyimi. Kemudian setelah dilakukan perbaikan proses bisnis, maka didapatkan estimasi waktu yang dibutuhkan untuk menyelesaikan proses bisnis usulan. Dan hasilnya dengan bantuan sistem informasi dari yang sebelumnya membutuhkan waktu yang cukup lama, maka dapat dipersingkat.

4.4 Analisis Kebutuhan *Stakeholder* dan Pengguna

Pada tahap ini akan dijelaskan kebutuhan apa saja yang muncul dari penjabaran permasalahan dan solusi di atas. Terdapat dua tipe kebutuhan, yaitu kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional ini berisi tentang proses – proses apa saja atau layanan apa saja yang terdapat pada sistem. Sedangkan kebutuhan non-fungsional lebih menitik beratkan pada kemudahan pengguna dalam menggunakan sistem. Dalam penulisan kode kebutuhan, akan dilakukan seperti pada contoh Gambar 4.7 berikut:

X-SIBBA-XX



Gambar 4.7 Contoh pembuatan kode kebutuhan

Berikut adalah analisis kebutuhan yang diperlukan oleh *stakeholder* yang dijelaskan pada Tabel 4.6.

Table 4.6 Hasil Analisis Kebutuhan

Kode Kebutuhan	Keterangan	Stakeholder
F-SIBBA-01	Sistem dapat melakukan pengolahan pengguna (<i>add user, edit user, delete user</i>).	Sekretaris
F-SIBBA-02	Sistem harus dapat menyediakan fasilitas untuk penginputan data hasil belajar dalam bentuk <i>form</i> .	Tentor
F-SIBBA-03	Sistem harus dapat menampilkan informasi hasil belajar siswa yang dapat dilihat oleh siswa dan orang tua itu sendiri serta sekretaris dan tentornya.	Siswa, orang tua
F-SIBBA-04	Sistem harus dapat menampilkan presensi tentor.	Sekretaris
F-SIBBA-05	Sistem harus dapat menyediakan fasilitas komplain yang dapat dilakukan oleh siswa maupun orang tua.	Orang tua, Siswa,
F-SIBBA-06	Sistem dapat mengelola komplain yang masuk.	Sekretaris
F-SIBBA-07	Sistem dapat mengirimkan <i>feedback</i> dari komplain yang diajukan.	Sekretaris, Siswa, Orang Tua
F-SIBBA-08	Sistem harus dapat mengelola pengumuman.	Sekretaris, Tentor, Orang tua, Siswa



Table 4.6 Hasil Analisis Kebutuhan (lanjutan)

Kode Kebutuhan	Keterangan	Stakeholder
F-SIBBA-09	Sistem dapat membuat biodata siswa.	Siswa
F-SIBBA-10	Sistem dapat melakukan ubah password.	Sekretaris, Tentor, Siswa, Orang Tua
NF-SIBBA-11	Sistem dapat membedakan otorisasi pengguna.	Orang tua, Siswa, Sekretaris, Tentor
NF-SIBBA-12	Sistem harus dapat dibuka di berbagai jenis ukuran perangkat (<i>responsive</i>).	Orang tua, Siswa, Sekretaris, Tentor
NF-SIBBA-13	Sistem memiliki tampilan antarmuka yang baik dan mudah dipahami.	Orang tua, Siswa, Sekretaris, Tentor

Pada Tabel 4.7 menjelaskan tentang kode kebutuhan pengguna. Dimana tiap kode kebutuhan akan dijelaskan pada kolom keterangan. Dan tiap kebutuhan memiliki stakeholder yang berbeda. Berdasarkan hasil analisis di atas, diperoleh 7 kebutuhan fungsional dan 3 kebutuhan non-fungsional. Dan selanjutnya akan di buat fitur berdasarkan hasil analisis kebutuhan tersebut.

4.5 Identifikasi Aktor

Pada proses selanjutnya akan menjelaskan aktor – aktor yang terlibat dalam sistem ini. Aktor adalah obyek yang berinteraksi dengan sistem secara langsung. Obyek bisa berupa orang ataupun sistem eksternal yang terlibat dalam sistem. Identifikasi aktor ini berdasarkan pada diagram *Use Case* yang telah dibuat sebelumnya. Berikut pada Tabel 4.8 akan dijelaskan aktor – aktor yang terlibat.

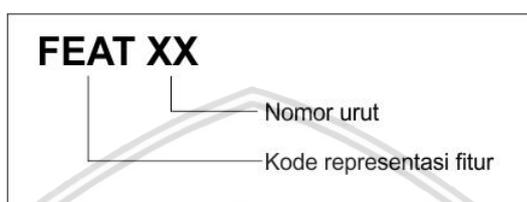
Table 4.8 Identifikasi Aktor

Aktor	Deskripsi
Sekretaris	Pengguna yang bertugas mengumpulkan dan mengelola data tentor dan siswa serta menerima komplain dari siswa ataupun orang tua.
Tentor	Pengguna yang bertugas menginputkan data siswa berupa buku prestasi siswa dan presensi tentor.

Table 4.8 Identifikasi Aktor (lanjutan)

Aktor	Deskripsi
Siswa	Pengguna yang dapat melihat hasil belajar serta memberikan evaluasi belajar kepada tentor.
Orang tua	Pengguna yang melihat hasil belajar anaknya serta memberikan komplain kepada LBB Al-Hasyimi terkait pelayanan yang diberikan.

4.6 Identifikasi Fitur



Gambar 4.8 Kodifikasi Fitur

Pada Tabel 4.8 merupakan penjelasan tentang jenis – jenis fitur yang terdapat pada sistem yang diperoleh berdasarkan analisis kebutuhan pengguna. Dengan adanya fitur ini merupakan representasi dari solusi yang ditawarkan untuk mengatasi permasalahan yang ada. Hasil analisis ini akan digunakan untuk melakukan spesifikasi kebutuhan sistem. Terdapat 13 fitur yang akan diuraikan pada Tabel 4.9 berikut.

Table 4.9 Identifikasi Fitur

Kode Fungsi	Nama Fitur	Keterangan	Prioritas
FEAT01	Mengelola data pengguna	Sistem harus dapat membuat, membaca, mengedit, dan menghapus pengguna, serta dapat menampilkan list pengguna.	Mo
FEAT02	Menambahkan hasil belajar	Sistem harus dapat menambahkan data yang ada pada buku prestasi atau hasil belajar siswa untuk disimpan dan kelola lebih lanjut oleh sistem.	Mo



Kode Fungsi	Nama Fitur	Keterangan	Prioritas
FEAT03	Melihat hasil belajar siswa	Sistem harus dapat menampilkan hasil belajar siswa yang dapat dilihat oleh sekretaris, tentor yang mengajar, orang tua dan siswa itu sendiri	Mo
FEAT04	Melihat presensi	Sistem dapat menampilkan presensi tentor	Mo
FEAT05	Mengirim komplain	Sistem menyediakan fitur komplain terhadap pengajaran tentor yang dapat digunakan oleh siswa dan orang tua	Mo
FEAT06	Mengelola komplain	Sistem harus dapat mengelola komplain yang masuk	Mo
FEAT07	Mengirim <i>feedback</i>	Sistem harus dapat menampilkan <i>feedback</i>	Mo
FEAT08	Mengelola pengumuman	Sistem harus dapat mengelola pengumuman	Co
FEAT09	Membuat biodata	Sistem harus dapat membuat biodata siswa	S
FEAT10	Mengubah password	Sistem harus dapat mengubah password sesuai keinginan pengguna	Co
FEAT11	Login	Fitur login untuk melakukan autentifikasi pengguna	Mo
FEAT12	Logout	Fitur logout untuk keluar dari akun	Mo

Table 4.9 Identifikasi Fitur (lanjutan)

Kode Fungsi	Nama Fitur	Keterangan	Prioritas
FEAT13	Responsive	Sistem dapat dibuka di berbagai ukuran perangkat/browser dengan tata letak atau tampilan menyesuaikan ukuran perangkat/browser	S

4.7 Spesifikasi Kebutuhan

Pada Tabel 4.10 ini akan dijelaskan spesifikasi kebutuhan dari sistem yang diperoleh berdasarkan analisis fitur sebelumnya. Pada spesifikasi kebutuhan ini terdapat 17 kebutuhan fungsional sistem dan 1 kebutuhan non-fungsional. Kode fitur diperoleh dari analisis fitur pada Tabel 4.9 dan kode kebutuhan diperoleh dari analisis kebutuhan pada Tabel 4.10 untuk kemudian didekomposisikan lagi menjadi kode fitur lengkap.

Table 4.40 Spesifikasi Kebutuhan

No	Kode Fitur	Kode Kebutuhan	Kode Fitur Lengkap	Keterangan
1	FEAT01	F-SIBBA-01	F-SIBBA-01-01	Sistem dapat menambahkan pengguna
			F-SIBBA-01-02	Sistem dapat mengedit data pengguna
			F-SIBBA-01-03	Sistem dapat menghapus akun pengguna
			F-SIBBA-01-04	Sistem dapat menampilkan list pengguna
2	FEAT02	F-SIBBA-02	F-SIBBA-02-01	Sistem dapat menambahkan data buku prestasi
3	FEAT03	F-SIBBA-03	F-SIBBA-03-01	Sistem dapat menampilkan data hasil belajar siswa

No	Kode Fitur	Kode Kebutuhan	Kode Fitur Lengkap	Keterangan
			F-SIBBA-03-02	Sistem dapat memberikan pemberitahuan jika hasil belajar sudah dimasukkan.
4	FEAT04	F-SIBBA-04	F-SIBBA-04-01	Sistem dapat menampilkan presensi tentor.
			F-SIBBA-04-02	Sistem dapat memberikan notifikasi jika ada presensi masuk.
5	FEAT05	F-SIBBA-05	F-SIBBA-05-01	Sistem dapat mengirimkan komplain
6	FEAT06	F-SIBBA-06	F-SIBBA-06-01	Sistem dapat mengelola komplain.
7	FEAT07	F-SIBBA-07	F-SIBBA-06-01	Sistem dapat menampilkan <i>feedback</i> .
8	FEAT08	F-SIBBA-08	F-SIBBA-08-01	Sistem dapat membuat pengumuman.
			F-SIBBA-08-02	Menampilkan pengumuman.
9	FEAT09	F-SIBBA-09	F-SIBBA-09-01	Sistem dapat membuat biodata siswa
			F-SIBBA-09-02	Sistem dapat mengedit biodata siswa
10	FEAT10	F-SIBBA-10	F-SIBBA-10-01	Sistem dapat mengubah password pengguna
11	FEAT11	F-SIBBA-11	F-SIBBA-11-01	Sistem dapat melakukan

No	Kode Fitur	Kode Kebutuhan	Kode Fitur Lengkap	Keterangan
				autentifikasi pengguna
12	FEAT12	F-SIBBA-12	F-SIBBA-12-01	Pengguna dapat keluar dari sistem dengan aman
13	FEAT13	NF-SIBBA-13	NF-SIBBA-13-01	Tampilan atau tata letak sistem dapat menyesuaikan ukuran perangkat atau browser

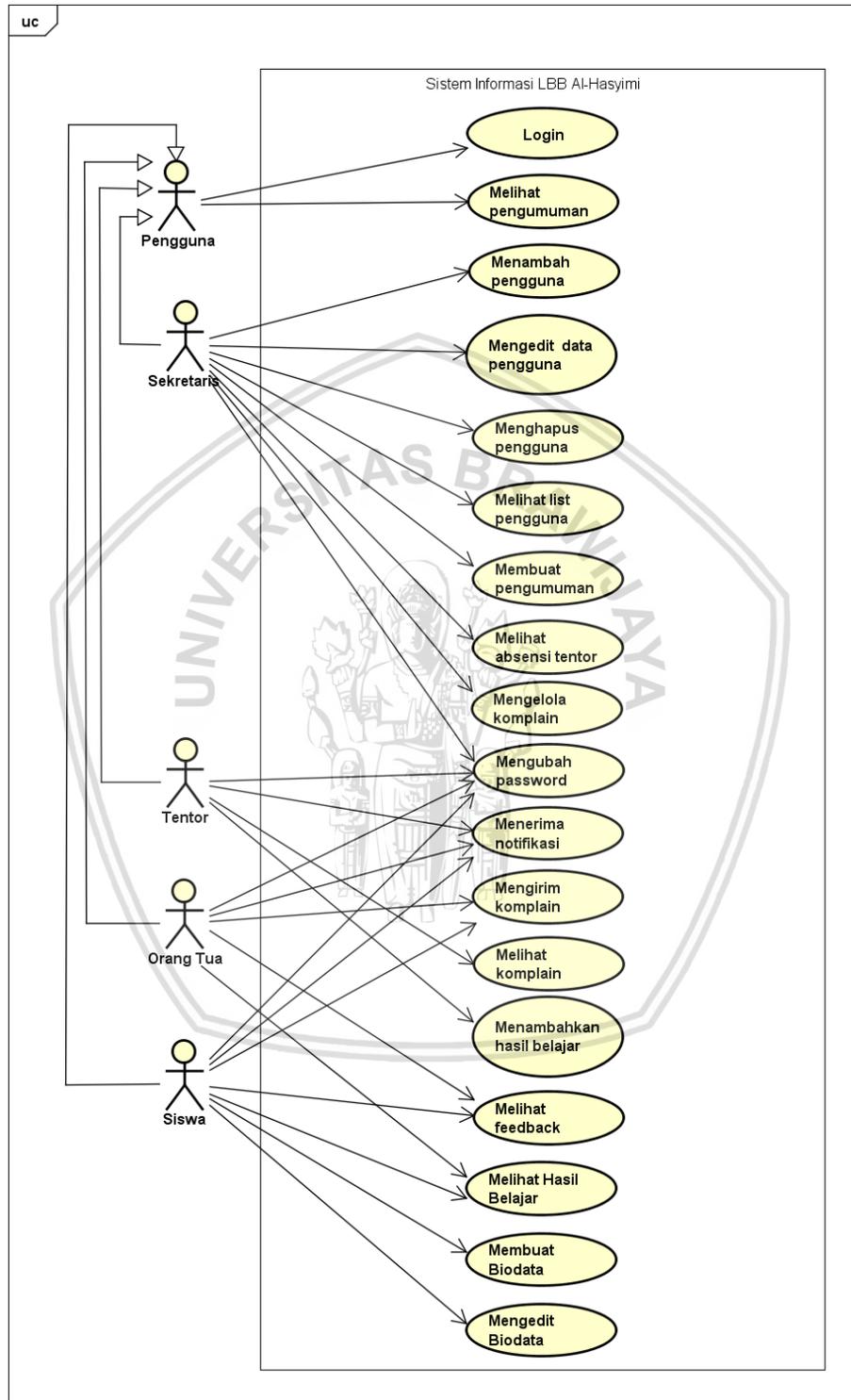
4.8 Pemodelan *Use Case*

Pada bab ini akan menjelaskan tentang pemodelan *Use Case* dengan menggunakan diagram. Sebelum menuju pada diagram *Use Case*, berikut Tabel 4.12 adalah penjelasan jenis *Use Case* yang diperoleh dari proses spesifikasi kebutuhan. Terdapat 16 *Use Case* yang nantinya akan dimodelkan dalam bentuk diagram.

Table 4.5 Kebutuhan Fungsional

Kode <i>Use Case</i>	Kode Fitur Lengkap	<i>Use Case</i>
UC-001	F-SIBBA-01-01	Menambah pengguna
UC-002	F-SIBBA-01-02	Mengedit data pengguna
UC-003	F-SIBBA-01-03	Menghapus pengguna
UC-004	F-SIBBA-01-04	Melihat list pengguna
UC-005	F-SIBBA-02-01	Menambahkan hasil belajar
UC-006	F-SIBBA-03-01	Melihat hasil belajar
UC-007	F-SIBBA-03-02	Menerima notifikasi
UC-008	F-SIBBA-04-01	Melihat presensi tentor
UC-007	F-SIBBA-04-02	Menerima notifikasi
UC-009	F-SIBBA-05-01	Mengirim komplain
UC-010	F-SIBBA-06-01	Mengelola komplain
UC-011	F-SIBBA-07-01	Melihat <i>feedback</i>
UC-012	F-SIBBA-08-01	Membuat pengumuman
UC-013	F-SIBBA-08-02	Melihat pengumuman
UC-014	F-SIBBA-09-01	Membuat biodata
UC-015	F-SIBBA-09-02	Mengedit biodata
UC-016	F-SIBBA-10-01	Mengubah <i>password</i>
UC-017	F-SIBBA-11-01	<i>Login</i>

Pemodelan *Use Case* ini akan menggambarkan hubungan antara aktor dan fungsionalitas sistem. Berikut adalah gambaran diagram *Use Case* pada sistem ini:



Gambar 4.9 Diagram *Use Case*

Pada Gambar 4.9 menunjukkan diagram *Use Case* dari sistem yang dikembangkan. Diagram *Use Case* tersebut dibuat berdasarkan analisis kebutuhan yang dilakukan sebelumnya. Pada diagram *Use Case* tersebut terdapat 4 aktor yang berinteraksi dengan sistem dan terdapat 17 kebutuhan fungsionalitas sistem.

4.9 Use Case Scenario

Pada bagian ini akan dijelaskan lebih detail tentang skenario interaksi aktor dengan sistem atau yang disebut dengan *Use Case scenario*. Pembuatan *Use Case scenario* ini akan ini berdasarkan masing – masing *Use Case diagram*.

4.9.1 Use Case Scenario Menambah Pengguna

Berikut adalah tabel penjelasan *Use Case scenario* menambah pengguna.

Table 4.6 Use Case Scenario Menambah Pengguna

Item	Deskripsi
<i>Use Case Code</i>	UC-001
<i>Use Case Name</i>	Menambah Pengguna
<i>Actor</i>	Sekretaris
<i>Description</i>	Sekretaris dapat menambahkan pengguna baru
<i>Precondition</i>	<i>Login</i> berhasil sebagai sekretaris
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Sekretaris memilih fungsi menu pengguna 2. Sistem menampilkan data pengguna 3. Sekretaris memilih fungsi tambah 4. Sistem menampilkan <i>form</i> 5. Sekretaris mengisi <i>form</i> dengan lengkap dan benar 6. Sekretaris memilih fungsi simpan 7. Sistem menyimpan data dan menambahkan pengguna baru
<i>Alternatif Flow</i>	<ol style="list-style-type: none"> 1. Jika <i>form</i> terdapat yang kosong, maka sistem akan memberikan pesan “Data tidak lengkap”, dan kembali ke langkah 4
<i>Postcondition</i>	Pengguna baru ditambahkan oleh sistem

4.9.2 Use Case Scenario Mengedit Data Pengguna

Berikut adalah tabel penjelasan *Use Case scenario* mengedit data pengguna.

Table 4.7 Use Case Scenario Mengedit Data Pengguna

Item	Deskripsi
<i>Use Case Code</i>	UC-002
<i>Use Case Name</i>	Mengedit data pengguna
<i>Actor</i>	Sekretaris
<i>Brief Description</i>	Sekretaris dapat mengedit data pengguna
<i>Precondition</i>	Login berhasil sebagai sekretaris
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Sekretaris memilih menu pengguna 2. Sistem menampilkan list pengguna 3. Sekretaris memilih fungsi edit 4. Sistem menampilkan <i>form</i> data pengguna untuk diedit 5. Sekretaris mengedit data yang ingin diubah 6. Sekretaris memilih fungsi simpan 7. Sistem menyimpan perubahan data
<i>Alternatif Flow</i>	<ol style="list-style-type: none"> 1. Jika ada data yang kosong maka sistem akan memberi pesan "Data tidak boleh kosong", dan kembali ke langkah 5
<i>Postcondition</i>	Data berhasil diubah oleh sistem

4.9.3 Use Case Scenario Menghapus Pengguna

Berikut adalah tabel penjelasan *Use Case scenario* menghapus pengguna.

Table 4.8 Use Case Scenario Menghapus Pengguna

Item	Deskripsi
<i>Use Case Code</i>	UC-003
<i>Use Case Name</i>	Menghapus pengguna
<i>Actor</i>	Sekretaris
<i>Brief Description</i>	Sekretaris dapat menghapus pengguna yang sudah tidak aktif

Table 4.9 Use Case Scenario Menghapus Pengguna (lanjutan)

Item	Deskripsi
<i>Precondition</i>	Login berhasil sebagai sekretaris
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Sekretaris memilih fungsi menu pengguna 2. Sistem menampilkan list pengguna 3. Sekretaris memilih fungsi hapus 4. Sistem menampilkan peringatan "apakah pengguna ingin dihapus" 5. Sekretaris memilih pilihan "iya" 6. Sistem menghapus pengguna
<i>Alternatif Flow</i>	-
<i>Postcondition</i>	Pengguna dihapus oleh sistem

4.9.4 Use Case Scenario Melihat List Pengguna

Berikut adalah tabel penjelasan *Use Case scenario* melihat pengguna.

Table 4.10 Use Case Scenario Menghapus Pengguna

Item	Deskripsi
<i>Use Case Code</i>	UC-004
<i>Use Case Name</i>	Melihat list pengguna
<i>Actor</i>	Sekretaris
<i>Brief Description</i>	Sekretaris dapat melihat list pengguna
<i>Precondition</i>	Login berhasil sebagai sekretaris
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Sekretaris memilih fungsi menu pengguna 2. Sistem menampilkan list pengguna
<i>Alternatif Flow</i>	-
<i>Postcondition</i>	List pengguna ditampilkan oleh sistem

4.9.5 Use Case Scenario Menambahkan Hasil Belajar

Berikut adalah tabel penjelasan *Use Case scenario* menambahkan hasil belajar.

Table 4.11 Use Case Scenario Menambahkan Data Hasil Belajar

Item	Deskripsi
<i>Use Case Code</i>	UC-005
<i>Use Case Name</i>	Menambahkan data hasil belajar
<i>Actor</i>	Tentor
<i>Brief Description</i>	Tentor menambahkan data hasil belajar siswa setelah melakukan bimbingan belajar
<i>Precondition</i>	Status login berhasil sebagai tentor
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Tentor memilih menu tambahkan hasil belajar 2. Sistem menampilkan <i>form</i> hasil belajar siswa 3. Tentor menambahkan hasil belajar siswa dengan lengkap dan benar 4. Tentor memilih fungsi simpan untuk menyimpan data 5. Sistem memberikan pesan kepada tentor data berhasil disimpan
<i>Alternatif Flow</i>	<ol style="list-style-type: none"> 1. Jika data yang dimasukkan tidak lengkap maka sistem akan memberi tahu kalau "Data harus diisi dengan lengkap" dan kembali ke langkah 1.
<i>Post-Condition</i>	Data disimpan dan dikirim oleh sistem

4.9.6 Use Case Scenario Melihat Hasil Belajar

Berikut adalah tabel penjelasan *Use Case scenario* melihat hasil belajar.

Table 4.12 Use Case Scenario Melihat Hasil Belajar Siswa

Item	Deskripsi
<i>Use Case Code</i>	UC-006
<i>Use Case Name</i>	Melihat Hasil Belajar Siswa
<i>Actor</i>	Siswa, Orang Tua
<i>Brief Description</i>	Siswa dan orang tua dapat melihat hasil belajar siswa
<i>Precondition</i>	Status login berhasil sebagai siswa atau orang tua

Table 4.13 Use Case Scenario Melihat Hasil Belajar Siswa (lanjutan)

Item	Deskripsi
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Siswa atau orang tua memilih menu hasil belajar 2. Sistem menampilkan informasi hasil belajar siswa
<i>Alternatif Flow</i>	-
<i>Postcondition</i>	Data hasil belajar siswa ditampilkan oleh sistem

4.9.7 Use Case Scenario Menerima Notifikasi

Berikut adalah tabel penjelasan *Use Case scenario* menerima notifikasi.

Table 4.14 Use Case Scenario Menerima notifikasi

Item	Deskripsi
<i>Use Case Code</i>	UC-007
<i>Use Case Name</i>	Menerima notifikasi
<i>Actor</i>	Siswa, Orang Tua
<i>Brief Description</i>	Siswa dan orang tua dapat menerima notifikasi
<i>Precondition</i>	Status login berhasil sebagai siswa atau orang tua
<i>Basic Flow</i>	1. Sistem menerima notifikasi
<i>Alternatif Flow</i>	1. Jika tidak ada data baru masuk, maka tidak akan ada notifikasi
<i>Postcondition</i>	Notifikasi berhasil diterima

4.9.8 Use Case Scenario Melihat Presensi Tentor

Berikut adalah tabel penjelasan *Use Case scenario* presensi tentor.

Table 4.15 Use Case Scenario Melihat Presensi Tentor

Item	Deskripsi
<i>Use Case Code</i>	UC-008
<i>Use Case Name</i>	Melihat presensi tentor
<i>Actor</i>	Sekretaris

Table 4.16 Use Case Scenario Melihat Presensi Tentor (lanjutan)

Item	Deskripsi
<i>Brief Description</i>	Sekretaris dapat melihat presensi tentor
<i>Precondition</i>	Pengguna sudah dalam kondisi login
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Sekretaris memilih menu presensi tentor 2. Sistem menampilkan presensi tentor 3. Sekretaris memilih tombol lihat detail 4. Sistem menampilkan detail presensi tentor
<i>Alternatif Flow</i>	-
<i>Postcondition</i>	Presensi tentor berhasil ditampilkan

4.9.9 Use Case Scenario Mengirim Komplain

Berikut adalah tabel penjelasan *Use Case scenario* mengirim komplain.

Table 4.17 Use Case Scenario Mengirim Komplain

Item	Deskripsi
<i>Use Case Code</i>	UC-009
<i>Use Case Name</i>	Mengirim Komplain
<i>Actor</i>	Siswa dan Orang tua
<i>Description</i>	Siswa atau orang tua mengirimkan komplain kepada terhadap kinerja tentor
<i>Precondition</i>	Status login berhasil sebagai siswa atau orang tua
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Siswa atau orang tua memilih menu Berikan komplain 2. Sistem menampilkan <i>form</i> komplain 3. Siswa atau orang tua menambahkan komplain 4. Siswa atau orang tua memilih fungsi kirim untuk mengirim komplain 5. Sistem mengirim komplain ke sekretaris 6. Sistem memberikan pesan kepada siswa atau orang tua komplain berhasil dikirim

Table 4.18 Use Case Scenario Mengirim Komplain (lanjutan)

Item	Deskripsi
<i>Alternatif Flow</i>	1. Jika data yang dimasukkan kosong maka sistem akan memberi tahu kalau “Data tidak boleh kosong” dan kembali ke langkah 3.
<i>Postcondition</i>	Komplain dikirim oleh sistem

4.9.10 Use Case Scenario Mengelola Komplain

Berikut adalah tabel penjelasan *Use Case scenario* mengelola komplain.

Table 4.19 Use Case Scenario Mengelola Komplain

Item	Deskripsi
<i>Use Case Code</i>	UC-010
<i>Use Case Name</i>	Mengelola Komplain
<i>Actor</i>	Sekretaris
<i>Description</i>	Sekretaris mengirimkan pemberitahuan kepada tentor
<i>Precondition</i>	Status login berhasil sebagai sekretaris
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Sekretaris memilih menu komplain 2. Sistem menampilkan list komplain masuk 3. Sekretaris memilih komplain yang ingin ditanggapi 4. Sistem menampilkan <i>form</i> 5. Sekretaris mengisi <i>form</i> dan memilih fungsi kirim 6. Sistem mengirim pesan
<i>Alternatif Flow</i>	-
<i>Postcondition</i>	Pesan dikirim oleh sistem

4.9.11 Use Case Scenario Melihat Feedback

Berikut adalah tabel penjelasan *Use Case scenario* melihat feedback.



Table 4.20 Use Case Scenario Mengelola Komplain

Item	Deskripsi
<i>Use Case Code</i>	UC-011
<i>Use Case Name</i>	Melihat feedback
<i>Actor</i>	Siswa, Orang Tua
<i>Description</i>	Siswa atau orang tua melihat feedback
<i>Precondition</i>	Status login berhasil sebagai siswa atau orang tua
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Siswa atau orang tua memilih menu inbox 2. Sistem menampilkan feedback masuk
<i>Alternatif Flow</i>	-
<i>Postcondition</i>	Feedback berhasil ditampilkan

4.9.12 Use Case Scenario Membuat Pengumuman

Berikut adalah tabel penjelasan *Use Case scenario* membuat pengumuman.

Table 4.21 Use Case Scenario Membuat Pengumuman

Item	Deskripsi
<i>Use Case Code</i>	UC-012
<i>Use Case Name</i>	Membuat Pengumuman
<i>Actor</i>	Sekretaris
<i>Description</i>	Sekretaris menambahkan pengumuman baru
<i>Precondition</i>	Status login berhasil sebagai sekretaris
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Sekretaris memilih menu pengumuman 2. Sistem menampilkan list pengumuman 3. Sekretaris memilih fungsi tambah 4. Sistem menampilkan <i>form</i> pengumuman 5. Sekretaris mengisi <i>form</i> dan memilih fungsi simpan 6. Sistem menyimpan dan menampilkan pengumuman
<i>Alternatif Flow</i>	<ol style="list-style-type: none"> 1. Jika ada form yang tidak diisi, maka kembali ke langkah nomor 5

Table 4.22 Use Case Scenario Membuat Pengumuman (lanjutan)

Item	Deskripsi
<i>Postcondition</i>	Pengumuman berhasil ditambahkan oleh sistem

4.9.13 Use Case Scenario Melihat Pengumuman

Berikut adalah tabel penjelasan *Use Case scenario* melihat pengumuman.

Table 4.23 Use Case Scenario Melihat Pengumuman

Item	Deskripsi
<i>Use Case Code</i>	UC-013
<i>Use Case Name</i>	Melihat pengumuman
<i>Actor</i>	Sekretaris, Tentor, Siswa, Orang Tua
<i>Brief Description</i>	Pengguna dapat melihat pengumuman
<i>Precondition</i>	Pengguna sudah dalam kondisi login
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Sekretaris memilih menu pengumuman 2. Sistem menampilkan daftar pengumuman 3. Sekretaris memilih salah satu judul pengumuman 4. Sistem menampilkan pengumuman
<i>Alternatif Flow</i>	-
<i>Postcondition</i>	Pengumuman berhasil ditampilkan oleh sistem

4.9.14 Use Case Scenario Membuat Biodata

Berikut adalah tabel penjelasan *Use Case scenario* membuat biodata.

Table 4.24 Use Case Scenario Membuat Biodata

Item	Deskripsi
<i>Use Case Code</i>	UC-014
<i>Use Case Name</i>	Membuat biodata
<i>Actor</i>	Siswa
<i>Brief Description</i>	Siswa dapat membuat biodata

Table 4.25 Use Case Scenario Membuat Biodata (lanjutan)

Item	Deskripsi
<i>Precondition</i>	Siswa dalam kondisi login
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Siswa memilih menu biodata 2. Sistem menampilkan <i>form</i> biodata 3. Siswa memasukkan data biodata 4. Siswa memilih fungsi tombol simpan
<i>Alternatif Flow</i>	-
<i>Postcondition</i>	Biodata berhasil disimpan

4.9.15 Use Case Scenario Mengedit Biodata

Berikut adalah tabel penjelasan *Use Case scenario* mengedit biodata.

Table 4.26 Use Case Scenario Mengedit Biodata

Item	Deskripsi
<i>Use Case Code</i>	UC-015
<i>Use Case Name</i>	Membuat biodata
<i>Actor</i>	Siswa
<i>Brief Description</i>	Siswa dapat membuat biodata
<i>Precondition</i>	Siswa dalam kondisi login
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Siswa memilih menu biodata 2. Sistem menampilkan <i>form</i> biodata 3. Siswa memasukkan data biodata 4. Siswa memilih fungsi tombol simpan
<i>Alternatif Flow</i>	-
<i>Postcondition</i>	Biodata berhasil disimpan

4.9.16 Use Case Scenario Mengubah Password

Berikut adalah tabel penjelasan *Use Case scenario* mengubah password



Table 4.27 Use Case Scenario Mengubah Password

Item	Deskripsi
<i>Use Case Code</i>	UC-016
<i>Use Case Name</i>	Mengubah Password
<i>Actor</i>	Sekretaris, Tentor, Orang Tua, Siswa
<i>Brief Description</i>	Pengguna dapat mengubah password
<i>Precondition</i>	Pengguna dalam kondisi login
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Siswa memilih menu My Akun 2. Sistem menampilkan <i>form</i> 3. Siswa memasukkan data 4. Siswa memilih fungsi tombol simpan
<i>Alternatif Flow</i>	<ol style="list-style-type: none"> 1. Jika ada yang tidak diisi atau tidak lengkap, maka kembali ke nomot 3
<i>Postcondition</i>	Password berhasil diubah

4.9.17 Use Case Scenario Login

Berikut adalah tabel penjelasan *Use Case* login.

Table 4.28 Use Case Scenario Login

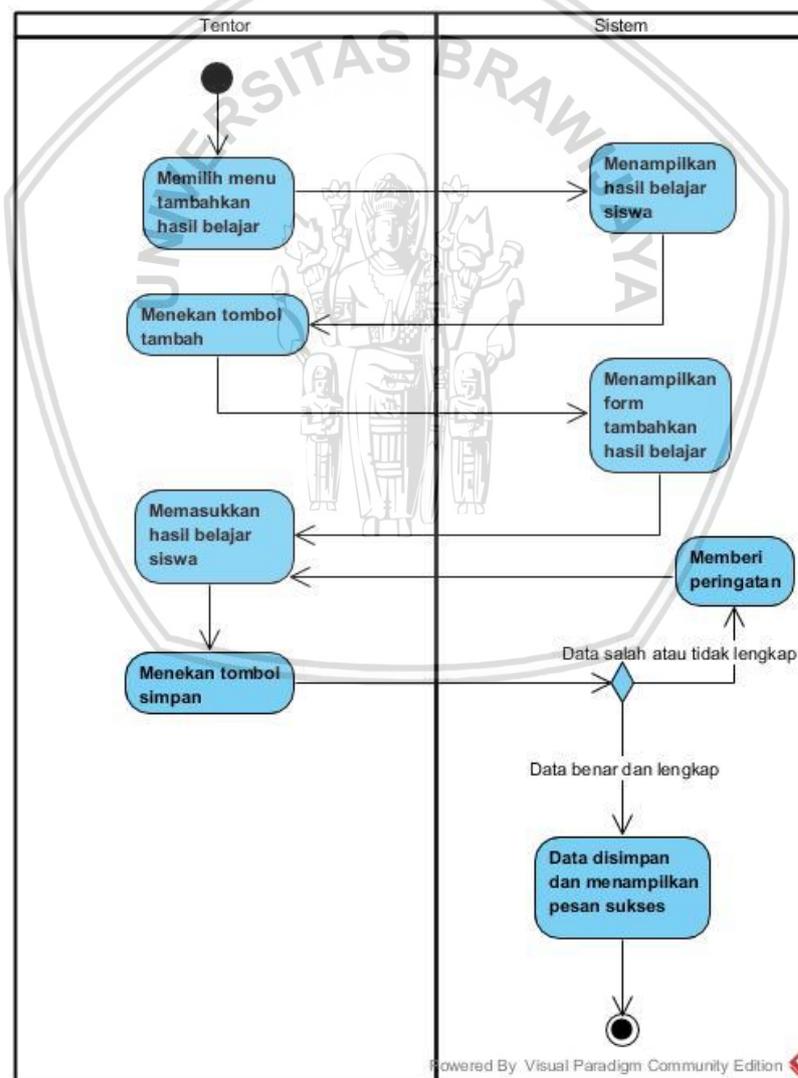
Item	Deskripsi
<i>Use Case Code</i>	UC-017
<i>Use Case Name</i>	Login
<i>Actor</i>	Pengguna
<i>Brief Description</i>	<i>Use Case</i> untuk mengautentifikasi pengguna berdasarkan hak aksesnya
<i>Precondition</i>	Belum terdapat status login
<i>Basic Flow</i>	<ol style="list-style-type: none"> 1. Pengguna menambahkan <i>username</i> dan <i>password</i> 2. Sistem melakukan autentifikasi pengguna 3. Sistem menampilkan fitur – fitur berdasarkan hak akses pengguna
<i>Alternatif Flow</i>	<ol style="list-style-type: none"> 1. Jika <i>username</i> dan <i>password</i> tidak cocok maka akan kembali ke langkah 1
<i>Postcondition</i>	Pengguna berhasil terautentifikasi

4.10 Activity Diagram

Pada bab ini akan menjelaskan bagaimana alur kerja pada *Use Case* sistem. Dimana alur tersebut akan digambarkan dengan menggunakan notasi *Activity Diagram*.

4.10.1 Activity Diagram Menambahkan Data Hasil Belajar

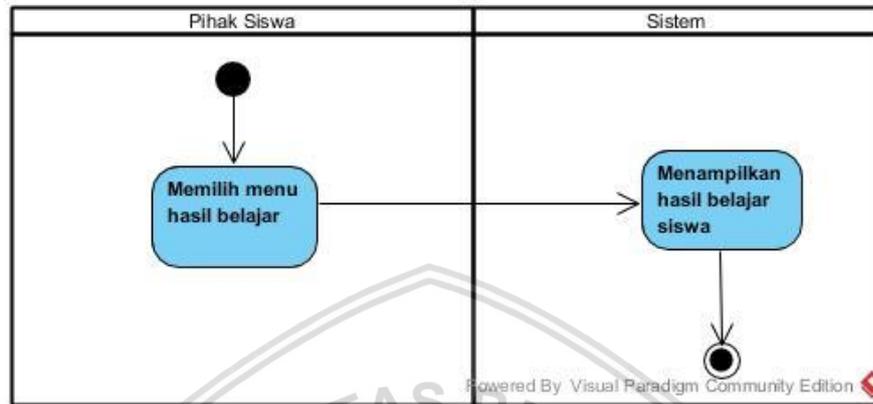
Berikut pada Gambar 4.5 adalah *Activity Diagram* menambahkan data hasil belajar. Dimana proses awal dimulai saat aktor bernama tentor sudah dalam kondisi login berada dalam sistem, tepatnya di halaman dashboard. Tmentor memilih nama siswa yang akan ditambahkan nilai atau hasil belajarnya. Kemudian sistem akan menampilkan *form* untuk diisi oleh tentor. Setelah diisi lengkap maka tentor akan memilih fungsi submit. Lalu sistem akan menyimpan data tersebut dan mengirimkan ke pengguna terkait. Jika data yang diisi tidak lengkap maka sistem akan menampilkan pesan “data harus diisi dengan lengkap”.



Gambar 4.10 Activity Diagram Menambahkan data hasil belajar

4.10.2 Activity Diagram Melihat Hasil Belajar Siswa

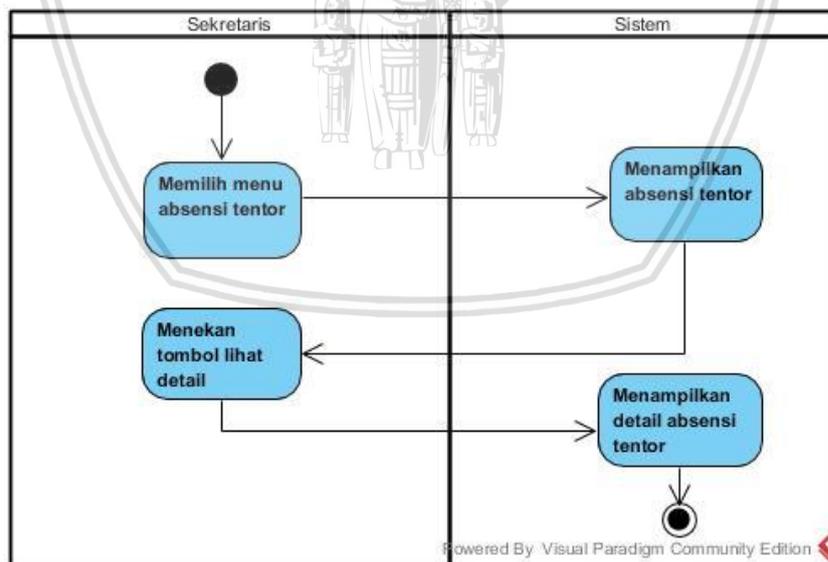
Berikut pada Gambar 4.11 adalah *Activity Diagram* menambahkan data hasil belajar. Dimana proses ini diawali ketika pengguna memilih fungsi lihat hasil belajar, kemudian sistem akan mengambil data tersebut dari database berdasarkan id tertentu. Kemudian menampilkannya ke pengguna.



Gambar 4.11 Activity Diagram Melihat data hasil belajar

4.10.3 Activity Diagram Melihat Presensi Tantor

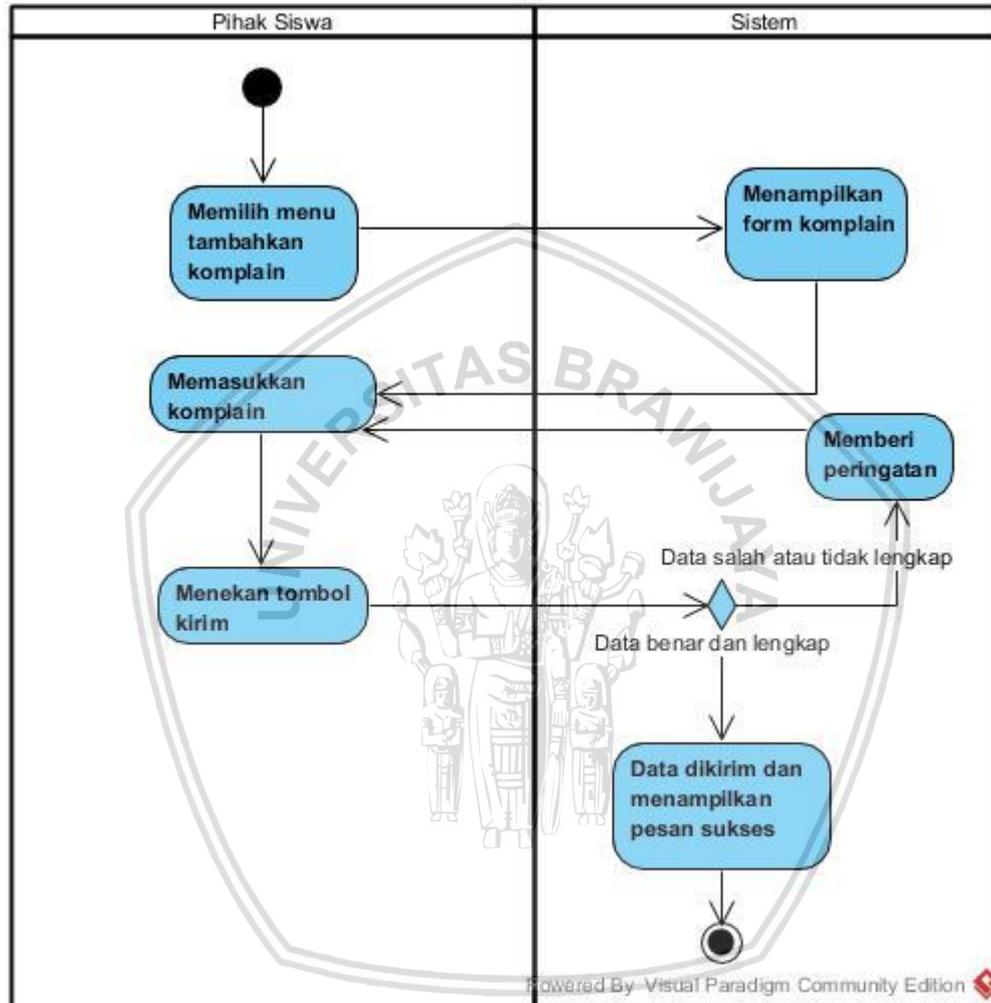
Berikut pada Gambar 4.12 adalah *Activity Diagram* melihat presensi tentor. Proses ini diawali ketika aktor sekretaris memilih fungsi presensi tentor. Kemudian sistem akan menampilkan *list* presensi tentor. Aktor dapat melihat detail presensi tentor dengan memilih fungsi lihat detail.



Gambar 4.12 Activity Diagram Melihat Presensi Tantor

4.10.4 Activity Diagram Melakukan Komplain

Berikut pada Gambar 4.7 adalah *Activity Diagram* memberi komplain. Proses ini diawali ketika aktor orang tua atau siswa memilih fungsi komplain. Kemudian sistem akan menampilkan *form* komplain. Dan pengguna akan mengisi *form* tersebut. Jika sudah selesai maka pengguna akan memilih fungsi submit dan sistem akan mengirim komplain ke sekretaris.



Gambar 4.13 Activity Diagram Memberi komplain

BAB 5 PERANCANGAN

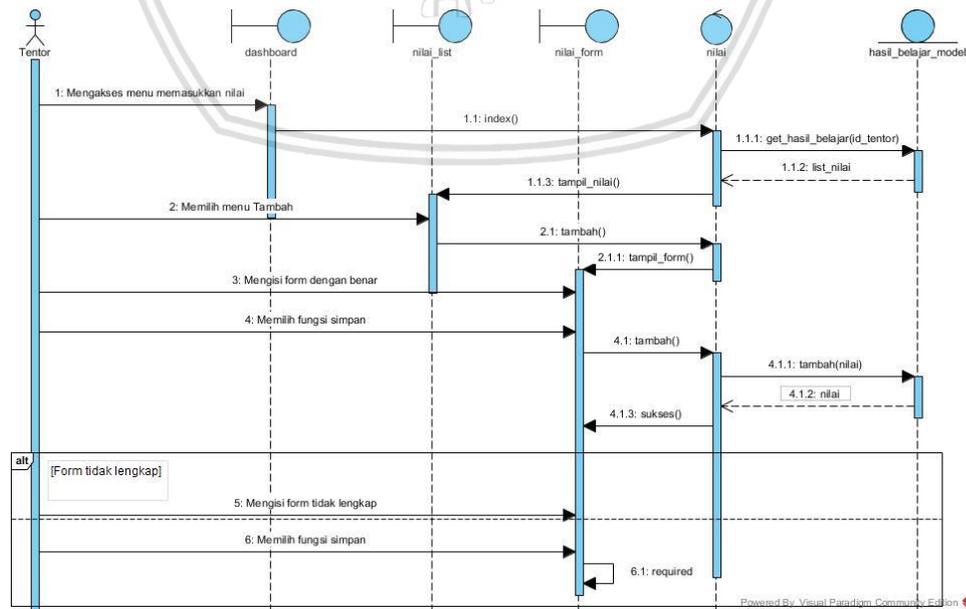
Pada bab ini akan menjelaskan tentang perancangan dari pengembangan sistem informasi lembaga bimbingan belajar Al-Hasyimi ini. Pada bab ini berisi beberapa prancangan diantaranya ada *sequence diagram*, *class diagram*, *physical data model*, perancangan algoritma menggunakan *pseudocode*, serta perancangan *user interface*.

5.1 Perancangan *Sequence Diagram*

Sequence diagram digunakan untuk melakukan visualisasi hubungan antara entitas aktor, entitas boundary, objek control, dan objek model dalam proses memenuhi kebutuhan pengguna. *Sequence diagram* dibuat berdasarkan pada beberapa *Use Case* yang sudah dibuat pada bab sebelumnya. Beberapa *Use Case* tersebut diantaranya, menambahkan data hasil belajar, melihat hasil belajar, melihat presensi, dan mengirim komplain.

5.1.1 *Sequence Diagram* Menambahkan Hasil Belajar

Pada Gambar 5.1 menjelaskan tentang *sequence diagram* menambahkan data hasil belajar. Dimana pada *sequence diagram* ini terdapat aktor tentor, dashboard, nilai_list, dan nilai_form sebagai objek boundary, nilai sebagai objek control, hasil_belajar_model sebagai objek model. Dimana proses awal dimulai ketika aktor tentor sudah dalam kondisi login mengakses halaman dashboard, kemudian memilih menu masukkan nilai. *Controller* nilai akan menampilkan list nilai yang sudah ditambahkan sebelumnya. Kemudian aktor memilih fungsi tambah untuk menambahkan nilai, untuk selanjutnya disimpan di tabel hasil belajar. Jika pengisian *form* ada yang kosong atau salah maka akan muncul pemberitahuan, dan aktor diminta untuk menambahkan data kembali dengan benar.

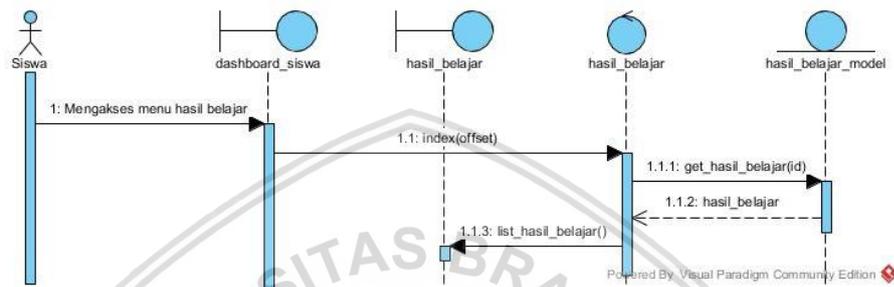


Gambar 5.1 Squence Diagram Menambahkan Hasil Belajar



5.1.2 Sequence Diagram Melihat Hasil Belajar Siswa

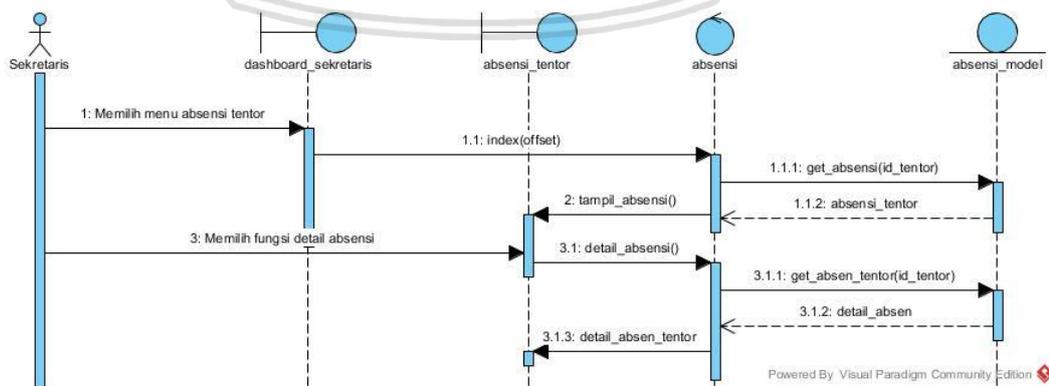
Pada Gambar 5.2 menjelaskan tentang *sequence diagram* melihat hasil belajar siswa. Dimana pada *sequence diagram* ini terdapat aktor siswa, dashboard_siswa dan hasil_belajar sebagai objek boundary, hasil_belajar sebagai objek control dan hasil_belajar_model sebagai objek model. Dimana proses awal dimulai ketika aktor sudah dalam kondisi login, pada halaman dashboard memilih menu hasil belajar, kemudian controller hasil_belajar akan mengambil data hasil belajar pada siswa menggunakan id siswa.



Gambar 5.2 Sequence Diagram Melihat Hasil Belajar

5.1.3 Sequence Diagram Melihat Presensi Tantor

Pada Gambar 5.3 menjelaskan tentang *sequence diagram* melihat presensi tentor. Dimana pada *sequence diagram* ini terdapat aktor sekretaris, dashboard_sekretaris, dan presensi_tantor sebagai objek boundary, presensi sebagai objek control, dan presensi_model objek model. Dimana proses ini diawali dengan aktor sekretaris sudah dalam kondisi login, kemudian mengakses menu presensi tentor dan pada halaman dashboard_sekretaris akan meminta controller presensi untuk mengambil data hasil belajar berdasarkan id_tantor. Kemudian ditampilkan list presensi tentor. Selain itu aktor juga dapat melihat detail presensi tentor.

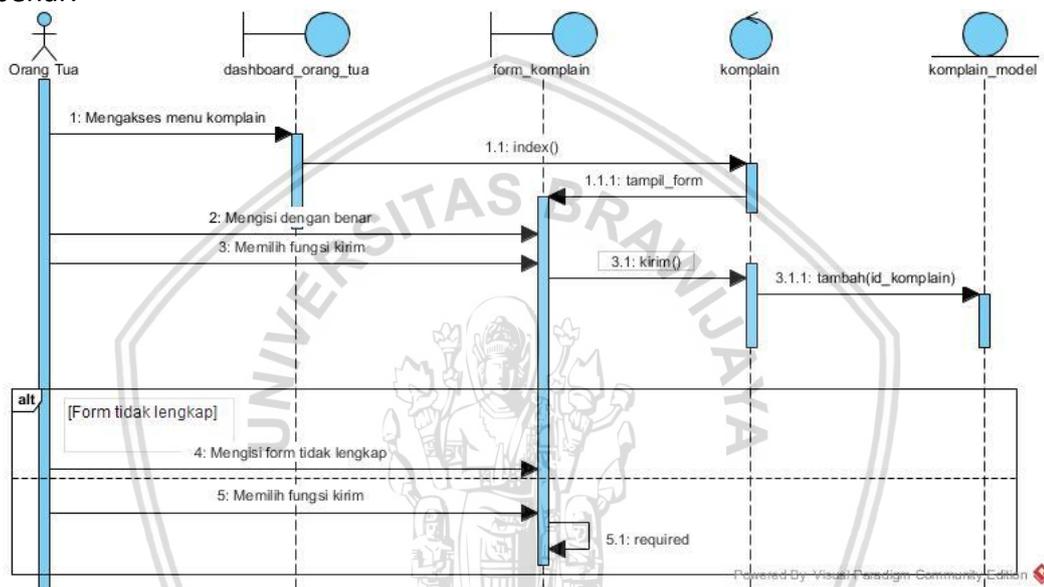


Gambar 5.3 Sequence Diagram Melihat Presensi Tantor



5.1.4 Sequence Diagram Melakukan Komplain

Pada Gambar 5.4 menjelaskan tentang *sequence diagram* mengirim komplain. Dimana pada *sequence diagram* ini terdiri dari aktor siswa atau orang tua, *dashboard_orang_tua* dan *form_komplain* sebagai objek boundary, *komplain* sebagai objek control, dan *komplain_model* sebagai objek model. Dimana proses awal dimulai dari siswa atau orang tua mengakses menu komplain pada halaman dashboard, kemudian controller komplain akan menampilkan *form* komplain dan siswa atau orang tua yang ingin melakukan komplain mengisi *form* komplain. Terakhir memilih fungsi kirim untuk mengirim komplain tersebut. Jika ada *form* yang tidak lengkap, maka akan diberi pemberitahuan untuk mengisinya dengan benar.



Gambar 5.4 Sequence Diagram Melakukan Komplain

5.2 Pemodelan Object

Pada bab ini akan dijelaskan tentang pendokumentasian beberapa objek yang terdapat pada sistem. Diagram kelas digunakan sebagai media visualisasi daripada pemodelan objek tersebut. Terdapat 2 diagram kelas yang akan dijelaskan pada bab ini, yaitu diagram kelas analisis dan diagram kelas perancangan. Diagram kelas analisis digunakan sebagai dasar dalam pembuatan *physical data model*. Diagram kelas perancangan digunakan untuk memvisualisasikan kelas-kelas dalam sistem dan hubungan antar kelas.

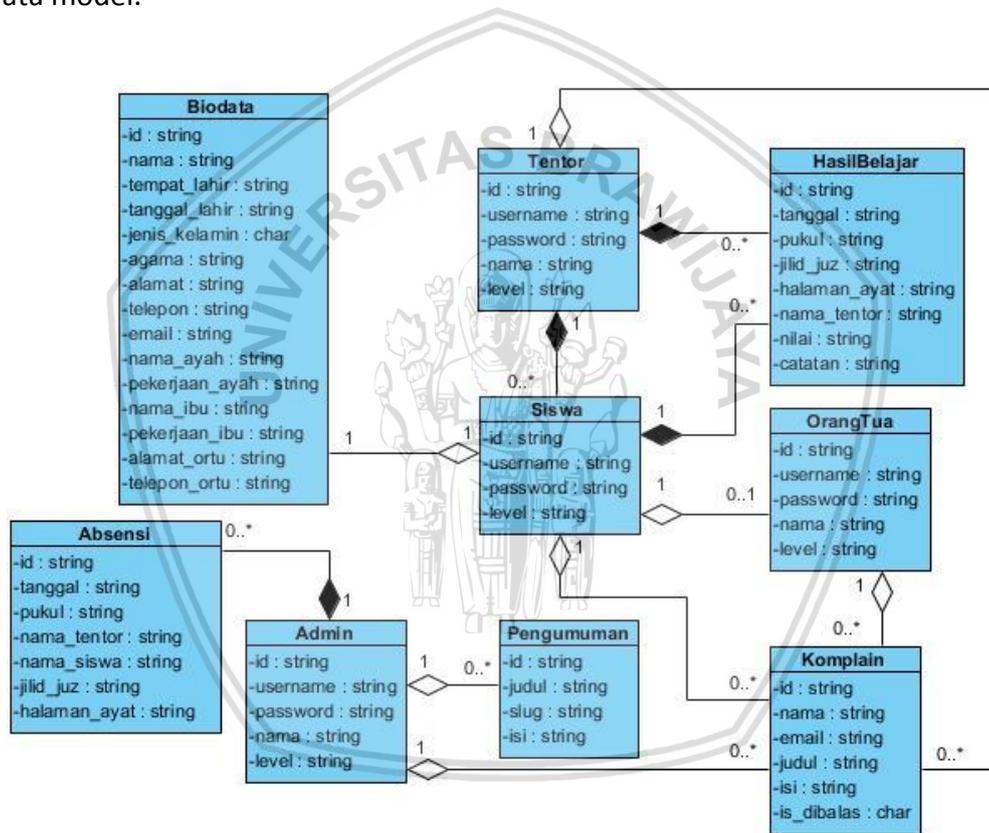
5.2.1 Diagram Kelas Analisis

Pada Gambar 5.5 merupakan diagram kelas analisis yang memvisualisasikan hubungan antar kelas yang memiliki atribut-atribut tertentu. Diagram kelas analisis ini akan digunakan untuk panduan dalam pemodelan kelas perancangan dan perancangan *database*. Terdapat 8 kelas pada diagram kelas analisis yang beberapa diantaranya memiliki hubungan satu sama lain. Kelas tersebut

diantaranya kelas Tentor, Siswa, OrangTua, Admin, Biodata, HasilBelajar, Pengumuman, Komplain, dan Presensi.

Pada kelas Tentor dapat memiliki banyak HasilBelajar dan banyak Siswa dimana hubungan diantaranya memiliki kebergantungan yang kuat. Kemudian kelas Siswa dapat memiliki satu Tentor, satu Biodata, satu OrangTua, banyak Komplain dan banyak HasilBelajar. Kelas Admin dapat memiliki banyak Presensi dan Pengumuman. Kelas OrangTua dapat memiliki banyak Siswa dan banyak Komplain.

Pemilihan nama kelas dan atribut ini berdasarkan dari analisis kebutuhan sistem yang sudah dijelaskan pada bab analisis kebutuhan. Penamaan kelas dapat dijadikan acuan dalam perancangan kelas diagram model dan controller. Sedangkan penamaan atribut dapat dijadikan acuan sebagai perancangan physical data model.



Gambar 5.5 Diagram Kelas Analisis

5.2.2 Diagram Kelas Perancangan

Pada bab ini akan dijelaskan tentang diagram kelas perancangan yang dibuat berdasarkan dari diagram kelas analisis pada Gambar 5.5, serta objek-objek pada *sequence diagram* yang menggunakan pola *Model-View-Controller* dengan memanfaatkan *framework CodeIgniter*. Diagram kelas perancangan terdiri dari diagram kelas model dan diagram kelas controller.

Pada Gambar 5.6 merupakan visualisasi dari diagram kelas model. Pada diagram kelas model ini terdapat 15 kelas model. Kelas *my_model* merupakan

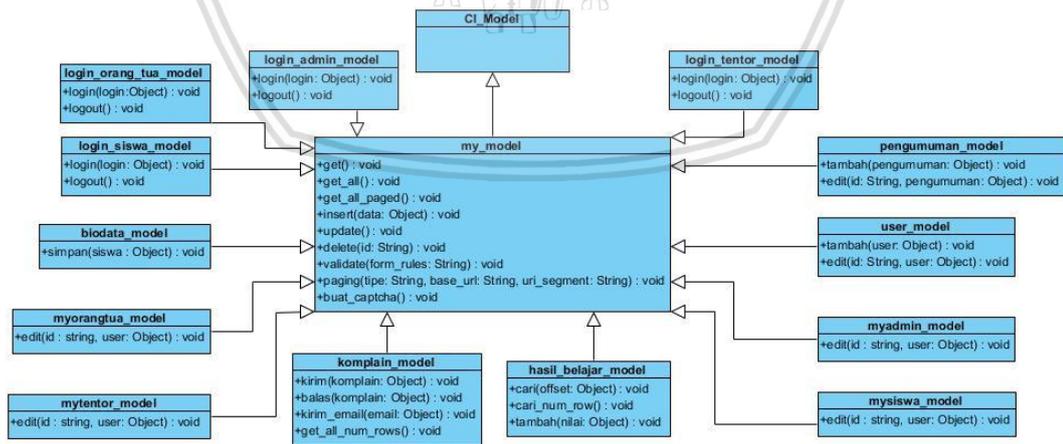


kelas induk yang meng-*extends* kelas *CI_Model* yang merupakan bawaan dari *framework CodeIgniter*. Kelas *my_model* ini dibuat untuk membuat pemrograman menjadi lebih efisien dan menghindari terjadinya redundansi. Kelas tersebut akan di-*extends* oleh kelas – kelas model lainnya.

Kelas *login_model* digunakan untuk mengambil data pengguna sebagai autentifikasi hak akses pengguna dari tabel tiap tipe pengguna. Kelas *biodata_model* digunakan untuk mengambil data biodata siswa dari tabel *tb_biodata*. Kelas *pengumuman_model* digunakan untuk memasukkan dan mengambil data pengumuman dari tabel *tb_pengumuman*. Kelas *komplain_model* digunakan untuk memasukkan dan mengambil data *tb_komplain* dari tabel *komplain*. Kelas *hasil_belajar_model* digunakan untuk memasukkan dan mengambil data hasil belajar dari tabel *tb_hasil_belajar*.

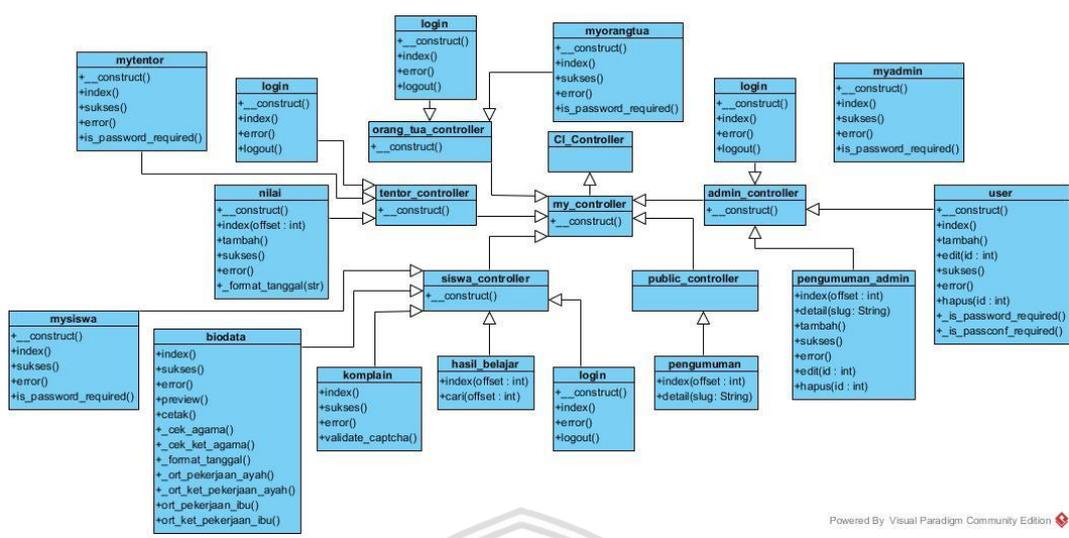
Pada Gambar 5.7 merupakan visualisasi dari diagram kelas controller. Pada diagram kelas controller ini terdapat 22 kelas controller. Hampir sama dengan diagram kelas model, diagram kelas controller ini memiliki kelas *my_controller* yang merupakan kelas induk yang meng-*extends* kelas *CI_Controller* dimana kelas ini merupakan bawaan dari *framework CodeIgniter*. Kelas *my_controller* ini dibuat untuk memanggil kelas model sesuai dengan nama kelas controller-nya. Kelas tersebut akan di-*extends* oleh kelas – kelas controller lainnya.

Terdapat kelas *tentor_controller*, *dashboard_controller*, dan *admin_controller* yang berguna untuk mengambil data *session* dari tiap pengguna. Kemudian *tentor_controller* merupakan induk kelas dari kelas nilai yang berguna untuk menambahkan hasil belajar siswa. Kelas *dashboard_controller* merupakan induk kelas dari kelas *biodata*, *komplain*, dan *hasil belajar*. Kelas *admin_controller* merupakan induk kelas dari kelas *pengumuman_admin*, *user*, dan kelas *presensi*.



Gambar 5.6 Diagram Kelas Model





Gambar 5.7 Diagram Kelas Controller

5.3 Perancangan Database

Pada Gambar 5.8 menjelaskan tentang rancangan *database* yang dibuat berdasarkan diagram kelas analisis pada Gambar 5.10. Perancangan *database* ini menggunakan *Physical Data Model* (PDM) untuk menggambarkan struktur *database* yang dibutuhkan untuk memenuhi fungsionalitas sistem. Dalam sistem ini terdapat 10 tabel yang mana tiap tabel ada yang memiliki hubungan dengan tabel yang lain. Pada Tabel 5.1 akan dijelaskan masing-masing fungsi tabel pada sistem ini.

Table 5.1 Penjelasan Fungsi Tabel

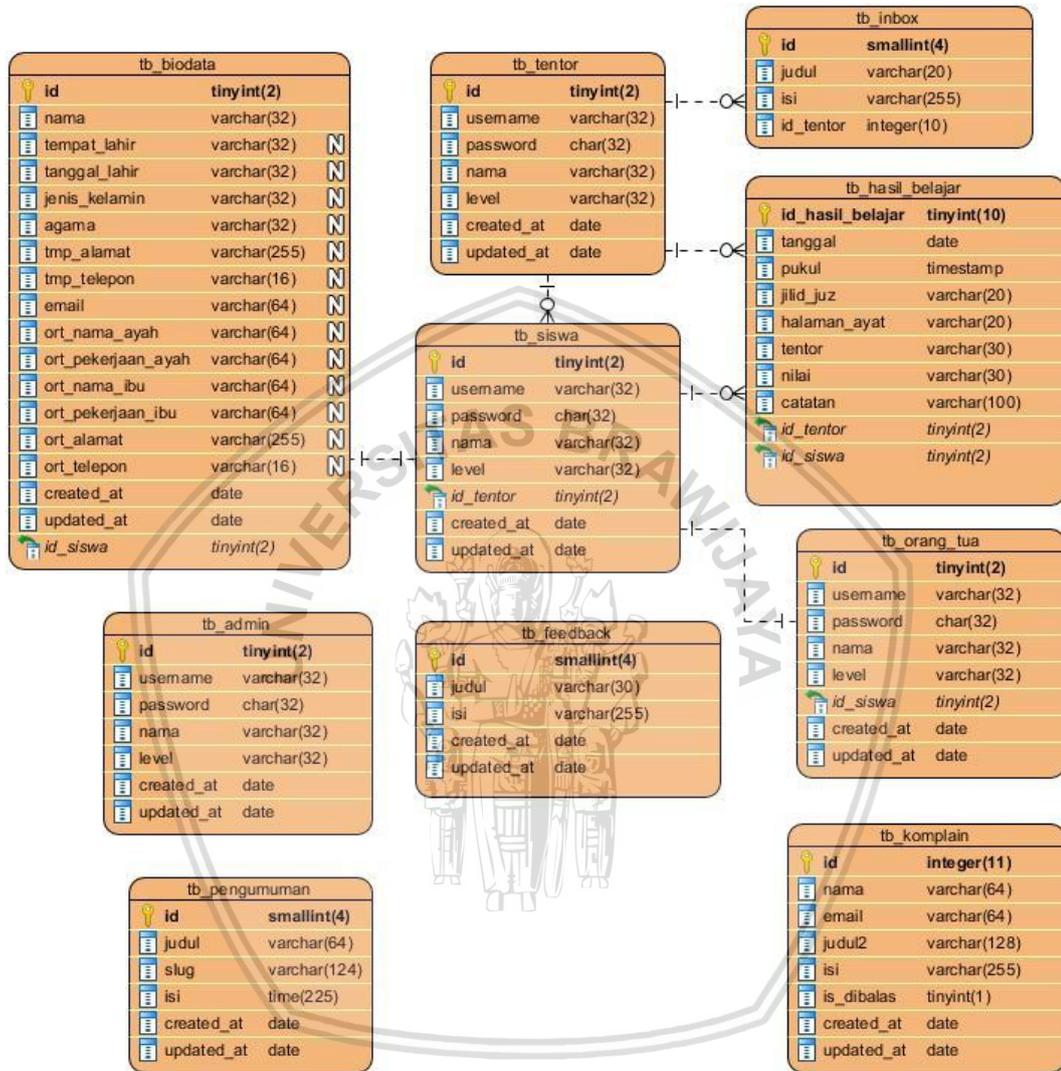
No	Nama Tabel	Fungsi Tabel
1.	tb_admin	Tabel ini digunakan untuk menyimpan data dari admin serta digunakan untuk pembeda hak akses antara pengguna yang memiliki level admin dengan pengguna lain.
2.	tb_tentor	Tabel ini digunakan untuk menyimpan data dari tentor serta digunakan untuk pembeda hak akses antara pengguna yang memiliki level tentor dengan pengguna lain. Tabel ini memiliki hubungan dengan tabel tb_siswa dan tb_hasil_belajar, dimana tiap tentor dapat memiliki banyak siswa dan tentor juga sebagai pengelola data hasil belajar siswa.
3.	tb_siswa	Tabel ini digunakan untuk menyimpan data siswa serta digunakan untuk pembeda hak akses antara pengguna yang memiliki level siswa dengan pengguna yang lain. Tabel ini memiliki hubungan dengan tabel tb_tentor,

Tabel 5.1 Penjelasan Fungsi Tabel (lanjutan)

No	Nama Tabel	Fungsi Tabel
		tb_orang_tua, tb_hasil_belajar, dan tb_biodata. Setiap siswa memiliki satu tentor dan orang tua. Kemudian tiap siswa dapat memiliki beberapa hasil belajar. Terakhir tiap siswa memiliki satu biodata.
4.	tb_orang_tua	Tabel ini digunakan untuk menyimpan data orang tua serta digunakan untuk pembeda hak akses antara pengguna yang memiliki level orang tua dengan pengguna yang lain. Tabel ini memiliki hubungan dengan tabel tb_siswa, dimana setiap orang tua dapat memiliki satu siswa atau anak.
5.	tb_hasil_belajar	Tabel ini digunakan untuk menyimpan data hasil belajar siswa. Tabel ini memiliki hubungan dengan tabel tb_tentor sebagai pengelola data hasil belajar siswa dan tabel tb_siswa sebagai pengguna yang memiliki dan dapat melihat data hasil belajar tersebut.
6.	tb_biodata	Tabel ini digunakan untuk menyimpan data biodata siswa. Tabel ini memiliki hubungan dengan tabel tb_siswa, dimana tiap siswa memiliki biodatanya masing-masing.
7.	tb_komplain	Tabel ini digunakan untuk menyimpan data komplain.
8.	tb_pengumuman	Tabel ini digunakan untuk menyimpan data pengumuman.
9.	tb_inbox	Tabel ini digunakan untuk menyimpan data inbox ke tentor
10.	tb_feedback	Tabel ini digunakan untuk menyimpan data feedback ke pihak siswa

Berikut pada Gambar 5.8 merupakan desain dari *physical data model* pada *database* Sistem Informasi Lembaga Bimbingan Belajar Al-Hasyimi. Desain *database* ini dibuat berdasarkan perancangan kelas analisis. Namun terdapat perbedaan disini, dimana kelas presensi dijadikan satu pada tabel tb_hasil_belajar. Hal ini dikarenakan untuk mengambil data presensi didapat dari tabel tb_hasil_belajar dengan mengakumulasi jumlah id_hasil_belajar sesuai dengan id_tentor pada tabel tb_hasil_belajar.

Terdapat tiga tabel yang tidak memiliki relasi antara satu dengan yang lain. Tabel tersebut adalah `tb_admin`, `tb_pengumuman`, dan `tb_komplain`. Hal tersebut dikarenakan untuk tidak ada id dari tabel lain yang mereference pada ketiga tabel tersebut. Kemudian untuk mengoperasikan ketiga tabel tersebut dapat memanfaatkan bahasa MySQL.



Gambar 5.8 Physical Data Model

5.4 Perancangan Algoritma

Pada bab ini akan dijelaskan tentang algoritma program dalam menyusun operasi logika pada fungsi – fungsi sistem. Algoritma ini dibuat sebagai panduan dalam melakukan implementasi sistem. Algoritma dibuat menggunakan *pseudocode*.

5.4.1 Pseudocode Menambahkan Hasil Belajar

Pada Tabel 5.2 merupakan algoritma *pseudocode* menambahkan hasil belajar. Pada *pseudocode* ini diasumsikan pengguna sudah berhasil diinisiasi id-



nya. Algoritma ini dimulai dengan melakukan konversi data hasil belajar yang sebelumnya masih dalam bentuk array menjadi object. Kemudian data dalam bentuk object tersebut akan dimasukkan ke *database* hasil belajar. Jika pengisian *form* belum lengkap atau terdapat kesalahan, maka sistem akan memberi pemberitahuan dan data tidak akan dimasukkan ke tabel. Jika sudah benar, maka akan dimasukkan ke database hasil belajar dan sistem akan memberikan pesan sukses kepada pengguna.

Table 5.2 Pseudocode Menambahkan Hasil Belajar

No	Pseudeucode
1	Mulai
2	Mengkonversi data hasil belajar menjadi object
3	Menambahkan data ke tabel hasil belajar setelah dilakukan validasi <i>form</i>
4	Jika tidak lengkap atau salah, maka muncul peringatan data harus diisi dengan benar
5	Jika data diisi dengan benar, maka data berhasil ditambahkan
6	Kembali kehalaman hasil belajar
7	Selesai

5.4.2 Pseudocode Melihat Hasil Belajar

Pada Tabel 5.3 merupakan algoritma *pseudocode* melihat hasil belajar. Pada *pseudocode* ini diasumsikan pengguna sudah berhasil diinisiasi id-nya. Algoritma ini dimulai dengan mengambil data hasil belajar dari tabel hasil belajar. Kemudian akan dilakukan pengecekan apakah data hasil belajar tersebut ada. Jika tidak maka sistem akan menampilkan pesan data hasil belajar belum ada. Jika terdapat data hasil belajar, maka sistem akan menampilkan data hasil belajar tersebut.

Table 5.3 Pseudocode Melihat Hasil Belajar

No	Pseudeucode
1	Mulai
2	Mengambil data hasil belajar berdasarkan id pengguna yang login
3	Melakukan pengecekan apakah terdapat data hasil belajar pada id tersebut
4	Jika data hasil belajar masih kosong, maka akan memberikan pesan data hasil belajar belum ada
5	Jika terdapat data hasil belajar, maka akan menampilkan data hasil belajar
6	Selesai

5.4.3 Pseudocode Melihat Presensi Tentor

Pada Tabel 5.4 merupakan algoritma *pseudocode* melihat presensi tentor. Pada *pseudocode* ini diasumsikan pengguna sudah berhasil diinisiasi id-nya. Algoritma ini dimulai dengan melakukan pengambilan data presensi dari tabel hasil belajar yang dijoin dengan tabel tentor, karena presensi ini diambil



berdasarkan id tentor. Kemudian dilakukan akumulasi penjumlahan menggunakan count pada id hasil belajar. Jika tidak terdapat data presensi, maka sistem akan memberikan pesan data presensi tidak ada. Jika terdapat data presensi, maka sistem akan menampilkan data presensi tersebut yang sudah digroup berdasarkan id tentor.

Table 5.4 Pseudocode Melihat Presensi Tentor

No	Pseudeuocode
1	Mulai
2	Mengambil data dari tabel hasil belajar
3	Mengakumulasi jumlah pertemuan berdasarkan id hasil belajar dan id tentor
4	Menampilkan data presensi menggunakan group by id tentor
5	Melakukan pengecekan apakah terdapat data presensi
6	Jika tidak maka akan menampilkan pesan tidak ada data presensi
7	Jika ada maka akan menampilkan data presensi
8	Selesai

5.4.4 Pseudocode Melakukan Komplain

Pada Tabel 5.5 merupakan algoritma *pseudocode* melakukan komplain. Algoritma ini dimulai dengan melakukan konversi data komplain yang sebelumnya masih dalam bentuk array menjadi object. Jika pengisian *form* belum lengkap atau terdapat kesalahan, maka sistem akan memberi pemberitahuan dan data tidak akan dimasukkan ke tabel. Jika sudah benar, maka akan dimasukkan ke database hasil belajar dan sistem akan memberikan pesan sukses kepada pengguna.

Table 5.5 Pseudocode Melakukan Komplain

No	Pseudeuocode
1	Mulai
2	Mengkonversi data komplain menjadi object
3	Jika tidak lengkap atau salah, maka muncul peringatan data harus diisi dengan benar
4	Jika data diisi dengan benar, maka data berhasil ditambahkan
5	Kembali kehalaman dashboard
6	Selesai

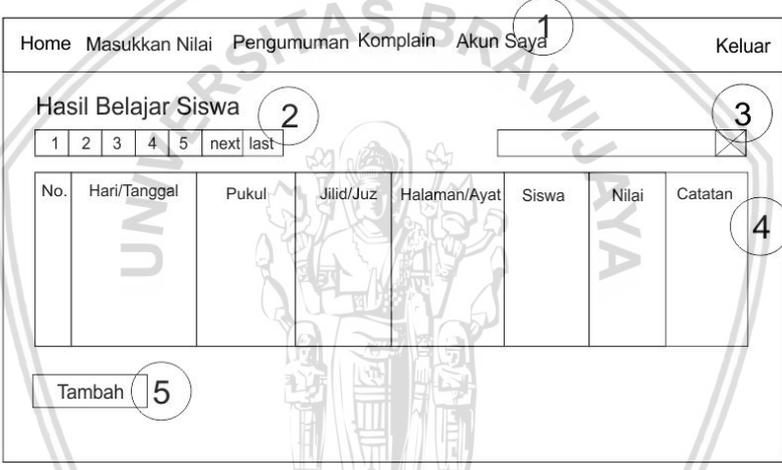
5.5 Perancangan User Interface

Pada bab ini akan dijelaskan tentang perancangan *user interface* dari Sistem Informasi Lembaga Bimbingan Belajar Al-Hasyimi ini. Perancangan ini dibuat berdasarkan fungsi utama yang ada pada sistem. Selain itu perancangan ini juga mempertimbangkan kemudahan pengguna dalam mengoperasikan sistem ini. Perancangan *user interface* yang akan dijelaskan dalam bab ini diantaranya adalah menambahkan hasil belajar siswa, melihat hasil belajar siswa, melihat presensi

tentor, dan melakukan komplain. Dalam penjelasan *user interface* ini juga disertai dengan penomoran pada bagian tertentu untuk mempermudah pembaca dalam memahami perancangan *user interface* tersebut.

5.5.1 User Interface Menambahkan Hasil Belajar

Pada Gambar 5.9 menjelaskan tentang rancangan *user interface* menambahkan hasil belajar. Setelah memilih menu “Tambahkan Hasil Belajar” pada halaman dashboard tentor, selanjutnya sistem akan menampilkan hasil belajar siswa yang dibimbingnya. Pada bagian nomor 1 merupakan baris menu yang disediakan pada halaman dashboard tentor. Nomor 2 merupakan fungsi *paging* agar baris yang ditampilkan tidak terlalu panjang, sehingga pengguna tidak perlu untuk scroll ke bawah. Nomor 3 merupakan fungsi pencarian untuk memudahkan pengguna dalam mencari data yang dibutuhkan. Nomor 4 merupakan tabel hasil belajar siswa yang dibimbing oleh tentor tersebut. Nomor 5 merupakan tombol fungsi tambah untuk menambahkan hasil belajar baru.



Gambar 5.9 User Interface Hasil Belajar Siswa

Kemudian setelah memilih fungsi tambah, maka sistem akan menampilkan *form* hasil belajar siswa yang harus diisi oleh tentor. Rancangan *user interface form* hasil belajar ditunjukkan pada Gambar 5.10 ini. Pada nomor 2 merupakan tampilan field *form* yang harus diisi dan Nomor 3 merupakan tombol fungsi simpan untuk menyimpan data yang sudah dimasukkan. Pada *form* hasil belajar terdapat 7 field yang wajib diisi oleh tentor, diantaranya ada nama siswa, hari/tanggal, pukul, jilid/juz, halaman/ayat, nilai dan catatan.

The screenshot shows a web form titled "Masukkan Nilai". At the top, there is a navigation bar with links: "Home", "Masukkan Nilai", "Pengumuman", "Komplain", "Akun Saya" (circled with 1), and "Keluar". The form itself has several input fields: "Siswa" (a dropdown menu circled with 2), "Hari/Tanggal", "Pukul", "Jilid/Juz" (radio buttons), "Halaman/Ayat" (radio buttons), and "Nilai". To the right of these fields is a "Catatan" text area. Below the "Pukul" field is a "Simpan" button circled with 3. At the bottom right, there is a "Keterangan Nilai:" text area.

Gambar 5.10 User Interface Form Hasil Belajar

5.5.2 User Interface Hasil Belajar Siswa

Pada Gambar 5.11 menjelaskan tentang rancangan *user interface* hasil belajar siswa. Tampilan ini hampir sama dengan tampilan hasil belajar pada halaman dashboard tentor. Namun pada halaman ini tidak terdapat tombol tambah dan terdapat gambar logo yang ditunjukkan pada nomor 1. Pada bagian nomor 2 merupakan baris menu yang disediakan pada halaman dashboard tentor. Nomor 3 merupakan fungsi *paging* agar baris yang ditampilkan tidak terlalu panjang, sehingga pengguna tidak perlu untuk scroll ke bawah. Nomor 4 merupakan fungsi pencarian untuk memudahkan pengguna dalam mencari data yang dibutuhkan. Nomor 5 merupakan tabel hasil belajar.

The screenshot shows a web page titled "Hasil Belajar Siswa". At the top left is a "LOGO" box circled with 1. Below it is a navigation bar with links: "Home", "Hasil Belajar", "Pengumuman", "Komplain", "Akun Saya" (circled with 2), and "Keluar". The main content area has a "Hasil Belajar Siswa" header circled with 3. Below the header is a *paging* control with buttons "1", "2", "3", "4", "5", "next", and "last", and a search input field circled with 4. Below the search field is a table circled with 5. The table has the following columns: "No.", "Hari/Tanggal", "Pukul", "Jilid/Juz", "Halaman/Ayat", "Tentor", "Nilai", and "Catatan".

Gambar 5.11 User Interface Hasil Belajar Siswa

5.5.3 User Interface Presensi Tantor

Pada Gambar 5.12 menjelaskan tentang rancangan *user interface* presensi tentor. Pada bagian nomor 1 merupakan baris menu yang disediakan pada halaman dashboard admin. Nomor 2 merupakan fungsi *paging* agar baris yang ditampilkan tidak terlalu panjang, sehingga pengguna tidak perlu untuk scroll ke bawah. Nomor 3 merupakan fungsi pencarian untuk memudahkan pengguna

dalam mencari data yang dibutuhkan. Nomor 4 merupakan tabel presensi tentor siswa yang dibimbing oleh tentor tersebut.

Gambar 5.12 User Interface Presensi Tendor

5.5.4 User Interface Form Komplain

Pada Gambar 5.13 menjelaskan tentang rancangan *user interface form* komplain. Pada bagian nomor 2 merupakan baris menu yang disediakan pada halaman dashboard. Pada nomor 3 merupakan tampilan field *form* yang harus diisi dan Nomor 4 merupakan tombol fungsi kirim untuk mengirim komplain.

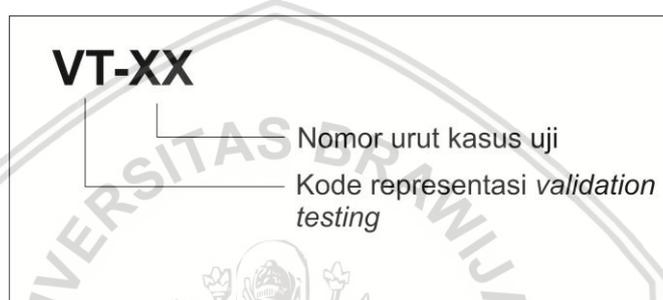
Gambar 5.13 User Interface Form Komplain

5.6 Perancangan Pengujian

Perancangan pengujian ini dibuat sebagai acuan dalam melaksanakan pengujian sistem informasi LBB Al-Hasyimi. Dalam pengujian ini, terdapat 3 metode pengujian yang digunakan, yaitu *validation testing*, *compatibility testing*, dan *user acceptance testing*.

5.6.1 Validation Testing

Validation testing merupakan metode pengujian yang bertujuan untuk memastikan apakah kebutuhan fungsional sistem sudah berjalan sesuai dengan harapan pengguna. Pada pengujian ini akan dilakukan pada 4 fitur utama sistem informasi LBB Al-Hasyimi. Dalam kasus uji pengujian ini terdapat dua jenis kasus uji, yaitu *basic flow* dan *alternative flow* atau yang disebut *alt1*.



Gambar 5.14 Kodifikasi Validation Testing

Perancangan pengujian akan ditampilkan pada tabel dimana tiap tabel terdapat beberapa baris diantaranya *test case code* yaitu kode kasus uji yang dijelaskan pada Gambar 5.14, *test case name* yaitu nama dari kasus uji tersebut, *code Use Case* yang merupakan kode dari *Use Case* yang sama pada pengujian ini, *testing objectives* yaitu tujuan dari tiap kasus uji, *test case* yaitu nama kasus uji, *testing step* yaitu langkah-langkah pengujian, *expectation* yaitu harapan dari dilakukannya pengujian tersebut, *result* yaitu hasil dari pengujian, terakhir adalah *status* yang apakah pengujian *valid* atau tidak.

Berikutnya merupakan perancangan *validation testing* yang akan dijabarkan pada tiap tabel. Tabel 7.1 merupakan perancangan *validation testing basic flow* ketika tentor menambahkan hasil belajar siswa dengan benar. Tabel 7.2 merupakan perancangan *validation testing alternative flow* ketika tentor tidak mengisi *form* hasil belajar dengan lengkap dan benar. Tabel 7.3 merupakan perancangan *validation testing basic flow* ketika siswa melihat hasil belajar siswa. Tabel 7.4 merupakan perancangan *validation testing basic flow* ketika sekretaris melihat presensi siswa. Tabel 7.5 merupakan *validation testing basic flow* ketika orang tua melakukan komplain dengan benar. Terakhir Tabel 7.6 merupakan perancangan *validation testing alternative flow* ketika orang tua melakukan komplain dengan tidak mengisi *form* secara lengkap dan benar.

Table 5.6 Kasus Uji Menambahkan Hasil Belajar Siswa: Basic Flow

Test Case Code	VT-001
Test Case Name	Menambahkan hasil belajar siswa
Code Use Case	UC-SIBBA-01
Testing Objectives	Pengujian ini bertujuan agar tentor dapat menambahkan hasil belajar siswa
Test Case	Mengisi seluruh field hasil belajar
Testing Step	<ol style="list-style-type: none"> 1. Penguji memilih menu masukkan nilai belajar 2. Penguji memilih menuh tambah 3. Penguji mengisi seluruh field dengan benar 4. Penguji menekan tombol simpan
Expectation	Sistem menambahkan hasil belajar
Result
Status

Table 5.7 Kasus Uji Menambahkan Hasil Belajar Siswa: Alt1

Test Case Code	VT-002
Test Case Name	Menambahkan hasil belajar siswa
Code Use Case	UC-SIBBA-01
Testing Objectives	Pengujian ini bertujuan agar tentor tidak dapat menambahkan nilai jika field tidak diisi dengan lengkap
Test Case	Mengisi sebagian field hasil belajar siswa
Testing Step	<ol style="list-style-type: none"> 5. Penguji memilih menu masukkan nilai belajar 6. Penguji memilih menuh tambah 7. Penguji mengisi sebagian field dengan benar 8. Penguji menekan tombol simpan
Expectation	Sistem memberikan peringatan pada field yang kosong untuk diisi
Result
Status

Table 5.8 Kasus Uji Melihat Hasil Belajar Siswa: Basic Flow

Test Case Code	VT-003
Test Case Name	Melihat hasil belajar siswa
Code Use Case	UC-SIBBA-02
Testing Objectives	Pengujian ini bertujuan agar siswa dapat melihat hasil belajar
Test Case	Melihat hasil belajar
Testing Step	9. Penguji memilih menu masukkan hasil belajar
Expectation	Sistem menampilkan hasil belajar siswa
Result
Status

Table 5.9 Kasus Uji Melihat Presensi Tentor: Basic Flow

Test Case Code	VT-004
Test Case Name	Melihat presensi tentor
Code Use Case	UC-SIBBA-03
Testing Objectives	Pengujian ini bertujuan agar sekretaris dapat melihat presensi tentor
Test Case	Melihat presensi tentor
Testing Step	10. Penguji memilih menu presensi tentor 11. Penguji memilih menu lihat detail
Expectation	Sistem menampilkan detail presensi tentor
Result
Status

Table 5.10 Kasus Uji Melakukan Komplain: Basic Flow

Test Case Code	VT-005
Test Case Name	Melakukan komplain
Code Use Case	UC-SIBBA-04
Testing Objectives	Pengujian ini bertujuan agar orang tua atau siswa dapat melakukan komplain
Test Case	Melakukan komplain
Testing Step	12. Penguji memilih menu komplain

	13. Penguji mengisi <i>form</i> dengan lengkap dan benar 14. Penguji memilih fungsi tombol kirim
Expectation	Sistem mengirim komplain
Result
Status

Table 5.11 Kasus Uji Melakukan Komplain: Alt1

Test Case Code	VT-006
Test Case Name	Melakukan komplain
Code Use Case	UC-SIBBA-04
Testing Objectives	Pengujian ini bertujuan agar orang tua atau siswa tidak dapat mengirim komplain jika <i>form</i> tidak diisi dengan lengkap dan benar
Test Case	Melakukan komplain dengan mengkosongkan beberapa field
Testing Step	<ol style="list-style-type: none"> 1. Penguji memilih menu komplain 2. Penguji sebagian mengisi <i>form</i> 3. Penguji memilih fungsi tombol kirim
Expectation	Sistem memberikan pemberitahuan <i>form</i> harus diisi
Result
Status

5.6.2 Compatibility Testing

Compatibility testing merupakan metode pengujian yang bertujuan untuk mengetahui apakah sistem yang dikembangkan ini dapat berjalan dengan baik pada berbagai jenis *browser* dengan versi yang terbaru. Pengujian dengan metode ini dilakukan karena pada LBB Al-Hasyimi ini menggunakan beberapa *browser* yang berbeda baik menggunakan *desktop* bahkan *mobile*. Untuk melakukan *compatibility testing*, dibutuhkan sebuah aplikasi yang bernama SortSite. Aplikasi SortSite versi 5 yang didownload dari website resmi www.powermapper.com. Aplikasi SortSite akan mendeteksi beberapa kesalahan yang terjadi dari sistem yang diujikan, misal adanya *link* yang rusak, *script* yang tidak sesuai, format file yang salah, dsb. Pada Tabel 5.7 merupakan kategori penilaian tingkat permasalahan *compatibility* pada tiap *browser*.



Table 5.12 Kategori Tingkat Permasalahan *Compatibility*

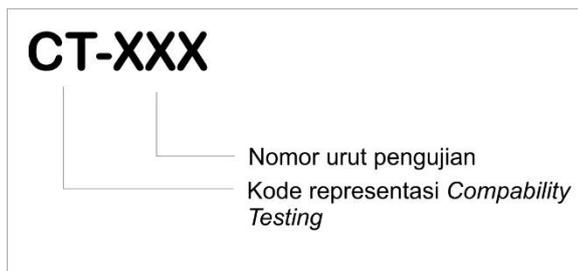
Simbol	Keterangan
	Tidak ada masalah baik pada konten maupun fungsionalitas sistem.
	Terdapat beberapa konten atau fungsi yang rusak atau tidak berjalan.
	Terdapat permasalahan mayor baik pada layout maupun performa.
	Terdapat permasalahan minor baik pada layout maupun performa.

Dalam *compatibility testing* ini akan menguji komponen-komponen pembangun sistem informasi Lembaga Bimbingan Al-Hasyim apakah komponen tersebut kompatibel dengan beberapa jenis *browser* yang menjadi kasus uji. Pada Tabel 5.15 menjelaskan jenis-jenis komponen yang akan diujikan.

Table 5.13 Komponen - Komponen yang Diujikan

No	Nama Komponen
1	CodeIgniter versi 2.2.0
2	Bootstrap versi 3.2.0
3	HTML5Shiv versi 3.7.2
4	Respond versi 1.4.2
5	JQuery versi 1.11.1
6	Bootstrap Datepicker
7	Hovernav
8	TinyMCE versi 4.1.6
9	Script style.css
10	Script javascript.js
11	Format gambar
12	Font Chaptcha

Pada Gambar 5.15 merupakan kodifikasi pengujian kompatibility yang akan digunakan untuk rancangan kasus uji. Rancangan kasus uji *compability testing* dijelaskan pada tabel – tabel berikut. Terdapat 16 kasus uji yang diambil dari tiap jenis dan versi *browser* sebagai kasus uji.



Gambar 5.15 Kodifikasi *Compability Testing*

Tabel 5.6 Kasus Uji *Compability Testing* pada *Browser Internet Explorer versi 9*

Test Case Code	CT-001
Test Case Name	<i>Compability Testing</i> pada <i>Browser Internet Explorer versi 9</i>
Testing Objectives	Pengujian ini bertujuan mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser Internet Explorer</i> versi 9
Test Case	Menguji <i>compability</i> sistem terhadap <i>browser Internet Explorer</i> versi 9
Testing Step	<ol style="list-style-type: none"> 1. Penguji membuka aplikasi SortSite 2. Penguji memasukkan url sistem pada menu bar <i>address</i> 3. Penguji melakukan <i>checking</i> dengan menekan tombol <i>check</i> 4. Penguji melihat hasil <i>compability testing</i>
Expectation	Sistem dapat berjalan dengan baik pada <i>browser Internet Explorer</i> versi 9
Result
Status

Tabel 5.7 Kasus Uji *Compability Testing* pada *Browser Internet Explorer versi 10*

Test Case Code	CT-002
Test Case Name	<i>Compability Testing</i> pada <i>Browser Internet Explorer versi 10</i>
Testing Objectives	Pengujian ini bertujuan mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser Internet Explorer</i> versi 10
Test Case	Menguji <i>compability</i> sistem terhadap <i>browser Internet Explorer</i> versi 10

Testing Step	<ol style="list-style-type: none"> 1. Penguji membuka aplikasi SortSite 5. Penguji memasukkan url sistem pada menu bar <i>address</i> 6. Penguji melakukan <i>checking</i> dengan menekan tombol <i>check</i> 7. Penguji melihat hasil <i>compability testing</i>
Expectation	Sistem dapat berjalan dengan baik pada <i>browser Internet Explorer</i> versi 10
Result
Status

Tabel 5.8 Kasus Uji *Compability Testing* pada *Browser Internet Explorer* versi 11

Test Case Code	CT-003
Test Case Name	<i>Compability Testing</i> pada <i>Browser Internet Explorer</i> versi 11
Testing Objectives	Pengujian ini bertujuan mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser Internet Explorer</i> versi 11
Test Case	Menguji <i>compability</i> sistem terhadap <i>browser Internet Explorer</i> versi 11
Testing Step	<ol style="list-style-type: none"> 1. Penguji membuka aplikasi SortSite 2. Penguji memasukkan url sistem pada menu bar <i>address</i> 3. Penguji melakukan <i>checking</i> dengan menekan tombol <i>check</i> 4. Penguji melihat hasil <i>compability testing</i>
Expectation	Sistem dapat berjalan dengan baik pada <i>browser Internet Explorer</i> versi 11
Result
Status

Tabel 5.9 Kasus Uji *Compability Testing* pada *Browser Edge* versi 16

Test Case Code	CT-004
Test Case Name	<i>Compability Testing</i> pada <i>Browser Edge</i> versi 16



Testing Objectives	Pengujian ini bertujuan mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser Browser Edge versi 16</i>
Test Case	Menguji <i>compability</i> sistem terhadap <i>Browser Edge versi 16</i>
Testing Step	<ol style="list-style-type: none"> 1. Penguji membuka aplikasi SortSite 2. Penguji memasukkan url sistem pada menu bar <i>address</i> 3. Penguji melakukan <i>checking</i> dengan menekan tombol <i>check</i> 4. Penguji melihat hasil <i>compability testing</i>
Expectation	Sistem dapat berjalan dengan baik pada <i>Browser Edge versi 16</i>
Result
Status

Tabel 5.10 Kasus Uji *Compability Testing* pada *Browser Firefox* versi 60

Test Case Code	CT-005
Test Case Name	<i>Compability Testing</i> pada <i>Browser Firefox</i> versi 60
Testing Objectives	Pengujian ini bertujuan mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser Browser Firefox</i> versi 60
Test Case	Menguji <i>compability</i> sistem terhadap <i>Browser Firefox</i> versi 60
Testing Step	<ol style="list-style-type: none"> 1. Penguji membuka aplikasi SortSite 2. Penguji memasukkan url sistem pada menu bar <i>address</i> 3. Penguji melakukan <i>checking</i> dengan menekan tombol <i>check</i> 4. Penguji melihat hasil <i>compability testing</i>
Expectation	Sistem dapat berjalan dengan baik pada <i>Browser Firefox</i> versi 60
Result
Status

Tabel 5.11 Kasus Uji *Compability Testing* pada *Browser Safari* versi ≤ 10

Test Case Code	CT-006
Test Case Name	<i>Compability Testing</i> pada <i>Browser Safari</i> versi ≤ 10
Testing Objectives	Pengujian ini bertujuan mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser Browser Safari</i> versi ≤ 10
Test Case	Menguji <i>compability</i> sistem terhadap <i>Browser Safari</i> versi ≤ 10
Testing Step	<ol style="list-style-type: none"> 1. Penguji membuka aplikasi SortSite 2. Penguji memasukkan url sistem pada menu bar <i>address</i> 3. Penguji melakukan <i>checking</i> dengan menekan tombol <i>check</i> 4. Penguji melihat hasil <i>compability testing</i>
Expectation	Sistem dapat berjalan dengan baik pada <i>Browser Safari</i> versi ≤ 10
Result
Status

Tabel 5.12 Kasus Uji *Compability Testing* pada *Browser Safari* versi 11

Test Case Code	CT-007
Test Case Name	<i>Compability Testing</i> pada <i>Browser Safari</i> versi ≤ 11
Testing Objectives	Pengujian ini bertujuan mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser Browser Safari</i> versi ≤ 11
Test Case	Menguji <i>compability</i> sistem terhadap <i>Browser Safari</i> versi ≤ 11
Testing Step	<ol style="list-style-type: none"> 1. Penguji membuka aplikasi SortSite 2. Penguji memasukkan url sistem pada menu bar <i>address</i> 3. Penguji melakukan <i>checking</i> dengan menekan tombol <i>check</i> 4. Penguji melihat hasil <i>compability testing</i>
Expectation	Sistem dapat berjalan dengan baik pada <i>Browser Safari</i> versi ≤ 11

Result
Status

Tabel 5.13 Kasus Uji *Compability Testing* pada *Browser Opera* versi 51

Test Case Code	CT-008
Test Case Name	<i>Compability Testing</i> pada <i>Browser Opera</i> versi 51
Testing Objectives	Pengujian ini bertujuan mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser Browser Opera</i> versi 51
Test Case	Menguji <i>compability</i> sistem terhadap <i>Browser Opera</i> versi 51
Testing Step	<ol style="list-style-type: none"> 1. Penguji membuka aplikasi SortSite 2. Penguji memasukkan url sistem pada menu bar <i>address</i> 3. Penguji melakukan <i>checking</i> dengan menekan tombol <i>check</i> 4. Penguji melihat hasil <i>compability testing</i>
Expectation	Sistem dapat berjalan dengan baik pada <i>Browser Opera</i> versi 51
Result
Status

Tabel 5.14 Kasus Uji *Compability Testing* pada *Browser Chrome* versi 66

Test Case Code	CT-009
Test Case Name	<i>Compability Testing</i> pada <i>Browser Chrome</i> versi 66
Testing Objectives	Pengujian ini bertujuan mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser Browser Chrome</i> versi 66
Test Case	Menguji <i>compability</i> sistem terhadap <i>Browser Chrome</i> versi 66
Testing Step	<ol style="list-style-type: none"> 1. Penguji membuka aplikasi SortSite 2. Penguji memasukkan url sistem pada menu bar <i>address</i> 3. Penguji melakukan <i>checking</i> dengan menekan tombol <i>check</i>

	4. Penguji melihat hasil <i>compability testing</i>
Expectation	Sistem dapat berjalan dengan baik pada <i>Browser Chrome</i> versi 66
Result
Status

Tabel 5.15 Kasus Uji *Compability Testing* pada *Browser iOS* versi ≤ 9

Test Case Code	CT-010
Test Case Name	<i>Compability Testing</i> pada <i>Browser iOS</i> versi ≤ 9
Testing Objectives	Pengujian ini bertujuan mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser Browser iOS</i> versi ≤ 9
Test Case	Menguji <i>compability</i> sistem terhadap <i>Browser iOS</i> versi ≤ 9
Testing Step	<ol style="list-style-type: none"> 1. Penguji membuka aplikasi SortSite 2. Penguji memasukkan url sistem pada menu bar <i>address</i> 3. Penguji melakukan <i>checking</i> dengan menekan tombol <i>check</i> 4. Penguji melihat hasil <i>compability testing</i>
Expectation	Sistem dapat berjalan dengan baik pada <i>Browser iOS</i> versi ≤ 9
Result
Status

Tabel 5.16 Kasus Uji *Compability Testing* pada *Browser iOS* versi 10

Test Case Code	CT-011
Test Case Name	<i>Compability Testing</i> pada <i>Browser iOS</i> versi 10
Testing Objectives	Pengujian ini bertujuan mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser Browser iOS</i> versi 10
Test Case	Menguji <i>compability</i> sistem terhadap <i>Browser iOS</i> versi 10
Testing Step	<ol style="list-style-type: none"> 1. Penguji membuka aplikasi SortSite 2. Penguji memasukkan url sistem pada menu bar <i>address</i>

	<ol style="list-style-type: none"> 3. Penguji melakukan <i>checking</i> dengan menekan tombol <i>check</i> 4. Penguji melihat hasil <i>compability testing</i>
Expectation	Sistem dapat berjalan dengan baik pada <i>Browser iOS</i> versi 10
Result
Status

Tabel 5.17 Kasus Uji *Compability Testing* pada *Browser iOS* versi 11

Test Case Code	CT-012
Test Case Name	<i>Compability Testing</i> pada <i>Browser iOS</i> versi 11
Testing Objectives	Pengujian ini bertujuan mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser Browser iOS</i> versi 11
Test Case	Menguji <i>compability</i> sistem terhadap <i>Browser iOS</i> versi 11
Testing Step	<ol style="list-style-type: none"> 1. Penguji membuka aplikasi SortSite 2. Penguji memasukkan url sistem pada menu bar <i>address</i> 3. Penguji melakukan <i>checking</i> dengan menekan tombol <i>check</i> 4. Penguji melihat hasil <i>compability testing</i>
Expectation	Sistem dapat berjalan dengan baik pada <i>Browser iOS</i> versi 11
Result
Status

Tabel 5.18 Kasus Uji *Compability Testing* pada *Browser Android* versi ≤ 3

Test Case Code	CT-013
Test Case Name	<i>Compability Testing</i> pada <i>Browser Android</i> versi ≤ 3
Testing Objectives	Pengujian ini bertujuan mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser Android</i> versi ≤ 3
Test Case	Menguji <i>compability</i> sistem terhadap <i>Browser Android</i> versi ≤ 3



Testing Step	<ol style="list-style-type: none"> 1. Penguji membuka aplikasi SortSite 2. Penguji memasukkan url sistem pada menu bar <i>address</i> 3. Penguji melakukan <i>checking</i> dengan menekan tombol <i>check</i> 4. Penguji melihat hasil <i>compability testing</i>
Expectation	Sistem dapat berjalan dengan baik pada <i>Browser Android</i> versi ≤ 3
Result
Status

Tabel 5.19 Kasus Uji *Compability Testing* pada *Browser Android* versi 4

Test Case Code	CT-014
Test Case Name	<i>Compability Testing</i> pada <i>Browser Android</i> versi 4
Testing Objectives	Pengujian ini bertujuan mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser Android</i> versi 4
Test Case	Menguji <i>compability</i> sistem terhadap <i>Browser Android</i> versi 4
Testing Step	<ol style="list-style-type: none"> 1. Penguji membuka aplikasi SortSite 2. Penguji memasukkan url sistem pada menu bar <i>address</i> 3. Penguji melakukan <i>checking</i> dengan menekan tombol <i>check</i> 4. Penguji melihat hasil <i>compability testing</i>
Expectation	Sistem dapat berjalan dengan baik pada <i>Browser Android</i> versi 4
Result
Status

Tabel 5.20 Kasus Uji *Compability Testing* pada *Browser BlackBerry* versi ≤ 7.1

Test Case Code	CT-015
Test Case Name	<i>Compability Testing</i> pada <i>Browser BlackBerry</i> versi ≤ 7.1



Testing Objectives	Pengujian ini bertujuan mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser BlackBerry</i> versi ≤ 7.1
Test Case	Menguji <i>compability</i> sistem terhadap <i>Browser BlackBerry</i> versi ≤ 7.1
Testing Step	<ol style="list-style-type: none"> 1. Penguji membuka aplikasi SortSite 2. Penguji memasukkan url sistem pada menu bar <i>address</i> 3. Penguji melakukan <i>checking</i> dengan menekan tombol <i>check</i> 4. Penguji melihat hasil <i>compability testing</i>
Expectation	Sistem dapat berjalan dengan baik pada <i>Browser BlackBerry</i> versi ≤ 7.1
Result
Status

Tabel 5.21 Kasus Uji *Compability Testing* pada *Browser BlackBerry* versi 10

Test Case Code	CT-016
Test Case Name	<i>Compability Testing</i> pada <i>Browser BlackBerry</i> versi 10
Testing Objectives	Pengujian ini bertujuan mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser BlackBerry</i> versi 10
Test Case	Menguji <i>compability</i> sistem terhadap <i>Browser BlackBerry</i> versi 10
Testing Step	<ol style="list-style-type: none"> 1. Penguji membuka aplikasi SortSite 2. Penguji memasukkan url sistem pada menu bar <i>address</i> 3. Penguji melakukan <i>checking</i> dengan menekan tombol <i>check</i> 4. Penguji melihat hasil <i>compability testing</i>
Expectation	Sistem dapat berjalan dengan baik pada <i>Browser BlackBerry</i> versi 10
Result
Status



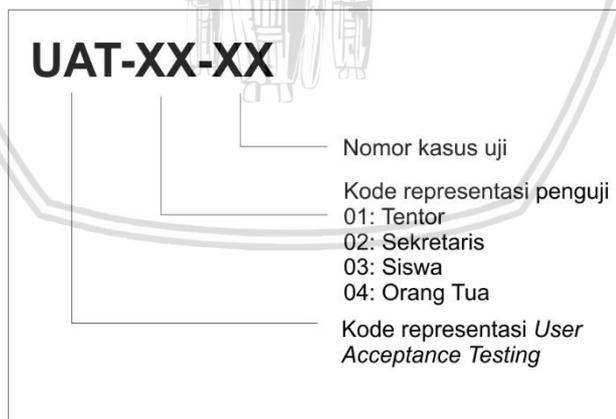
Adapun rancangan tampilan hasil pengujian *compability* dapat dilihat pada Tabel 5.22. Dimana pada kolom vertikal terdapat kategori penilaian dan kolom horizontal terdapat kolom jenis dan versi browser. Titik temu antara kategori dan jenis browser merupakan hasil dari *compability testing*

Tabel 5.22 Rancangan Tampilan Hasil Pengujian *Compability*

<i>Browser Version</i>	<i>IE</i> 9 10 11	<i>E</i> 15	<i>Firefox</i> 60	<i>Safari</i> <10 11	<i>Opera</i> 51	<i>Chrome</i> 66	<i>iOS</i> <=9 10 11	<i>Android</i> <=3 4	<i>BB</i> <=7 10
<i>Critical Issues</i>									
<i>Major Issues</i>									
<i>Minor issues</i>									

5.6.3 User Acceptance Testing

User Acceptance Testing (UAT) merupakan metode pengujian yang bertujuan untuk mengetahui tentang bagaimana penerimaan pengguna terhadap sistem yang dikembangkan. Selain itu pengujian ini juga untuk memastikan sistem dapat berjalan sesuai dengan kebutuhan pengguna. Dalam pengujian ini, akan dilakukan kepada tiap aktor dalam sistem ini. Aktor atau pengguna tersebut adalah seseorang yang telah mengikuti wawancara terkait pengembangan sistem ini yang selanjutnya akan disebut sebagai responden. Responden yang terlibat diantaranya dua tentor, satu sekretaris, satu siswa, dan satu orang tua. Setiap responden akan menguji sistem dengan mengikuti langkah-langkah yang sudah disediakan.



Gambar 5.15 Kodifikasi *UAT*

Gambar 5.15 menjelaskan tentang kodifikasi pada pengujian *UAT* yang nantinya akan digunakan sebagai *test case code* pada tiap tabel kasus uji. Tabel 5.18, Tabel 5.19, Tabel 5.20, dan Tabel 5.21 merupakan tabel kasus uji *UAT* yang digunakan dalam pengujian ini.

Table 5.14 Kasus Uji UAT Menambahkan Hasil Belajar

Test Case Code	UAT-01-01
Test Case Name	Menambahkan hasil belajar siswa
Use Case Code	UC-SIBBA-01
Testing Objectives	Pengujian ini bertujuan agar tentor dapat menambahkan hasil belajar siswa
Test Case	Mengisi seluruh field hasil belajar
Testing Step	<ol style="list-style-type: none"> 1. Penguji memilih menu masukkan nilai belajar 2. Penguji memilih menuh tambah 3. Penguji mengisi seluruh field dengan benar 4. Penguji menekan tombol simpan
Expectation	Sistem menambahkan hasil belajar
Result
Status

Table 5.15 Kasus Uji UAT Melihat Hasil Belajar Siswa

Test Case Code	UAT-03-02
Test Case Name	Melihat hasil belajar siswa
Use Case Code	UC-SIBBA-02
Testing Objectives	Pengujian ini bertujuan agar siswa dapat melihat hasil belajar
Test Case	Melihat hasil belajar
Testing Step	<ol style="list-style-type: none"> 1. Penguji memilih menu masukkan hasil belajar
Expectation	Sistem menampilkan hasil belajar siswa
Result
Status

Table 5.16 Kasus Uji UAT Melihat Presensi Tentor

Test Case Code	UAT-02-03
Test Case Name	Melihat presensi tentor
Use Case Code	UC-SIBBA-03

Testing Objectives	Pengujian ini bertujuan agar sekretaris dapat melihat presensi tentor
Test Case	Melihat presensi tentor
Testing Step	<ol style="list-style-type: none"> 1. Penguji memilih menu presensi tentor 2. Penguji memilih menu lihat detail
Expectation	Sistem menampilkan detail presensi tentor
Result
Status

Table 5.17 Kasus Uji UAT Melakukan Komplain

Test Case Code	UAT-04-04
Test Case Name	Melakukan komplain
Use Case Code	UC-SIBBA-04
Testing Objectives	Pengujian ini bertujuan agar orang tua atau siswa dapat melakukan komplain
Test Case	Melakukan komplain
Testing Step	<ol style="list-style-type: none"> 1. Penguji memilih menu komplain 2. Penguji mengisi <i>form</i> dengan lengkap dan benar 3. Penguji memilih fungsi tombol kirim
Expectation	Sistem mengirim komplain
Result
Status

Kemudian akan dibuat perancangan *UAT* terkait pertanyaan-pertanyaan yang dijelaskan pada tabel-tabel berikut. Tiap pertanyaan memiliki kriteria yang berbeda. Terdapat 4 kriteria yaitu *performance*, bagaimana sistem reaksi sistem saat dijalankan, *usability*, bagaimana kemudahan pengguna dalam mengoperasikan sistem, *compatibility and interoperability*, bagaimana sistem saat dijalankan pada peramban tertentu, dan *backup and recovery*, apakah sistem dapat membantu permasalahan pengguna.

Table 5.18 Pertanyaan Kasus Uji Menambahkan Hasil Belajar oleh Tendor

No	Kriteria	Daftar Pertanyaan
1	<i>Performance</i>	Apakah fitur menambahkan hasil belajar dapat diakses dengan mudah dan ringan?



No	Kriteria	Daftar Pertanyaan
2	<i>Usability</i>	Apakah fitur menambahkan hasil belajar mudah untuk dioperasikan?
		Apakah form menambahkan hasil belajar mudah dipahami?
3	<i>Compatibility and Interoperability</i>	Apakah fitur menambahkan hasil belajar dapat berjalan dengan baik pada perangkat dan browser yang digunakan?
4	<i>Backup and Recovery</i>	Apakah Sistem Informasi LBB Al-Hasyimi membantu tentor dalam mengelola hasil belajar siswa?

Table 5.19 Pertanyaan Kasus Uji Melihat Hasil Belajar oleh Siswa dan Orang Tua

No	Kriteria	Daftar Pertanyaan
1	<i>Performance</i>	Apakah fitur melihat hasil belajar dapat diakses dengan mudah dan ringan?
2	<i>Usability</i>	Apakah fitur melihat hasil belajar mudah untuk dipahami?
3	<i>Compatibility and Interoperability</i>	Apakah fitur melihat hasil belajar siswa dapat berjalan dengan baik pada perangkat dan browser yang digunakan?
4	<i>Backup and Recovery</i>	Apakah Sistem Informasi LBB Al-Hasyimi membantu dalam siswa dan orang tua dalam menerima dan melihat hasil belajar siswa?

Table 5.20 Pertanyaan Kasus Uji Melihat Presensi Tendor oleh Sekertaris

No	Kriteria	Daftar Pertanyaan
1	<i>Performance</i>	Apakah fitur melihat presensi tentor dapat diakses dengan mudah dan ringan?
2	<i>Usability</i>	Apakah fitur melihat presensi tentor mudah untuk dipahami?
3	<i>Compatibility and Interoperability</i>	Apakah fitur melihat presensi tentor dapat berjalan dengan baik pada perangkat dan browser yang digunakan?
4	<i>Backup and Recovery</i>	Apakah Sistem Informasi LBB Al-Hasyimi membantu dalam sekretaris dalam memonitoring kegiatan bimbingan belajar oleh tentor?

Table 5.21 Pertanyaan Kasus Uji Melakukan komplain oleh Siswa dan Orang Tua

No	Kriteria	Daftar Pertanyaan
1	<i>Performance</i>	Apakah fitur melakukan komplain dapat diakses dengan mudah dan ringan?
2	<i>Usability</i>	Apakah fitur mengirim komplain mudah untuk dioperasikan?
		Apakah form komplain mudah untuk dipahami?
3	<i>Compatibility and Interoperability</i>	Apakah fitur mengirim komplain dapat berjalan dengan baik pada perangkat dan browser yang digunakan?
4	<i>Backup and Recovery</i>	Apakah Sistem Informasi LBB Al-Hasyimi membantu dalam siswa dan orang tua dalam melakukan komplain?

Pada Tabel 5.24 adalah rancangan pertanyaan yang akan diberikan kepada tiap responden dalam pengujian UAT ini. Pertanyaan ini dibuat berdasarkan perancangan pertanyaan kasus uji. Terdapat kriteria jawaban yang harus dipilih salah satu oleh responden diantaranya SS (Sangat Setuju), S (Setuju), N (Netral), TS (Tidak Setuju), dan STS (Sangat Tidak Setuju). Terakhir adalah baris total untuk menjumlahkan hasil tiap kategori penilaian oleh responden.

Table 5.22 Rancangan Pertanyaan Responden

Pertanyaan	Jawaban				
	SS	S	N	TS	STS
Total					

Selanjutnya akan dilakukan perhitungan hasil pengujian menggunakan skala *linkert*. Proses perhitungan pada skala *likert* diawali dengan memberikan bobot nilai pada setiap jawaban. Tabel 5.25 berisi rincian bobot nilai jawaban, sedangkan tabel 5.26 berisi persentase untuk setiap jawaban.

Table 5.23 Keterangan Bobot Nilai

A=Sangat setuju/sangat baik	5
B=Setuju/baik	4
C=Netral	3
D=Kurang/tidak setuju	2
E=Sangat Kurang/sangat tidak setuju	1

Table 5.24 Keterangan Presentase 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

$$\text{Total Nilai} = (n \times 5) + (n \times 4) + (n \times 3) + (n \times 2) + (n \times 1)$$

Variabel n merupakan jumlah responden yang menjawab. Selanjutnya untuk melakukan perhitungan harus diketahui terlebih dahulu nilai terendah (Y) dan nilai tertinggi (X). Rumus yang digunakan bisa 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.

$$Y = N1 \times n \times U$$

$$X = N2 \times n \times U$$

Sedangkan untuk menghitung hasil dari *UAT* dengan skala *likert* terdapat pada rumus 2.4.

$$\text{Rumus Index \%} = (\text{Total Nilai} / Y) \times 100$$

BAB 6 IMPLEMENTASI

Pada bab ini akan menjelaskan implementasi sistem yang dibuat berdasarkan dengan perancangan sistem pada bab sebelumnya. Implementasi menggunakan *framework CodeIgniter* dengan kolaborasi *bootstrap*. Selanjutnya akan dijelaskan beberapa kode program dan beberapa antarmuka sistem.

6.1 Implementasi Algoritma

Pada subbab ini akan menjelaskan tentang implementasi dari algoritma yang telah dibuat pada bab sebelumnya. Selanjutnya akan ditampilkan beberapa kode program dari fungsi sistem diantaranya, menambahkan hasil belajar siswa, melihat hasil belajar siswa, melihat presensi tentor, dan melakukan komplain.

6.1.1 Menambahkan Hasil Belajar Siswa

Pada Tabel 6.1 merupakan baris kode program dari implementasi fungsi menambahkan hasil belajar siswa yang dapat dilakukan oleh tentor. Implementasi ini dibuat berdasarkan dari perancangan algoritma *pseudocode* pada Tabel 5.2.

Table 6.1 Implementasi Menambahkan Hasil Belajar

No	Kode Program
1	<code>public function tambah()</code>
2	<code>{</code>
3	<code> \$this->data['main_view'] = 'tentor/nilai_form';</code>
4	<code> \$this->data['form_action'] =</code>
5	<code> site_url('tentor/nilai/tambah');</code>
6	<code> if (! \$_POST) {</code>
7	<code> \$nilai = (object) \$this->hasil_belajar-></code>
8	<code> default_value;</code>
9	<code> } else {</code>
10	<code> \$nilai = \$this->input->post(null, true);</code>
11	<code> }</code>
12	<code> if (! \$this->hasil_belajar->validate('form_rules')) {</code>
13	<code> \$this->data['values'] = (object) \$nilai;</code>
14	<code> \$this->load->view(\$this->layout, \$this->data);</code>
15	<code> return;</code>
16	<code> }</code>
17	<code> if (\$this->hasil_belajar->tambah(\$nilai)) {</code>
18	<code> \$this->session->set_flashdata('pesan', 'nilai</code>
19	<code> berhasil disimpan. Kembali ke halaman ' .</code>
20	<code> anchor('tentor/nilai', 'hasil_belajar.',</code>
	<code> 'class="alert- link");</code>
	<code> redirect('tentor/nilai/sukses');</code>

Tabel 6.2 Implementasi Melihat hasil belajar (lanjutan)

21	} else {
22	\$this->session->set_flashdata('pesan_error', 'nilai gagal disimpan. Kembali ke halaman ' .
23	anchor('tentor/nilai', 'nilai.', 'class="alert-link"')));
24	redirect('tentor/nilai/error');
25	}
26	}
27	}

6.1.2 Melihat Hasil Belajar Siswa

Pada Tabel 6.2 merupakan baris kode program dari implementasi fungsi melihat hasil belajar siswa yang dapat dilakukan oleh siswa dan orang tua. Implementasi ini dibuat berdasarkan dari perancangan algoritma *pseudocode* pada Tabel 5.3.

Table 6.2 Implementasi Melihat Hasil Belajar

No	Kode Program
1	public function index(\$offset = null)
2	{
3	\$no_siswa = \$this->session->userdata('no_peserta');
4	\$hasil_belajar = \$this->db->query("select * from
5	tb_hasil_belajar h join tb_peserta p on h.id_siswa =
6	p.id where p.id = \$no_siswa")->result();
7	if (empty(\$hasil_belajar)) {
8	\$this->data['hasil_belajar'] = 'Tidak ada data
9	hasil_belajar.';
10	} else {
11	\$this->data['hasil_belajar'] = \$hasil_belajar;
12	\$this->data['paging'] = \$this->hasil_belajar-
13	>paging('biasa',
14	site_url('dashboard/hasil_belajar/halaman/'), 4);
15	}
16	\$this->data['form_action'] =
17	site_url('dashboard/hasil_belajar/cari');
18	\$this->load->view(\$this->layout, \$this->data);
19	}

6.1.3 Melihat Presensi Tentor

Pada Tabel 6.3 merupakan baris kode program dari implementasi fungsi melihat presensi tentor yang dapat dilakukan oleh sekretaris. Implementasi ini dibuat berdasarkan dari perancangan algoritma *pseudocode* pada Tabel 5.4.

Table 6.3 Implementasi Melihat Presensi Tentor

No	Kode Program
1	public function index(\$offset = 0)
2	{
3	\$presensi = \$this->db->query("select
4	count(id_hasil_belajar) as pertemuan, tentor from
5	tb_hasil_belajar group by id_tentor")->result();
6	if (empty(\$presensi)) {
7	\$this->data['presensi'] = 'Tidak ada data
8	presensi.';
9	} else {
10	\$this->data['presensi'] = \$presensi;
11	\$this->data['paging'] = \$this->presensi-
12	>paging('biasa',
13	site_url('admin/presensi/halaman/'), 4);
14	}
15	\$this->load->view(\$this->layout, \$this->data);
16	}

6.1.4 Melakukan Komplain

Pada Tabel 6.4 merupakan baris kode program dari implementasi fungsi melakukan komplain yang dapat dilakukan oleh orang tua dan / atau siswa. Implementasi ini dibuat berdasarkan dari perancangan algoritma *pseudocode* pada Tabel 5.5.

Table 6.4 Implementasi Melakukan Komplain

No	Kode Program
1	public function index()
2	{
3	\$kontak = (object) \$this->input->post(null, true);
4	if (! \$_POST) {

```

5      $this->data['values'] = (object) $this->komplain-
6      >default_values;
7      } else {
8          $this->data['values'] = $komplain;
9      }
10     if (! $this->komplain->validate('form_rules')) {
11         $this->data['captcha'] = $this->komplain-
12         >buat_captcha();
13         $this->data['form_action'] = site_url('komplain');
14         $this->load->view($this->layout, $this->data);
15         return;
16     }
17     if($this->komplain->simpan($komplain)) {
18         $this->session->set_flashdata('pesan', 'Pesan
19         berhasil dikirim. Balasan akan dikirim ke email
20         Anda.'.anchor(base_url(), ' Kembali ke halaman
21         home.', 'class="alert-link"'));
22         redirect(komplain-sukses');
23     } else {
24         $this->session->set_flashdata('pesan_error', 'Maaf,
25         pengiriman pesan gagal. Kami mengalami gangguan,
26         coba ' . anchor('kontak', 'ulangi', 'class="alert-
27         link"') .' beberapa saat lagi.');
```

6.2 Implementasi *User Interface*

Pada subbab ini akan menjelaskan tentang implementasi dari *user interface* yang telah dibuat pada bab sebelumnya. Selanjutnya akan ditampilkan *user interface* dari fungsi sistem diantaranya, menambahkan hasil belajar siswa, melihat hasil belajar siswa, melihat presensi tutor, dan melakukan komplain.

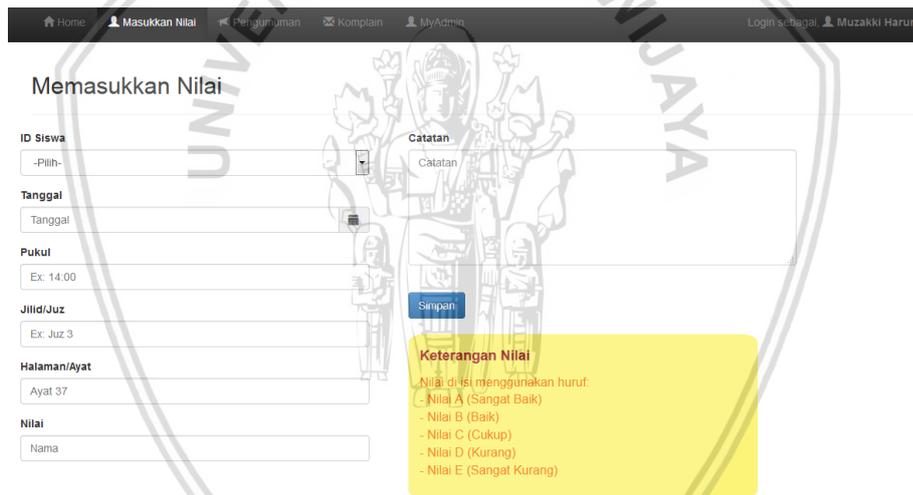
6.2.1 Menambahkan Hasil Belajar

Pada Gambar 6.1 menjelaskan kondisi awal halaman hasil belajar siswa. Pada gambar tersebut masih dalam keadaan kosong, yang artinya masih belum terdapat hasil belajar yang masuk. Untuk menambahkan hasil belajar baru, maka tutor harus memilih tombol tambah, kemudian akan dialihkan ke halaman *form* hasil belajar yang harus diisi oleh tutor.



Gambar 6.1 Menambahkan Hasil Belajar

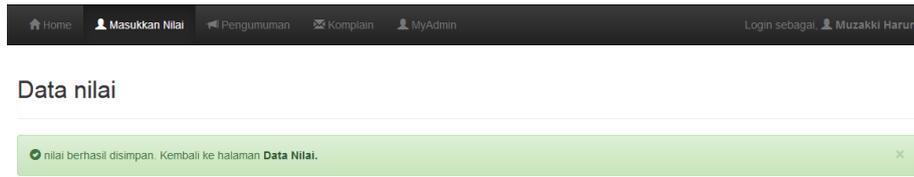
Pada Gambar 6.2 adalah tampilan *form* hasil belajar yang harus diisi oleh tentor. Halaman ini dibuat berdasarkan perancangan *user interface* pada Gambar 5.10. Pada halaman ini terdapat *form* yang terdiri dari 7 field yang masing-masing field harus diisi oleh tentor dengan benar dan tidak boleh kosong. Field pertama merupakan list siswa yang ingin diberi nilai. Selanjutnya field kedua dan ketiga merupakan tanggal dan waktu dilaksanakannya bimbingan. Field keempat dan kelima merupakan jilid/juz dan halaman/ayat berapa terakhir siswa saat bimbingan. Field keenam merupakan nilai siswa yang diisi berdasarkan aturan yang sudah dijelaskan pada keterangan nilai. Terakhir adalah field ketujuh yang berisi tentang catatan khusus untuk siswa.



Gambar 6.2 Menambahkan Hasil Belajar

Gambar 6.3 adalah notifikasi data berhasil ditambahkan. Notifikasi ini akan muncul ketika tentor sudah menambahkan data dengan lengkap dan benar. Kemudian memilih fungsi tombol simpan, maka data akan tersimpan dan berhasil ditambahkan.





Gambar 6.3 Menambahkan Hasil Belajar

6.2.2 Melihat Hasil Belajar

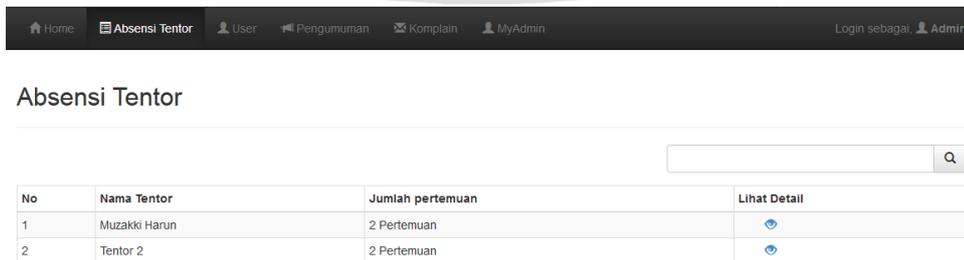
Pada Gambar 6.4 adalah *user interface* hasil belajar siswa. Halaman ini dibuat berdasarkan perancangan *user interface* pada Gambar 5.11. Halaman ini dapat dilihat oleh siswa dan orang tua siswa. Terdapat fitur paging dan pencarian untuk memudahkan pengguna dalam melihat hasil belajar. Kolom yang terdapat pada halaman ini sama dengan apa yang dimasukkan oleh tentor pada *form* hasil belajar siswa.



Gambar 6.4 Hasil Belajar Siswa

6.2.3 Melihat Presensi Tmentor

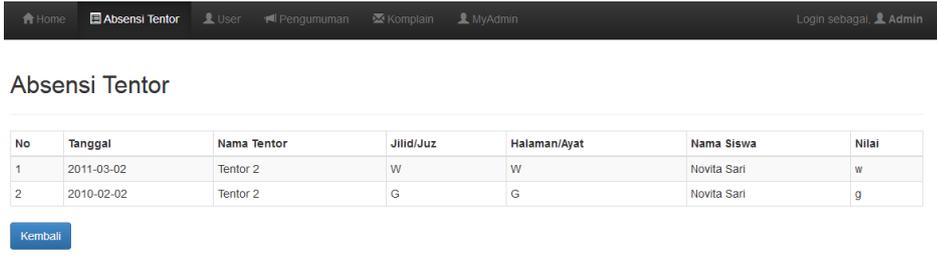
Pada Gambar 6.5 adalah *user interface* presensi tmentor. Halaman ini dibuat berdasarkan perancangan *user interface* pada Gambar 5.12. Halaman ini hanya dapat diakses oleh seorang sekretaris / admin. Halaman ini hanya akan menampilkan berapa kali bimbingan yang telah dilakukan tiap tmentor. Untuk melihat lebih lengkap untuk setiap tmentor, maka dapat memilih fungsi lihat detail



Gambar 6.5 Presensi Tmentor

Pada Gambar 6.6 adalah *user interface* detail presensi tmentor. Halaman ini dibuat berdasarkan perancangan *user interface* pada Gambar 5.13. Halaman ini akan menampilkan presensi tiap tmentor secara detail. Dengan adanya halaman ini, maka admin / sekretaris dapat memonitoring keaktifan tiap tmentor.

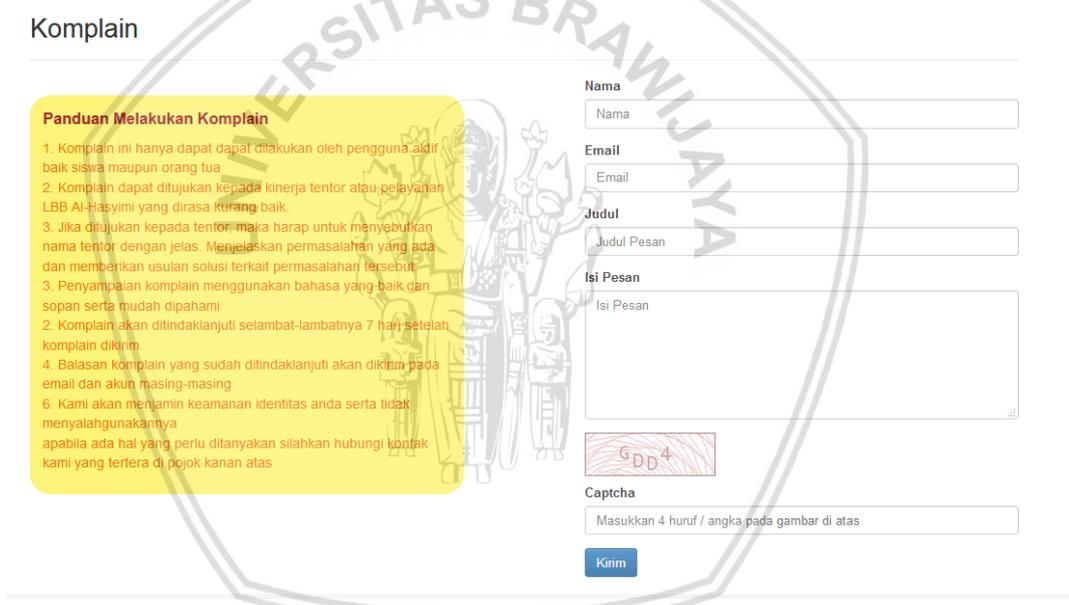




Gambar 6.6 Detail Presensi Tentor

6.2.4 Form Komplain

Pada Gambar 6.5 adalah *user interface form* komplain untuk siswa dan orang tua. Halaman ini dibuat berdasarkan perancangan *user interface* pada Gambar 5.14. Pada halaman ini terdapat dua bagian, sisi kiri merupakan panduan untuk melakukan komplain yang harus diikuti dan sisi kanan merupakan *form* komplain yang terdiri dari 5 field yang harus diisi dengan benar dan lengkap.



Gambar 6.6 Form Komplain

BAB 7 PENGUJIAN

Pada bab ini akan menjelaskan tentang hasil pengujian sistem. Pengujian dilakukan untuk memastikan sistem dapat berjalan dengan baik dan sesuai dengan kebutuhan *stakeholder*. Dalam melakukan pengujian sistem ini akan menggunakan tiga metode pengujian, yaitu *Validation Testing*, *Utility Testing*, dan *User Acceptance Testing*. *Validation Testing* untuk menguji apakah fungsional sistem sudah berjalan sesuai ekspektasi. *Utility Testing* untuk menguji seberapa besar dampak yang diberikan sistem dalam mengefisienkan proses bisnis yang ada. *User Acceptance Testing* untuk menguji apakah sistem dapat diterima oleh *stakeholder*.

7.1 Validation Testing

Validation Testing untuk menguji apakah fungsional sistem sudah berjalan sesuai ekspektasi dari pengguna. Pengujian ini dilakukan berdasarkan dengan perancangan *validation testing* pada bab perancangan. Hasil dari *validation testing* akan ditampilkan pada Tabel 7.1 yang terdiri dari *validation testing code*, *testing objectives*, *result*, dan *status*.

Table 7.1 Hasil *Validation Testing*

No	Validation Testing Code	Testing Objectives	Result	Status
1	VT-001	Pengujian ini bertujuan agar tentor dapat memasukkan hasil belajar siswa.	Sistem berhasil menambahkan hasil belajar.	Valid
2	VT-002	Pengujian ini bertujuan agar tentor tidak dapat memasukkan nilai jika field tidak diisi dengan lengkap.	Sistem berhasil memberikan peringatan pada field yang kosong untuk diisi.	Valid
3	VT-003	Pengujian ini bertujuan agar siswa dapat melihat hasil belajar.	Sistem berhasil menampilkan hasil belajar siswa.	Valid
4	VT-004	Pengujian ini bertujuan agar sekretaris dapat melihat presensi tentor.	Sistem berhasil menampilkan detail presensi tentor.	Valid
5	VT-005	Pengujian ini bertujuan agar orang tua atau siswa dapat melakukan komplain.	Sistem berhasil mengirim komplain.	Valid

Table 7.2 Hasil Validation Testing (lanjutan)

6	VT-006	Pengujian ini bertujuan agar orang tua atau siswa tidak dapat mengirim komplain jika form tidak diisi dengan lengkap dan benar.	Sistem berhasil memberikan pemberitahuan form harus diisi.	Valid
---	--------	---	--	-------

Hasil pengujian sistem menggunakan metode *validation testing* yang dilakukan pada beberapa fitur utama sistem akan dihitung menggunakan rumus sebagai berikut.

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

$$\text{Hasil} = (6 / 6) \times 100\%$$

$$\text{Hasil} = 100\%$$

Dengan demikian dapat disimpulkan bahwa beberapa fungsi utama yang diuji menggunakan *validation testing* dinyatakan *valid*. Fungsi-fungsi tersebut diantaranya adalah menambahkan hasil belajar siswa, melihat hasil belajar siswa, melihat presensi tentor, dan melakukan komplain. Jadi, pengguna sudah dapat dipastikan akan dapat menjalankan fungsi-fungsi tersebut dengan baik.

7.2 Compatibility Testing

Pada bab ini akan menjelaskan hasil dari pengujian *compatibility testing* dari sistem ini. Pengujian ini bertujuan untuk mengetahui apakah sistem yang dibuat ini dapat diakses pada beberapa jenis browser yang berbeda dengan versi yang terbaru saat ini. Penilaian untuk tiap tingkat kategori masalah dilakukan berdasarkan perancangan *compatibility testing* yang telah dibuat pada bab perancangan.

Browser Version	IE			Edge	Firefox	Safari		Opera	Chrome	iOS			Android		BlackBerry	
	9	10	11	16	60	≤ 10	11	51	66	≤ 9	10	11	≤ 3	4*	≤ 7.1	10.0
Critical Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Major Issues	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Minor Issues	⚠	⚠	⚠	✓	⚠	⚠	✓	✓	✓	⚠	✓	✓	✓	✓	⚠	✓

Gambar 7.1 Hasil Compatibility Testing

Hasil dari pengujian ini dapat dilihat pada Gambar 7.1 dibawah ini. Hasil pengujian menunjukkan bahwa terdapat permasalahan pada beberapa *browser*, yaitu Internet Explorer, Firefox, Safari, iOS, dan BlackBerry. Untuk lebih jelasnya akan dijelaskan pada Tabel berikut.

Table 7.3 Hasil *Compatibility Testing*

No	Compatibility Testing Code	Testing Objectives	Result	Status
1	CT-001	Pengujian ini bertujuan untuk mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser Internet Explorer</i> versi 9	Terdapat <i>major issue</i> dan <i>minor issue</i>	Tidak Valid
2	CT-002	Pengujian ini bertujuan untuk mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser Internet Explorer</i> versi 10	Terdapat <i>major issue</i> dan <i>minor issue</i>	Tidak Valid
3	CT-003	Pengujian ini bertujuan untuk mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser Internet Explorer</i> versi 11	Terdapat <i>major issue</i> dan <i>minor issue</i>	Tidak Valid
4	CT-004	Pengujian ini bertujuan untuk mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser Internet Edge 16</i>	Berjalan dengan baik atau tidak ada masalah	Valid
5	CT-005	Pengujian ini bertujuan untuk mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser Firefox</i> versi 60	Terdapat <i>minor issue</i>	Tidak Valid

6	CT-006	Pengujian ini bertujuan untuk mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser Safari versi >= 10</i>	Terdapat <i>minor issue</i>	<i>Tidak Valid</i>
7	CT-007	Pengujian ini bertujuan untuk mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser Safari versi 11</i>	Berjalan dengan baik atau tidak ada masalah	<i>Valid</i>
8	CT-008	Pengujian ini bertujuan untuk mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser Opera versi 51</i>	Berjalan dengan baik atau tidak ada masalah	<i>Valid</i>
9	CT-009	Pengujian ini bertujuan untuk mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser Chrome versi 66</i>	Berjalan dengan baik atau tidak ada masalah	<i>Valid</i>
10	CT-010	Pengujian ini bertujuan untuk mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser IOS versi <=9</i>	Terdapat <i>minor issue</i>	<i>Tidak Valid</i>
11	CT-011	Pengujian ini bertujuan untuk mengetahui apakah komponen – komponen yang diujikan dapat berjalan	Berjalan dengan baik atau tidak ada masalah	<i>Valid</i>

		dengan baik pada <i>browser IOS versi 10</i>		
12	CT-012	Pengujian ini bertujuan untuk mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser IOS versi 11</i>	Berjalan dengan baik atau tidak ada masalah	<i>Valid</i>
13	CT-013	Pengujian ini bertujuan untuk mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser Android versi <=3</i>	Berjalan dengan baik atau tidak ada masalah	<i>Valid</i>
14	CT-014	Pengujian ini bertujuan untuk mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser Android versi 4</i>	Berjalan dengan baik atau tidak ada masalah	<i>Valid</i>
15	CT-015	Pengujian ini bertujuan untuk mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser BlacBerry <= 7.1</i>	Terdapat <i>minor issue</i>	<i>Tidak Valid</i>
16	CT-016	Pengujian ini bertujuan untuk mengetahui apakah komponen – komponen yang diujikan dapat berjalan dengan baik pada <i>browser BlackBerry versi 10</i>	Berjalan dengan baik atau tidak ada masalah	<i>Valid</i>

7.3 User Acceptance Testing

Pada pengujian *User Acceptance Testing (UAT)* digunakan untuk mengetahui apakah sistem yang telah dibuat dapat diterima dengan baik oleh stakeholder atau tidak. Pengujian ini dilakukan dengan memberikan kuisisioner kepada sekretaris, tentor, siswa, dan orang tua sesuai dengan perancangan pengujian yang telah dibuat pada bab perancangan. Pengujian ini bertujuan untuk memastikan apakah sistem yang dikembangkan ini sudah dapat menyelesaikan permasalahan pada pengguna, sehingga sistem dapat diterima dengan baik oleh pengguna. Pada Tabel 7.2 merupakan rangkuman dari hasil *UAT*.

7.3.1 Analisis Hasil *UAT* Menambahkan Hasil Belajar

Berikut ini adalah hasil pengisian kuisisioner pengujian *UAT* dengan kasus uji menambahkan hasil belajar. Responden dalam pengujian kasus uji ini terdapat 2 tentor. Pada Tabel 7.4 merupakan hasil kuisisioner *UAT* menambahkan hasil belajar.

Table 7.4 Hasil Kuisisioner Kasus Uji Menambahkan Hasil Belajar oleh Tmentor

Pertanyaan	Jawaban				
	SS	S	N	TS	STS
Apakah fitur menambahkan hasil belajar dapat diakses dengan mudah dan ringan?		1	1		
Apakah fitur menambahkan hasil belajar mudah untuk dioperasikan?	1	1			
Apakah form menambahkan hasil belajar mudah dipahami?	1	1			
Apakah fitur menambahkan hasil belajar dapat berjalan dengan baik pada perangkat dan browser yang digunakan?			2		
Apakah Sistem Informasi LBB Al-Hasyimi membantu tentor dalam mengelola hasil belajar siswa?	1		1		
Total	3	3	4	0	0

Hasil dari *UAT* tersebut selanjutnya dianalisis menggunakan skala *likert*. Teori dan rumus perhitungan skala *likert* diimplementasikan sesuai dengan perancangan pada pengujian *UAT* ini. Berikut merupakan uraian poin-poin perhitungan hasil *UAT* menggunakan skala *likert*:

1. *UAT* Melihat Presensi Tmentor
 - a. Total nilai dari jawaban 2 responden (tentor):

$$A \text{ (Sangat Setuju)} = 3 \times 5 = 15$$

$$B \text{ (Setuju)} = 3 \times 4 = 12$$

$$C \text{ (Netral)} = 4 \times 3 = 12$$

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

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

$$\text{Total Nilai} = A + B + C + D + E$$

$$\text{Total Nilai} = 15 + 12 + 12 + 0 + 0 = 39$$

b. Nilai Y

$$Y = \text{Jumlah skala} \times \text{Jumlah responden} \times \text{Jumlah pertanyaan}$$

$$Y = 5 \times 2 \times 5 = 50$$

c. Nilai Presentase

$$\text{Presentase} = (\text{Total Nilai} / Y) \times 100\%$$

$$\text{Presentase} = (39 / 50) \times 100\% = 94\% \text{ (Sangat Setuju)}$$

7.3.2 Analisis Hasil *UAT* Melihat Hasil Belajar

Berikut ini adalah hasil pengisian kuisisioner pengujian *UAT* dengan kasus uji melihat hasil belajar. Responden dalam pengujian kasus uji ini terdapat 1 siswa dan 1 orang tua. Pada Tabel 7.5 merupakan hasil kuisisioner *UAT* melihat hasil belajar.

Table 7.5 Hasil Kuisisioner Kasus Uji Menambahkan Hasil Belajar oleh Siswa dan Orang Tua

Pertanyaan	Jawaban				
	SS	S	N	TS	STS
Apakah fitur melihat hasil belajar dapat diakses dengan mudah dan ringan?	2				
Apakah fitur melihat hasil belajar mudah untuk dipahami?	1	1			
Apakah fitur melihat hasil belajar siswa dapat berjalan dengan baik pada perangkat dan browser yang digunakan?	2				
Apakah Sistem Informasi LBB Al-Hasyimi membantu dalam siswa dan orang tua dalam menerima dan melihat hasil belajar siswa?	2				
Total	7	1	0	0	0

Hasil dari *UAT* tersebut selanjutnya dianalisis menggunakan skala *likert*. Teori dan rumus perhitungan skala *likert* diimplementasikan sesuai dengan perancangan pada pengujian *UAT* ini. Berikut merupakan uraian poin-poin perhitungan hasil *UAT* menggunakan skala *likert*:

1. *UAT* Melihat Presensi Tentor

- a. Total nilai dari jawaban 2 responden (siswa dan orang tua):

A (Sangat Setuju) = 7 x 5 = 35

B (Setuju) = 1 x 4 = 4

C (Netral) = 0 x 3 = 0

D (Tidak Setuju) = 0 x 2 = 0

E (Sangat Tidak Setuju) = 0 x 1 = 0

Total Nilai = A + B + C + D + E

Total Nilai = 35 + 1 + 0 + 0 + 0 = 36

- b. Nilai Y

Y = Jumlah skala x Jumlah responden x Jumlah pertanyaan

Y = 5 x 2 x 4 = 40

- c. Nilai Presentase

Presentase = (Total Nilai / Y) x 100%

Presentase = (36 / 40) x 100% = 90% (Sangat Setuju)

7.3.3 Analisis Hasil *UAT* Melihat Presensi tentor

Berikut ini adalah hasil pengisian kusioner pengujian *UAT* dengan kasus uji melihat presensi tentor. Responden dalam pengujian kasus uji ini terdapat hanya dilakukan oleh sekertaris. Pada Tabel 7.7 merupakan hasil kuisisioner *UAT* melihat presensi tentor.

Table 7.6 Hasil Kuisisioner Kasus Uji Melihat Presensi Tentor oleh Sekertaris

Pertanyaan	Jawaban				
	SS	S	N	TS	STS
Apakah fitur melihat presensi tentor dapat diakses dengan mudah dan ringan?	1				
Apakah fitur melihat presensi tentor mudah untuk dipahami?		1			
Apakah fitur melihat presensi tentor dapat berjalan dengan baik pada	1				



Pertanyaan	Jawaban				
	SS	S	N	TS	STS
perangkat dan browser yang digunakan?					
Apakah Sistem Informasi LBB Al-Hasyimi membantu dalam sekertaris dalam memonitoring kegiatan bimbingan belajar oleh tentor?	1				
Total	3	1	0	0	0

Hasil dari *UAT* tersebut selanjutnya dianalisis menggunakan skala *likert*. Teori dan rumus perhitungan skala *likert* diimplementasikan sesuai dengan perancangan pada pengujian *UAT* ini. Berikut merupakan uraian poin-poin perhitungan hasil *UAT* menggunakan skala *likert*:

1. *UAT* Melihat Presensi Tmentor

a. Total nilai dari jawaban 1 responden (sekertaris):

$$A (\text{Sangat Setuju}) = 3 \times 5 = 15$$

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

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

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

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

$$\text{Total Nilai} = A + B + C + D + E$$

$$\text{Total Nilai} = 15 + 4 + 0 + 0 + 0 = 19$$

b. Nilai Y

$$Y = \text{Jumlah skala} \times \text{Jumlah responden} \times \text{Jumlah pertanyaan}$$

$$Y = 5 \times 1 \times 4 = 20$$

c. Nilai Presentase

$$\text{Presentase} = (\text{Total Nilai} / Y) \times 100\%$$

$$\text{Presentase} = (19 / 20) \times 100\% = 95\% (\text{Sangat Setuju})$$

7.3.4 Analisis Hasil *UAT* Melakukan *Komplain*

Berikut ini adalah hasil pengisian kusioner pengujian *UAT* dengan kasus uji melakukan *komplain*. Responden dalam pengujian kasus uji ini terdapat 1 siswa dan 1 orang tua. Pada Tabel 7.5 merupakan hasil kuisisioner *UAT* menambahkan hasil belajar.

Table 7.7 Hasil Kuisisioner Kasus Uji Melakukan Komplain oleh Siswa dan Orang Tua

Pertanyaan	Jawaban				
	SS	S	N	TS	STS
Apakah fitur melakukan komplain dapat diakses dengan mudah dan ringan?	2				
Apakah fitur mengirim komplain mudah untuk dioperasikan?	1	1			
Apakah form komplain mudah untuk dipahami?	1	1			
Apakah fitur mengirim komplain dapat berjalan dengan baik pada perangkat dan browser yang digunakan?	2				
Apakah Sistem Informasi LBB Al-Hasyimi membantu dalam siswa dan orang tua dalam melakukan komplain?	2				
Total	8	2	0	0	0

Hasil dari *UAT* tersebut selanjutnya dianalisis menggunakan skala *likert*. Teori dan rumus perhitungan skala *likert* diimplementasikan sesuai dengan perancangan pada pengujian *UAT* ini. Berikut merupakan uraian poin-poin perhitungan hasil *UAT* menggunakan skala *likert*:

1. *UAT* Melihat Presensi Tentor

- a. Total nilai dari jawaban 2 responden (tentor):

$$A (\text{Sangat Setuju}) = 8 \times 5 = 40$$

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

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

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

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

$$\text{Total Nilai} = A + B + C + D + E$$

$$\text{Total Nilai} = 40 + 8 + 0 + 0 + 0 = 48$$

- b. Nilai Y

$$Y = \text{Jumlah skala} \times \text{Jumlah responden} \times \text{Jumlah pertanyaan}$$

$$Y = 5 \times 2 \times 5 = 50$$

- c. Nilai Presentase

$$\text{Presentase} = (\text{Total Nilai} / Y) \times 100\%$$

Presentase = $(48 / 50) \times 100\% = 96\%$ (Sangat Setuju)

Dari hasil analisis menggunakan skala *linkert* diatas maka dapat diambil kesimpulan bahwasanya untuk pengujian UAT menambahkan hasil belajar oleh tentor memperoleh presentase nilai sebesar 94% yang artinya tentor sangat setuju dengan fitur tersebut. Kemudian untuk pengujian UAT melihat hasil belajar yang oleh siswa maupun orang tua memperoleh presentase nilai sebesar 90% yang artinya siswa maupun orang tua tersebut sangat setuju dengan fitur tersebut. Selanjutnya untuk pengujian UAT melihat presensi tentor oleh sekertaris memperoleh presentase nilai sebesar 95% yang artinya sekertaris sangat setuju dengan fitur tersebut. Terakhir, untuk pengujian UAT melakukan komplain oleh siswa maupun orang tua memperoleh presentase nilai sebesar 96% yang artinya siswa ataupun orang tua sangat setuju dengan fitur tersebut. Jadi, dapat diambil kesimpulan bahwa Sistem Informasi Lembaga Bimbingan Belajar Al-Hasyimi ini dapat diterima dengan baik.



BAB 8 KESIMPULAN

8.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, dapat diambil kesimpulan bahwa Sistem Informasi Lembaga Bimbingan Belajar Al-Hasyimi dapat dikembangkan dengan melakukan beberapa proses pengembangan sebagai berikut:

1. Hasil analisis kebutuhan merupakan dokumentasi berupa hasil identifikasi *stakeholder*, masalah yang dihadapi, dan deskripsi kebutuhan pengguna. Selanjutnya, membuat proses bisnis usulan untuk menyelesaikan permasalahan yang ada pada proses bisnis saat ini. Diperoleh 3 bisnis usulan diantaranya mengelola hasil belajar, mengelola presensi, dan mengelola komplain. Ketiga bisnis usulan tersebut dipermudah dengan adanya bantuan dari sistem yang dikembangkan. Kemudian hasil dari pemodelan proses bisnis usulan tersebut akan dijadikan acuan untuk melakukan analisis kebutuhan fungsional sistem.
2. Kegiatan analisis terhadap kebutuhan pengguna dilakukan sehingga kebutuhan sistem dapat teridentifikasi. Identifikasi fitur pada sistem menghasilkan 13 fitur, dan spesifikasi kebutuhan pada sistem menghasilkan 17 kebutuhan fungsional dan 1 kebutuhan non-fungsional. Dari hasil spesifikasi kebutuhan tersebut akhirnya menghasilkan diagram *Use Case* yang menggambarkan interaksi antara aktor dan sistem.
3. Perancangan sistem yang dilakukan berdasarkan hasil analisis persyaratan dalam penelitian ini adalah model interaksi objek yang divisualisasikan ke dalam *sequence diagram*. Pemodelan objek dilakukan dengan menggunakan diagram kelas analisis sebagai acuan untuk pembuatan diagram kelas *model* dan *controller*. *PDM* yang berisi tabel-tabel sebagai media penyimpanan data sistem informasi yang terdiri dari 10 tabel. Perancangan algoritme menggunakan pseudocode dan *user interface* digunakan untuk acuan dalam implementasi dalam bentuk kode program. Implementasi sistem menggunakan *framework CodeIgniter* dengan pola *Model-View-Controller*.
4. Hasil pengujian validasi pada beberapa fungsi utama diantaranya menambahkan hasil belajar, melihat hasil belajar, melihat presensi tutor, dan melakukan komplain menunjukkan hasil pengujian 100%. Hasil tersebut menunjukkan bahwa fungsi-fungsi tersebut dapat berjalan dengan baik. Selain itu pada pengujian *compatibility* menggunakan software SortSite menunjukkan sistem dapat dijalankan diberbagai *browser* dengan versi yang berbeda-beda. Hasil *User Acceptance Testing (UAT)* yang dilakukan pada beberapa fitur, diantaranya menambahkan hasil belajar siswa memperoleh presentase sebesar 94%, melihat hasil belajar memperoleh presentase sebesar 90%, melihat presensi tutor memperoleh presentase 95%, dan melakukan komplain memperoleh presentase sebesar 96%. Berdasarkan hasil pengujian *UAT* tersebut, maka Sistem Informasi Lembaga Bimbingan Belajar Al-Hasyimi dapat diterima oleh pengguna.

8.2 Saran

Saran yang disampaikan untuk penelitian selanjutnya ialah:

1. Diperlukannya pengembangan lebih lanjut dari aplikasi ini dengan menambahkan fitur pendaftaran *online*, pembayaran *online*, dan fitur penjadwalan sehingga dapat lebih terkontrol dengan baik proses bimbingan ini.
2. Diperlukannya pengembangan *user interface* yang lebih baik untuk meningkatkan *user experience* dan *usability* pengguna, sehingga pengguna dapat lebih mudah dan nyaman menggunakan sistem ini.



DAFTAR PUSTAKA

- A. S., Rosa dan Shalahuddin, M. 2014. *Rekayasa Perangkat Lunak Terstruktur Dan Berorientasi Objek*. Informatika. Bandung: INFORMATIKA.
- Alshamrani, A., dan Bahattab, A. 2015. *A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model*. [online] Tersedia di: <<http://IJCSI.org>> [Diakses 12 Februari 2018].
- Bittner, K. dan Spence, I. 2002. *Use Case Modelling*. U.S : Addison Wesley.
- Burhanudin, T., 2011. *Lima hal kenapa website itu penting*. [online] Tersedia di: <<https://marketing.co.id/lima-hal-kenapa-website-itu-penting>> [Diakses 4 Februari 2018].
- Darmawan, D. dan Permana, Hendra P., 2013. *Desain dan Pemrograman website*. Bandung: PT Remaja RosdaKarya.
- Hambling, B. dan Goethem, P. V., 2013. *User Acceptance Testing*. United Kingdom: BTS.
- Jannah, W., Astuti, Indah F., & Maharani, S., 2015. *Rancang bangun sistem informasi Bimbingan Belajar berbasis Web (Studi Kasus: Lembaga Bimbingan Belajar TADICA)*. [online] Tersedia di: <<http://e-journals.unmul.ac.id/index.php/JIM/article/view/20>> [Diakses 2 Februari 2018].
- Kurniawan, Tri A. 2018. *Pemodelan Use Case (UML): Evaluasi Terhadap Beberapa Kesalahan dalam Praktik*. [online] Tersedia di: <<http://jtiik.ub.ac.id>> [Diakses 2 Februari 2018].
- Laudon, Kenneth C. & Laudon, Jane P. 2014. *Management Information System: Managing the digital firm 13 th*. [e-book] England: Pearson. Tersedia di: <<http://dinus.ac.id>> [Diakses 12 Februari 2018].
- Mahalakshmi, M., dan Sundararajan, M. 2013. *Traditional SDLC Vs Scrum Methodology – A Comparative Study*. Tersedia di: <<http://ijetae.com>> [Diakses 12 Februari 2018].
- Melnik, G., Read, K., & Maurer F. 2004. *Suitability of FIT User Acceptance Tests for Specifying Functional Requirements: Developer Perspective*. Tersedia di: <<http://springer.com>> [Diakses 12 Februari 2018].
- Naik, K. 2008. *Software Testing And Quality Assurance*. Universitas Of Waterloo:Wiley/
- O'Brien, James A., & Marakas, George M. 2010. *Management Information Systems 10 th*. [e-book] US : McGraw-Hill. Tersedia di: <<http://fumblog.um.ac.ir>> [Diakses 6 Februari 2018].
- Pratama, Antonius N.W., 2010. *Codeigniter: cara mudah membangun Aplikasi PHP*. Jakarta: PT TransMedia.

- Pressman, Roger S., 2001. *Software Engineering : A Practitioner's Approach — 5th ed.* [e-book] US : McGraw-Hill. Tersedia di: <<https://downloadnema.com>> [Diakses 27 September 2016].
- Rosing, Mark V., White, S., Cummins, F., & Man, Henk D. 2015. *Business Process Model and Notation.* [pdf] Tersedia di: <http://www.omg.org/news/whitepapers/Business_Process_Model_and_Notation.pdf> [Diakses 13 Februari 2018].
- Rumbaugh, J., Jacobson, I. & Booch, G., 2005. *The Unified Modeling Language reference manual.* 2nd ed. [e-book] Boston: Addison-Wesley. Tersedia di: <<https://www.utdallas.edu>> [Diakses 2 Februari 2018].
- Schade, A., 2014. *Responsive Web Design (RWD) and User Experience.* [online] Tersedia di: <<https://www.nngroup.com/articles/responsive-web-design-definition>> [Diakses 12 Februari 2018].
- Sommerville, I., 2011. *Software Engineering 9 th.* [e-book] U.S.A : Addison-Wesley. Tersedia di: <<https://edisciplinas.usp.br>> [Diakses 19 Desember 2016].
- Telkom University. 2017. *Panduan User Acceptance Testing (UAT).* [pdf] Tersedia di: <<http://dac.telkomuniversity.ac.id/wp-content/uploads/2015/06/PAKA06A-Panduan-User-Acceptance-Test-UAT-20170410.pdf>> [Diakses 12 Februari 2018].
- Weske, M., 2007. *Business Process Management Concept, Languages, Architertures.* [e-book] New York: Springer. Tersedia di: <<http://manajemen-pemasaran.com>> [Diakses 21 Desember 2016].