IMPLEMENTASI MONGODB DALAM PENGEMBANGAN SISTEM INFORMASI PENERIMAAN SANTRI BARU (PSB) BERBASIS WEB DENGAN METODE *WATERFALL*

**(Studi Kasus : MAS Nurul As’adiyah Callaccu kota Sengkang)**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan

memperoleh gelar Sarjana Komputer

Disusun oleh:

Nurul Inayah

NIM : 165150201111151



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2020

**DAFTAR ISI**

[**DAFTAR ISI** i](#_Toc42320100)

[**DAFTAR GAMBAR** iii](#_Toc42320101)

[**DAFTAR TABEL** iv](#_Toc42320102)

[**BAB 1 PENDAHULUAN** 1](#_Toc42320103)

[**1.1 Latar Belakang** 1](#_Toc42320104)

[**1.2 Rumusan Masalah** 2](#_Toc42320105)

[**1.3 Tujuan** 2](#_Toc42320106)

[**1.4 Manfaat** 2](#_Toc42320107)

[**1.5 Batasan Masalah** 3](#_Toc42320108)

[**1.6 Sistematika Pembahasan** 3](#_Toc42320109)

[**BAB 2 LANDASAN KEPUSTAKAAN** 5](#_Toc42320110)

[**2.1 Kajian Pustaka** 5](#_Toc42320111)

[**2.2 Sistem Informasi** 8](#_Toc42320112)

[**2.3 PPDB** 9](#_Toc42320113)

[**2.4 Rekayasa Perangkat Lunak** 9](#_Toc42320114)

[2.4.1 Model Pengembangan Perangkat Lunak 9](#_Toc42320115)

[2.4.2 Pendekatan Berorientesi Objek 14](#_Toc42320116)

[2.4.3 Pemodelan Berorientesi Objek 14](#_Toc42320117)

[**2.5 Teknologi Pengembangan Sistem** 17](#_Toc42320118)

[2.5.1 Bahasa Pemrograman PHP 17](#_Toc42320119)

[2.5.2 Bootstrap 18](#_Toc42320120)

[2.5.3 Cascading Style Sheets (CSS) 18](#_Toc42320121)

[2.5.4 NoSQL 18](#_Toc42320122)

[2.5.5 Codeigniter (CI) 19](#_Toc42320123)

[**BAB 3 METODOLOGI PENELITIAN** 20](#_Toc42320124)

[**3.1 Studi Pustaka** 20](#_Toc42320125)

[**3.2 Rekayasa Kebutuhan** 21](#_Toc42320126)

[**3.3 Perancangan Sistem** 21](#_Toc42320127)

[**3.4 Implementasi Sistem** 22](#_Toc42320128)

[**3.5 Pengujian** 22](#_Toc42320129)

[**3.7 Kesimpulan dan Saran** 22](#_Toc42320130)

[**BAB 4 ANALISIS KEBUTUHAN** 23](#_Toc42320131)

[**4.1 Elisitasi Kebutuhan** 23](#_Toc42320132)

[**4.2 Deskripsi Umum Sistem** 26](#_Toc42320133)

[**4.3 Identifikasi Aktor** 27](#_Toc42320134)

[**4.4 Spesifikasi Kebutuhan** 28](#_Toc42320135)

[4.4.1 Kebutuhan Fungsional 28](#_Toc42320136)

[4.4.2 Kebutuhan Non-Fungsional 43](#_Toc42320137)

[1 43](#_Toc42320138)

[*Usability* 43](#_Toc42320139)

[Sistem yang dikembangkan dapat digunakan dengan mudah dioperasikan oleh aktor. 43](#_Toc42320140)

[**4.5 Pemodelan Kebutuhan** 43](#_Toc42320141)

[4.5.1 Use Case Diagram 44](#_Toc42320142)

[4.5.2 Use Case Scenario 44](#_Toc42320143)

[**BAB 5 PERANCANGAN DAN IMPLEMENTASI** 80](#_Toc42320144)

[**5.1 Perancangan** 80](#_Toc42320145)

[5.1.1 Perancangan Arsitektur 80](#_Toc42320146)

[5.1.2 Perancangan *Database* 83](#_Toc42320147)

[5.1.3 Perancangan Komponen 83](#_Toc42320148)

[5.1.4 Perancangan Antarmuka Pengguna 83](#_Toc42320149)

[**DAFTAR PUSTAKA** 84](#_Toc42320150)

**DAFTAR GAMBAR**

[Gambar 2.1 Metode Waterfall 10](file:///C:\Users\ASUS\Pictures\skripsi\Progres\progres%20BAB%201-4%20(Autosaved).docx#_Toc37868635)

[Gambar 2.2 Blcak-box Testing 13](file:///C:\Users\ASUS\Pictures\skripsi\Progres\progres%20BAB%201-4%20(Autosaved).docx#_Toc37868636)

[Gambar 3.1 Diagram Alur Penelitian 20](file:///C:\Users\ASUS\Pictures\skripsi\Progres\progres%20BAB%201-4%20(Autosaved).docx#_Toc37868637)

[Gambar 4.1 Pengumuman Pembukaan PSB 23](#_Toc37868638)

[Gambar 4.2 Alur Pendaftaran 24](#_Toc37868639)

[Gambar 4.3 Seleksi Penerimaan dan Pengumuman 25](#_Toc37868640)

[Gambar 4.4 Pembayaran SPP atau Uang Madrasah 26](#_Toc37868641)

[Gambar 4.5 Aturan Penomoran 28](file:///C:\Users\ASUS\Pictures\skripsi\Progres\progres%20BAB%201-4%20(Autosaved).docx#_Toc37868642)

[Gambar 4.6 Use Case Diagram 44](#_Toc37868643)

**DAFTAR TABEL**

[Tabel 2.1 Daftar Kajian Pustaka 6](#_Toc37868574)

[Tabel 2.2 Contoh Simbol Use Case Diagram 15](#_Toc37868575)

[Tabel 2.3 Contoh Simbol Sequence Diagram 16](#_Toc37868576)

[Tabel 2.4 Contoh Notasi Class Diagram 17](#_Toc37868577)

[Tabel 4.1 Identifikasi Aktor 27](#_Toc37868578)

[Tabel 4.2 Kebutuhan Fungsional 29](#_Toc37868579)

[Tabel 4.3 Kebutuhan Non-Fungsional 43](#_Toc37868580)

[Tabel 4.4 Use Case Scenario Login 44](#_Toc37868581)

[Tabel 4.5 Use Case Scenario Buat Akun 45](#_Toc37868582)

[Tabel 4.6 Use Case Scenario Ganti Password 46](#_Toc37868583)

[Tabel 4.7 Use Case Scenario Logout 46](#_Toc37868584)

[Tabel 4.8 Use Case Scenario Unduh Panduan 47](#_Toc37868585)

[Tabel 4.9 Use Case Scenario Lihat Hasil Tes 47](#_Toc37868586)

[Tabel 4.10 Use Case Scenario Hubungi Admin 48](#_Toc37868587)

[Tabel 4.11 Use Case Scenario Lihat Informasi Sekolah 48](#_Toc37868588)

[Tabel 4.12 Use Case Scenario Ubah Informasi Sekolah 49](#_Toc37868589)

[Tabel 4.13 Use Case Scenario Lihat Informasi PSB 50](#_Toc37868590)

[Tabel 4.14 Use Case Scenario Ubah Informasi PSB 50](#_Toc37868591)

[Tabel 4.15 Use Case Scenario Lihat Daftar Pesan 51](#_Toc37868592)

[Tabel 4.16 Use Case Scenario Hapus Pesan 51](#_Toc37868593)

[Tabel 4.17 Use Case Scenario Balas Pesan 52](#_Toc37868594)

[Tabel 4.18 Use Case Scenario Lihat Daftar Pengguna 52](#_Toc37868595)

[Tabel 4.19 Use Case Scenario Tambah Pengguna 53](#_Toc37868596)

[Tabel 4.20 Use Case Scenario Hapus Pengguna 54](#_Toc37868597)

[Tabel 4.21 Use Case Scenario Isi Form Pendaftaran 54](#_Toc37868598)

[Tabel 4.22 Use Case Scenario Lihat Profil 55](#_Toc37868599)

[Tabel 4.23 Use Case Scenario Ubah Profil 55](#_Toc37868600)

[Tabel 4.24 Use Case Scenario Cetak Kartu Peserta 56](#_Toc37868601)

[Tabel 4.25 Use Case Scenario Lihat Tes Seleksi 57](#_Toc37868602)

[Tabel 4.26 Use Case Scenario Mulai Tes 57](#_Toc37868603)

[Tabel 4.27 Use Case Scenario Pilih Program Studi 58](#_Toc37868604)

[Tabel 4.28 Use Case Scenario Lihat Daftar Prasyarat 59](#_Toc37868605)

[Tabel 4.29 Use Case Scenario Unggah Berkas Prasyarat 59](#_Toc37868606)

[Tabel 4.30 Use Case Scenario Ubah Berkas Prasyarat 60](#_Toc37868607)

[Tabel 4.31 Use Case Scenario Lihat Daftar Peserta PSB 61](#_Toc37868608)

[Tabel 4.32 Use Case Scenario Cetak Daftar Peserta 61](#_Toc37868609)

[Tabel 4.33 Use Case Scenario Lihat Detail Peserta PSB 62](#_Toc37868610)

[Tabel 4.34 Use Case Scenario Verifikasi Pendaftaran 63](#_Toc37868611)

[Tabel 4.35 Use Case Scenario Lihat Daftar Arsip Data Santri 63](#_Toc37868612)

[Tabel 4.36 Use Case Scenario Lihat Detail Arsip Data Santri 64](#_Toc37868613)

[Tabel 4.37 Use Case Scenario Lihat Daftar Soal Tes 64](#_Toc37868614)

[Tabel 4.38 Use Case Scenario Tambah Soal Tes 65](#_Toc37868615)

[Tabel 4.39 Use Case Scenario Ubah Soal Tes 65](#_Toc37868616)

[Tabel 4.40 Use Case Scenario Kirim Soal Tes 66](#_Toc37868617)

[Tabel 4.41 Use Case Scenario Hapus Soal Tes 67](#_Toc37868618)

[Tabel 4.42 Use Case Scenario Lihat Daftar Jawaban 67](#_Toc37868619)

[Tabel 4.43 Use Case Scenario Lihat Detail Jawaban 68](#_Toc37868620)

[Tabel 4.44 Use Case Scenario Nilai Jawaban 68](#_Toc37868621)

[Tabel 4.45 Use Case Scenario Input Nilai Tes Akademik 69](#_Toc37868622)

[Tabel 4.46 Use Case Scenario Input Nilai Tes Wawancara 70](#_Toc37868623)

[Tabel 4.47 Use Case Scenario Lihat Akumulasi Tes 71](#_Toc37868624)

[Tabel 4.48 Use Case Scenario Cetak Hasil Akumulasi 71](#_Toc37868625)

[Tabel 4.49 Use Case Scenario Lihat Daftar Santri Aktif 72](#_Toc37868626)

[Tabel 4.50 Use Case Scenario Cetak Data Santri 72](#_Toc37868627)

[Tabel 4.51 Use Case Scenario Lihat Detail Data Santri 73](#_Toc37868628)

[Tabel 4.52 Use Case Scenario Lihat Riwayat SPP 73](#_Toc37868629)

[Tabel 4.53 Use Case Scenario Verifikasi SPP 74](#_Toc37868630)

[Tabel 4.54 Use Case Scenario Lihat Data Diri 74](#_Toc37868631)

[Tabel 4.55 Use Case Scenario Lihat Tagihan SPP 75](#_Toc37868632)

[Tabel 4.56 Use Case Scenario Konfirmasi Pembayaran SPP 75](#_Toc37868633)

[Tabel 4.57 Use Case Scenario Lihat Riwayat Pembayaran SPP 76](#_Toc37868634)

# **PENDAHULUAN**

## **1.1 Latar Belakang**

Proses Penerimaan Santri Baru (PSB) atau umumnya dikenal Penerimaan Peserta Didik Baru (PPDB) merupakan serangkaian kegiatan yang dilakukan oleh instansi pendidikan seperti sekolah untuk menyeleksi para calon siswa yang ingin masuk ke sekolah tertentu. PSB rutin dilaksanakan oleh beberapa sekolah di berbagai wilayah Indonesia salah satunya yaitu Madrasah Aliyah Swasta (MAS) Nurul As’adiyah Callacu yang berlokasi di Kota Sengkang, Sulsel.

MAS Nurul As’adiyah Callaccu merupakan salah satu lembaga pendidikan agama yang berdiri sejak tahun 1999 dipusat ibu kota Kabupaten Wajo, telah banyak melahirkan generasi intelektual islam muda yang berjiwa pejuang dan petualang mereka disiapkan menjadi generasi cerdas dan mampu memberikan dampak yang luar biasa terhadap kemajuan islam di Indonesia. Visi MAS Nurul As’adiyah Callaccu yaitu terciptanya sumber daya manusia yang cerdas, santun, terampil dan inovatif dilandasi keimanan dan ketaqwaan yang diridhai Allah SWT, untuk mewujudkan visi tersebut MAS Nurul As’adiyah Callaccu memiliki misi, (1) Menumbuhkan semangat belajar siswa sehingga mampu berprestasi secara optimis sesuai potensi yang dimiliki; (2) Menumbuhkan sikap dan kepribadian yang santun, beretika dan berestetika tinggi; (3) Menumbuhkan kesadaran wawasan kebangsaan, patriotisme dan memelihara persatuan dan kesatuan bangsa dalam kerangka NKRI; (4) Mengembangkan kemampuan vokasional skill yang mampu dan dapat diterima masyarakat luas; (5) Menumbuhkan kreatifitas dan inovasi siswa; (6) Menumbuhkan kesadaran siswa agar mampu menjadikan ajaran dan nilai nilai islam sebagai landasan pola berfikir.

Saat ini proses dan informasi PSB di MAS Nurul As’adiyah Callaccu masih manual yaitu menggunakan media kertas. Hal itu mengakibatkan beberapa masalah yaitu:

1. Kebanyakan calon santri yang mendaftar berasal dari luar daerah dengan persentase sebesar 70% sehingga calon santri tersebut harus mengunjungi sekolah untuk melakukan proses pendaftaran dan membawa pulang formulir untuk melengkapi berkas prasyarat.
2. Penyebaran informasi yang masih menggunakan brosur, sehingga penyebaran informasi hanya dapat disampaikan kebeberapa daerah dan orang tertentu saja, selain itu akan memakan tenaga dan biaya yang cukup besar bagi pihak sekolah.
3. Proses pendaftaran yang masih menggunakan kertas membuat adanya penumpukan dokumen, sebab berkas fisik yang diserahkan akan disimpan kedalam map sesuai nama masing-masing calon santri.
4. Proses yang masih menggunakan kertas tersebut juga menyebabkan tugas tambahan untuk panitia PSB yaitu memasukkan data ke excel, sehingga cukup menyita waktu dan tenaga panitia PSB mengingat jumlah pendaftar pada tahun 2019 mencapai 240 orang.
5. Pengumuman hasil seleksi masih menggunakan kertas yang ditempel pada papan pengumuman sekolah sehingga menyulitkan calon santri yaitu harus mengunjungi sekolah.

Melihat permasalahan yang dialami, maka peneliti menawarkan solusi berupa “Implementasi mongoDB untuk Sistem Informasi Penerimaan Santri Baru (PSB) pada MAS Nurul As’adiyah Callaccu Sengkang”, dengan melakukan penelitian ini diharapkan sistem yang dikembangkan dapat meminimalisir biaya karena proses pendaftaran dilakukan menggunakan sistem online dan mengurangi risiko dokumen hilang, sistem yang diusulkan juga diharapkan mengurangi beban kerja panitia karena tidak lagi memasukkan data dari kertas ke excel dan membantu penyebaran informasi.

## **1.2 Rumusan Masalah**

Berdasarkan pemaparan permasalahan pada bagian latar belakang, ditemukan rumusan masalah sebagai berikut.

1. Bagaimana hasil analisis Sistem Informasi Penerimaan Santri Baru Pada MAS Nurul As’adiyah Callaccu Kota Sengkang?
2. Bagaimana hasil perancangan Sistem Informasi Penerimaan Santri Baru Pada MAS Nurul As’adiyah Callaccu Kota Sengkang?
3. Bagaimana hasil mengimplementasi Sistem Informasi Penerimaan Santri Baru Pada MAS Nurul As’adiyah Callaccu Kota Sengkang?
4. Bagaimana hasil pengujian pada Sistem Informasi Penerimaan Santri Baru Pada MAS Nurul As’adiyah Callaccu Kota Sengkang?

## **1.3 Tujuan**

Berdasarkan pemaparan rumusan masalah diatas, adapun tujuan yang ingin diperoleh dari penelitian ini sebagai berikut yang telah dipaparkan adalah sebagai berikut.

1. Menganalisis Sistem Informasi Penerimaan Santri Baru Pada MAS Nurul As’adiyah Callaccu Kota Sengkang.
2. Merancang Sistem Informasi Penerimaan Santri Baru Pada MAS Nurul As’adiyah Callaccu Kota Sengkang.
3. Mengimplementasikan Sistem Informasi Penerimaan Santri Baru Pada MAS Nurul As’adiyah Callaccu Kota Sengkang.
4. Dapat melakukan pengujian Sistem Informasi Penerimaan Santri Baru Pada MAS Nurul As’adiyah Callaccu Kota Sengkang.

## **1.4 Manfaat**

Berdasarkan penelitian ini, maka manfaat yang diperoleh dari penelitian ini sebagai berikut.

1. Mencegah terjadinya penumpukan data.
2. Menghemat biaya dan tenaga selama proses penerimaan santri baru pada MAS Nurul As’adiyah Callaccu Sengkang.
3. Memberi akses yang luas pada masyarakat.
4. Meningkatkan reputasi sekolah.
5. Memudahkan siswa dalam mengakses informasi penerimaan santri baru pada MAS Nurul As’adiyah Callaccu Sengkang.
6. Mempermudah tenaga pendidik dalam penyimpanan data santri.

**1.5 Batasan Masalah**

Berdasarkan penelitian ini, batasan masalah dalam penelitian ini sebagai berikut:

Sistem hanya dapat dilakukan untuk proses penerimaan santri baru pada MAS Nurul As’adiyah Callaccu Sengkang.

Sistem hanya dapat mengelola proses pendaftaran, pembayaran, seleksi masuk, dan pengumuman.

Sistem ini berbasis website dengan menggunakan *framework codeigniter* dan CSS Bootstrap.

Sistem menggunakan database *MySql*.

**1.6 Sistematika Pembahasan**

Pada bagian ini akan menjelaskan dari pelaksanaan penelitian yang digunakan sebagai gambaran serta uraian dari laporan penelitian ini secara umum dibagi menjadi enam bab. Penjelasan singkat uraian dari beberapa bagian bab antara lain:

**BAB 1 Pendahuluan**

Pada bagian ini membahas mengenai latar belakang dilakukannya sebuah penelitian yang dijelaskan dalam sub bab dari latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan dalam membangun Sistem Informasi PSB Pada MAS Nurul As’adiyah Callaccu Sengkang.

**BAB 2 Landasan Kepustakaan**

Pada bagian ini membahas terkait dasar teori dan kajian pustaka perihal penelitian yang telah dilakukan sebelumnya sebagai bahan referensi dalam pengembangan Sistem Informasi PSB Pada MAS Nurul As’adiyah Callaccu Sengkang.

**BAB 3 Metodologi Penelitian**

Pada bagian ini akan membahas terkait alur penelitian yang akan dilakukan sehingga akan lebih terarah, alur dibagi menjadi beberapa sub bab studi pustaka, rekayasa kebutuhan, perancangan sistem, implementasi sistem, pengujian dan kesimpulan dan saran.

**BAB 4 Anlisis Kebutuhan**

Pada bagian ini akan membahas terkait kebutuhan apa yang harus ada dalam sistem yang akan dikembangkan yakni sistem informasi PSB dan mengidentifikasi aktor yang akan berinteraksi dengan sistem. Kebutuhan sistem diperoleh melalui wawancara dengan panitia PSB, kemudian kebutuhan tersebut akan dimodelkan dalam bentuk use case diagram serta use case scenario.

# **LANDASAN KEPUSTAKAAN**

## **2.1 Kajian Pustaka**

Kajian pustaka dilakukan oleh (Fandhilah, Rindina, Ferdiansyah, & Ishaq, 2019) terkait penerapan metode waterfall dalam pengembangan sistem informasi penerimaan siswa baru pada SMK negeri 2 Adiwerna. Pengembangan sistem penerimaan siswa baru memiliki kebutuhan yang sudah jelas diawal sehingga dengan menggunakan metode *waterfall* akan sesuai dengan proses pengembangan sistem. Dengan penerapan metode waterfall dihasilkan sebuah sistem penerimaaan siswa baru dan dengan sistem tersebut pendaftaran dapat dilakukan dengan efisien, dapat meminimalisir masalah yang saat ini terjadi pada proses pendaftaran dan dapat meningkatkan kredibilitas sekolah. Berdasarkan penelitian tersebut, peneliti akan menggunakan metode waterfall dalam pengembangan sistem penerimaan santri baru baru pada MAS Nurul As’Adiyah Callaccu Sengkang.

Kajian pustaka kedua dilakukan oleh (Mutiarin & Wijaya, 2017) terkait evaluasi penerapan SIAP-PPDB online dalam meningkatkan mutu layanan pendidikan pada koa Yogyakarta, penelitian dilakukan dengan metode deskriptif kualitatif meliputi observasi, wawancara dan dokumentasi. Hasil yang didapatkan dari penelitian tersebut yaitu meningkatnya mutu layanan program SIAP-PPDB, terbukti dengan meningkatnya angka kepuasan masyarakat dari tahun 2013 hingga 2015 pada penerapan SIAP-PPDB online Kota Yogyakarta.

Kajian pustaka ketiga oleh (Silalahi, 2018) terkait perbandingan performasi database MongoDb dan MySql dalam aplikasi *file* multimedia berbasis web, dalam pengujiannya peneliti menggunakan 5 variabel yaitu penggunaan processor, memori fisik, memori virtual dan kecepatan ekseskusi dengan 3 ukuran data yaitu kecil 0-5 MB, sedang 6-100 MB dan besar >100 MB. Pengujian dilakukan pada proses create, read dan delete, pada proses create mongoDB 3,6 kali lebih cepat pada ukuran data kecil, 3,7 kali lebih cepat pada ukuran data sedang dan 2,48 kali lebih cepat pada ukuran data besar dengan penggunaan memori virtual lebih tinggi. Pada proses read pemakaian processor di server aplikasi tidak berbeda jauh untuk *file* berukuran kecil dan sedang, tetapi ketika mengunduh *file* berukuran besar, pemakaian processor MySQL melonjak tinggi. Konsumsi memori MySQL untuk *file* berukuran sedang lebih sedikit daripada MongoDB. Dan ketika melayani *file* berukuran sedang dan besar, penggunaan memori virtual MySQL lebih besar. Pada proses delete pemakaian sumber daya komputer berimbang antara MySQL dan MongoDB. Kecepatan juga tidak begitu terasa berbeda (walaupun untuk *file* berukuran kecil MySQL lebih unggul). Kecepatan terasa berbeda ketika menghapus *file* berukuran besar di mana mongoDb lebih unggul.

Kajian pustaka keempat dilakukan oleh (Alshamrani & Bahattab, 2015) terkait perbandingan beberapa model SDLC diantaranya waterfall model, spiral model dan incremental/iterative model. Waterfall model memiliki keunggulan mudah untuk dipahami, kebutuhan sistem didefinisikan terlebih dahulu sebelum di design, meminimalkan planning overhead, dll. Dalam kegunaannya waterfall cocok untuk sistem yang mementingkan kualitas daripada biaya, kebutuhan sistem sangat jelas dan tetap dan porting produk yang ada ke *platform* baru. Spiral model cocok digunakan pada proyek-proyek berisiko tinggi, ketika perubahan yang signifikan diharapkan dan kebutuhan pengguna belum diketahui dengan pasti. Sedangkan untuk incremental/iterative model digunakan untuk sistem dengan risiko menengah, memungkinkan pengguna untuk menyesuaikan diri dengan sistem jika pada ptoyek menggunakan teknologi baru dan juga ketika berisiko tinggi untuk mengembangkan seluruh sistem sekaligus.

Kajian pustaka kelima dilakukan oleh (Bhakti et al., 2019) terkait pengembangan sistem PPDB pada Sekolah Menengah Kejuruan, penelitian dilakukan karena seringnya terjadi kehilangan data karena formulir pendaftaran ditumpuk dengan *file* lain serta kurangnya manajemen administrasi rekapitulasi data siswa, dari masalah tersebut dibuat sebuah sistem PPDB online guna mengatasi permasalahan tersebut. Penelitian dilakukan dengan model *waterfall* karena dapat tahapannya teratur dan mudah dipahami, sistem dibangun dengan berbasis web. Sistem diuji kepada pengguna aplikasi yaitu calon pelamar dan komite PPDB dengan jumlah sampel 20 calon pelamar dan 5 komite PPDB, hasil pengujian menyatakan sistem dalam kategori sangat layak dengan persentase dari calon pelamar sebesar 89,9% dan dari komite PPDB sebesar 88,2%.

**Tabel 2.1 Daftar Kajian Pustaka**

|  |  |  |  |
| --- | --- | --- | --- |
| Judul | Object | Metode | Hasil Penelitian |
| Implementasi Metode Waterfall Pada Pengembangan Sistem Informasi Penerimaan Siswa Baru Berbasis Wes Pada SMK Negeri 2 Adiwerna (Fandhilah, Rindina, Ferdiansyah, & Ishaq, 2019) | SMK Negeri 2 Adiwerna | Waterfall | * Sistem penerimaan siswa baru * Dengan sistem PSB pendaftaran dapat dilakukan dengan efisien, meminimalisir masalah media informasi dan meningkatkan promosi. Sistem PSB juga dapat meningkatkan kredibilitas sekolah dan nilai tambah dengan berbagai kemudahan akses |
| Evaluasi Penerapan SIAP-PPDB Online Dalam Meningkatkan Mutu Layanan Pendidikan (Mutiarin & Wijaya, 2017) | Kota Yogyakarta | Deskripsi kualitatif (observasi, wawancara dan dokumentasi) | * Mutu layanan program SIAP-PPDB semakin meningkat dengan baik, terbukti dengan meningkatnya kepuasan masyarakat pada penerapan SIAP-PPDB online Kota Yogyakarta. * SIAP-PPDB menciptakan penerimaan siswa baru dengan lebih praktis, efektif dan memberi fasilitas akses informasi bagi masyarakat dengan cepat, mudah dan akurat |
| Studi Perbandingan Per*form*asi Antara MongoDB dan MySQL Dalam Aplikasi *File* Multimedia Berbasis Web (Silalahi, 2018) | Aplikasi *File* Multimedia | Pengujian pada processor, kecepatan, penggunaan memori fisik dan memori virtual. | Aplikasi *File* Multimedia berbasis web menggunakan PHP, dengan basis data MongoDB lebih cepat dalam operasi CREATE untuk semua ukuran *file* dan lebih cepat dalam operasi DELETE untuk *file* berukuran besar, serta lebih irit dalam penggunaan sumber daya komputer server untuk operasi READ. |
| Perbandingan Antara Tiga Model SDLC Model Waterfall, Spiral Model, dan Incremental / Iterative Model (Alshamrani & Bahattab, 2015) | SDLC waterfall, spiran dan incremental/iterative | - |  |
| Pengembangan Aplikasi PPDB (Penerimaan Siswa Baru) Untuk Mengembangkan Kualitas Administrasi Rekapitulasi Siswa Baru di Sekolah Menengah Kejuruan Bumi Cikajang.(Bhakti et al., 2019) | SMK | Waterfall | Aplikasi PPDB berbasis web yang dapat membantu dalam proses rekapitulasi data calon siswa baru, sehingga mempermudah panitia PPDB dalam pelaksanaan penerimaan siswa baru berdasarkan hasil pengujian ke seluruh calon siswa dengan persentase 88,2% (sangat layak). |

Berdasarkan kajian pustaka terhadap 5 penelitian yang telah dilakukan oleh sebelumnya, disimpulkan bahwa PSB online penting untuk diterapkan pada sekolah dibuktikan dengan meningkatnya kepuasan masyarakat pada SIAP-PPDB, kemudian penerapan model waterfall sebagai model pengembangan perangkat lunak dapat menghasilkan sistem yang baik, waterfall juga cocok diterapkan pada sistem PSB karena kebutuhan sistem dapat di definisikan sejak awal, serta penerapan mongoDB dalam sebuah sistem karena lebih cepat dalam operasi CREATE untuk semua ukuran *file* dan lebih cepat dalam operasi DELETE untuk *file* berukuran besar, serta lebih irit dalam penggunaan sumber daya komputer.

## **2.2 Sistem Informasi**

Informasi merupakan kumpulan data yang telah diolah atau data yang memiliki arti (Sutabri, 2012b). Data adalah fakta yang menggambarkan suatu kejadian dan kesatuan nyata(Sutabri, 2012b). Sistem marupakan suatu kumpulan dari komponen, unsur, maupun variabel yang tersusun atau terorganisasi, saling membutuhkan satu sama lain dan terpadu(Sutabri, 2012a). Sistem dapat berupa sistem yang terbuka dan tertutup, sistem informasi umumnya merupakan siste yang terbuka, artinya sistem tersebut mampu menerima berbagai inputan dari lingkungan sekitarnya(Sutabri, 2012a).

Sistem Informasi merupakan suatu kumpulan dari komponen yang saling berhubungan yang mengumpulkan, memngolah, menyimpan, dan menyediakan keluaran dari sebuah informasi yang dibutuhkan untuk dapat menyelesaikan tugas bisnis(Satzinger et al., 2012). Sistem informasi merupakan suatu sistem dalam sebuah organisasi yang mempertemukan keinginan pengolahan transaksi harian yang membantu fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi agar dapat menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu (Sutabri, 2012a).

## **2.3 PPDB**

Penerimaan Peserta Didik Baru yang umumnya dikenal dengan PPDB, adalah proses penerimaan siswa-siswi baru baru pada TK dan Sekolah, namun di lingkungan pesantren biasanya dikenal Penerimaan Santri Baru (PSB). PPDB dilakukan berdasarkan nondiskriminatif, objektif, transparan, akuntabel dan berkeadilan (Kebudayaan, 2018). Berdasarkan Permendikbud Nomor 51 tahun 2018 BAB II Pasal 4 tentang tata cara PPDB menyatakan, sekolah yang diselenggarakan oleh Pemerintah Daerah melaksanakan PPDB pada bulan Mei setiap tahun, dengan tahapan dimulai dari: (a) Pengumuman pendaftaran penerimaan calon santri baru baru pada Sekolah yang bersangkutan yang dilakukan secara terbuka; (b) Pendaftaran; (c) Seleksi sesuai jalur pendaftaran; (d) Pengumuman penetapan santri baru baru; dan (e) Daftar ulang. Pengumuman pendaftaran penerimaan calon santri baru baru sebagaimana dimaksud melalui papan pengumuman Sekolah maupun media lainnya dan penetapan santri baru baru dilakukan berdasarkan hasil rapat dewan guru yang dipimpin oleh kepala Sekolah dan ditetapkan melalui keputusan kepala Sekolah (Kebudayaan, 2018).

Berdasarkan hasil wawancara kepada narasumber Muhammad Akis selaku panitia PSB pada MAS Nurul As’adiyah Callaccu, Penerimaan Santri Baru (PSB) merupakan rangkaian penerimaan siswa-siswi baru yang ingin melanjutkan studi ke jenjang yang lebih tinggi, dengan tahapan proses dimulai dari pengumuman pembukaan PSB, kemudian dilanjutkan pendaftaran dalam jangka waktu tertentu, setelah itu proses seleksi masuk dengan seleksi berupa tes tulis dan wawancara sesuai dengan aturan sekolah masing-masing dan terakhir adalah pengumuman atau penetapan siswa-siswi yang diterima atau dinyatakan lolos untuk dapat melanjutkan pendidikan ke sekolah yang dituju.

## **2.4 Rekayasa Perangkat Lunak**

Rekayasa Perangkat Lunak menurut Ian Sommerville (2003:7) adalah ilmu yang mempelajari segala aspek pembuatan perangkat lunak, dimulai dari tahap spesifikasi sistem hingga perawatan Sistem saat setelah digunakan. Selain itu, menurut Pressman (2012:15) adalah aplikasi dari pendekatan yang sistematik, disiplin dan mampu diukur pada pengembangan, operasi dan perawatan perangkat lunak. Sehingga, umumnya rekayasa perangkat lunak menggunakan pendekatan yang sistematis dan terorganisir kepada pekerjaan mereka sebab cara ini paling efektif untuk menghasilkan keluaran perangkat lunak yang berkualitas tinggi (Sommerville, 2003)

Pada rekayasa perangkat lunak, terdapat beberapa model pengembangan yang dapat digunakan, masing-masing memiliki alur yang berbeda disesuaikan dengan kebutuhan pengembangan sistem. Selain itu, juga memiliki 2 pendekatan yaitu terstruktur dan berorientasi objek.

### 2.4.1 Model Pengembangan Perangkat Lunak

Model pengembangan perangkat lunak akan digunakan sebagai pedoman selama tahap proses pengembangan sistem dimana didalamnya terdapat arahan yang mendefinisikan seluruh rangkaian alur aktivitas yang dilakukan selama proses pengembangan perangkat lunak. *SDLC (Sistems Development Life Cycle*) merupakan keseluruhan proses ruang lingkup sistem dimulai pada tahap Pengembangan (*building*), menyebarkan (*deploying*), menggunakan (*using*) dan memperbarui (*updating*) sistem informasi (Satzinger et al., 2010). SDLC Model memiliki banyak model yang merepresentasikan pendekatan berbeda dalam tahapan pengembangan perangkat lunak (Sommerville, 2011).

*Waterfall* adalah salah satu contoh model pengembangan SDLC dengan metode pengembangan didasarkan dengan prinsip sekuensial, pada prinsipnya output dari setiap tahap adalah satu atau lebih dokumen yang telah disepakati artinya dalam tiap fase harus terjadi proses validasi dan verifikasi karena fase berikutnya tidak dapat dimulai sebelum fase sebelumnya selesai. Metode ini lebih fokus pada spesifikasi kebutuhan yang didefinisikan dengan jelas sejak awal sehingga dapat menghasilkan kualitas sistem yang baik. Waterfall menekankan pada perencanaan dan analisis kebutuhan di fase awal sehingga dapat menghasilkan perancangan yang baik tanpa adanya kecacatan desain (Sommerville, 2011). Metode *Waterfall* memiliki tahapan utama dari Waterfall model yang mencerminkan aktifitas pengembangan dasar. Pada Gambar 2.1 merupakan gambar alur pengembangan menggunakan metode pengembangan *Waterfall*, terdapat 5 (lima) tahapan pada metode *Waterfall*, yaitu definisi persyaratan, perancangan sistem dan perangkat lunak, implementasi dan pengujian unit, integrasi dan pengujian sistem dan operasi dan pemeliharaan.

Definisi persyaratan

Perancangan sistem dan perangkat lunak

Implementasi dan pengujian unit

Integrasi dan pengujian sistem

Operasi dan pemeliharaan

Requirement definition

Sistem and software design

Implementation and unit testing

Integration and Sistem testing

Operating and maintenance

**Gambar 2.1 Metode Waterfall**

Sumber: (Sommerville, 2003)

Penjelasan tiap fase pada metode W*aterfall* menurut (Sommerville, 2003):

* + - 1. Analisis dan definisi persyaratan

Pelayanan, batasan dan tujuan sistem ditentukan melalui wawancara dengan user sistem, persyaratan ini kemudian diartikan secara rinci, yang nantinya berfungsi sebagai spesifikasi sistem. Spesifikasi perangkat lunak bertujuan untuk mengetahui layanan apa yang harus terdapat dalam sistem serta batasannya, hasil dari fase ini adalah dokumen persyaratan yang merupakan kumpulan spesifikasi sistem. Pengembangan sistem memerlukan spesifikasi sistem yang lebih rinci, ada 4 fase utama pada proses rekyasa kebutuhan yaitu:

1. *Studi Kelayakan*, perkiraan tentang apakah user yang didefinisikan dapat puas menggunakan perangkat lunak dan teknologi perangkat keras yang gunakan saat ini.
2. *Eisitasi dan Analisis rekayasa kebutuhan*, adalah proses turunan persyaratan sistem dengan melalui observasi sistem yang telah ada, diskusi dengan user yang akan memakai dan yang akan mengadakan, analisis pekerjaan, dll. Hasil dari fase ini akan membantu analis untuk memahami sistem yang akan dispesifikasi.
3. *Spesifikasi Kebutuhan*, adalah kegiatan mengartikan informasi yang didapatkan pada kegiatan analisis kebutuhan untuk menjadi dokumen yang mendefinisikan sekumpulan kebutuhan.
4. *Validasi Kebutuhan*, pada tahap ini proses memeriksa apakah persyaratan mampu direalisasikan, konsisten dan lengkap. Selain itu, kesalahan pada dokumen dapat ditemukan pada fase ini sehingga kemudian dimodifikasi untuk dapat menyelesaikan masalahnya.
   * + 1. Perancangan sistem dan perangkat lunak

Proses perancangan sistem membagi persyaratan dalam sistem perangkat keras atau perangkat lunak. Kegiatan ini menentukan arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan deskripsi abstraksi sistem perangkat lunak yang mendasar dan hubungan-hubungannya. Kegiatan proses perancangan yang spesifik adalah:

1. Perancangan Arsitektural, subsistem-subsistem yang membentuk sistem dan hubungan mereka diidentifikasi dan didokumentasikan.
2. Spesifikasi abstrak, untuk setiap subsistem, spesifikasi abstrak dari layanan dan batas operasinya harus ditentukan.
3. Perancangan Antarmuka, untuk setiap subsistem, interface dengan subsistem dirancang dan didokumentasikan. Spesifikasi interface ini harus sudah jelas karena memungkinkan subsistem dipakai tanpa mengetahui operasi subsistem.
4. Perancangan Komponen, layanan dialokasikan pada komponen yang berbeda dan interface komponen-komponen ini dirancang.
5. Perancangan Struktur Data, struktur data yang dipakai pada implementasi sistem dirancang secara rinci dan dispesifikasi.
6. Perancangan Algoritma, algoritma yang digunakan untuk memberikan layanan dirancang secara rinci dan dispesifikasikan.
   * + 1. Implementasi dan pengujian unit

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian unit melibatkan verifikasi bahwa setiap unit telah memenuhi spesifikasinya.

Pengembangan program untuk implementasi sistem merupakan lanjutan dari proses perancangan sistem, implementasi ini mencakup kode untuk mendefinisikan dan mengimplementasikan interface. Setelah semua diimplementasikan maka dilanjutkan dengan pengujian unit, pengujian unit merupakan bagian dari proses implementasi dan diharapkan bahwa komponen yang sesuai dengan spesifikasinya akan diserahkan sebagai bagian dari proses tersebut.

* + - 1. Integrasi dan pengujian sistem

Unit program atau program individual diintegrasikan dan diuji sebagai sistem yang lengkap untuk menjamin bahwa persyaratan sistem telah dipenuhi. Setelah pengujian sistem, perangkat lunak dikirim kepada pelanggan. Pengujian perangkat lunak yang digunakan ada 2 yaitu:

1. *White-box Testing*

Pengujian struktural yang biasa dikenal dengan *white-box testing* merupakan pendekatan terhadap pengujian yang diturunkan dari pengetahuan struktur dan implementasi perangkat lunak. Pada dasarnya white box testing Sistem di perlukan dalam membangun ataupun menganalisis sebuah Sistem apakah sudah berjalan sebagai mana mestinya atau tidak. Hal ini perlu dilakukan mengingat white box testing merupakan pengecekan Sistem secara lebih detail dan kompleks ketimbang Sistem black box testing, di white box testing Sistem tidak hanya di cek dari fungsional luarnya saja "interface" melainkan juga dari berbagai aspek Sistem, termasuk diagram alur Sistem dan berbagai komponen lainnya yang ada dalam Sistem (Singh, 2010). Tujuan dari pengujian struktural adalah menjamin bahwa setiap jalur program yang independen dieksekusi paling tidak satu kali, jalur program independen adalah yang menelusuri paling tidak satu edge baru pada *flow graph,* dalam istilah program berarti melatih satu atau lebih kondisi baru, percabangan true dan false untuk semua kondisi harus dieksekusi.

*Path testing* adalah strategi pengujian struktural yang bertujuan untuk melatih setiap jalur eksekusi independen melalui komponen atau program. Titik awal *path testing* merupakan *flow graph*, *flow graph* terdiri dari node yang mewakili keputusan dan edge yang menunjukkan aliran kontrol, melalui *flow graph* jalur independen dapat diketahui, setelah itu dilakukan perhitungan kompleksitas siklomatik dengan menggunakan rumus berikut ini:

CC (G) : Jumlah (edge) – jumlah (node) + 2

Setelah menemukan jalur independen melalui kode dengan menghitung kompleksitas siklomatik, langkah berikutnya adalah merancang kasus uji untuk eksekusi setiap jalur tersebut, jumlah minimum kasus uji yang dibutuhkan untuk menguji semua jalur program sama dengan kompleksitas siklomatik.

1. *Black-box Testing*

Pengujian fungsional atau *black-box testing* merupakan pendekatan pengujian yang ujinya diturunkan dari spesifikasi program atau komponen. Sistem merupakan kotak hitam yang perilakunya hanya dapat ditentukan dengan mempelajari *input* dan *output* yang berkaitan. Penguji memberikan input kepada komponen atau sistem dan meneliti output yang dihasilkan, jika output bukan merupakan yang diramalkan, maka uji tersebut telah dengan berhasil mendeteksi masalah dengan perangkat lunak. Ilustrasi pengujian *black-box testing* pada sistem dapat dilihat pada Gambar

Sistem

Input yang menyebabkan perilaku menyimpang

output yang mengungkap adanya cacat

Input data uji

Output hasil uji

**Gambar 2.2 Blcak-box Testing**

* + - 1. Operasi dan pemeliharaan

Dalam tahap Operation and Maintenance ini, sistem diinstal dan mulai digunakan. Selain itu juga memperbaiki error yang tidak ditemukan pada tahap pembuatan. Dalam tahap ini juga dilakukan pengembangan sistem seperti penambahan fitur dan fungsi baru.

### 2.4.2 Pendekatan Berorientesi Objek

Pendekatan berorientasi objek merupakan pendekatan yang memodelkan informasi yang ada pada domain masalah sebagai sekumpulan objek yang memiliki atribut dan perilaku tertentu (Pressman, 2001). Pendekatan berorientasi objek memungkinkan untuk memodelkan apa yang ada di dunia nyata untuk bisa diterapkan pada pengembangan sebuah perangkat lunak. Tahapan pengembangan perangkat lunak dapat dengan pendekatan berorientasi pada objek terdiri dari *Object Oriented Analysis*, *Object Oriented Design, Object Oriented Programming dan Object Oriented Testing.*

*Object oriented analysis* merupakan sebuah proses untuk melakukan analisis terhadap kebutuhan – kebutuhan perangkat lunak dan mendefinisikan objek – objek yang ada berdasarkan domain masalah. *Object Oriented Design* merupakan cara perancangan perangkat lunak di mana komponen-komponen fundamental pada design mempresentasikan objek dengan status pribadinya sendiri dan operasi ketimbang fungsi. *Object Oriented Programming* merupakan proses untuk mengubah rancangan yang telah dibuat dalam bentuk bahasa pemrograman. Pemrograman berorientasi objek menggunakan beberapa konsep dalam proses pemrogramannya, yaitu encapsulation, information hiding, inherintance, dan polymorphism*. Object Oriented Testing* merupakan proses untuk menguji hasil implementasi yang dilakukan berdasarkan klas – klas yang ada. Pengujian dilakukan dengan cara pengujian unit, pengujian integrasi, dan pengujian validasi

### 2.4.3 Pemodelan Berorientesi Objek

Pemodelan berorientasi objek akan dilakukan dengan menggunakan Unified Modeling Language (UML), menurut Whitten & Bentley (2007:371), Unified Modeling Language (UML) versi 2.0 adalah sekumpulan konversi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek. *Unified Modeling Languange (UML)* merupakan bahasa standar yang digunakan dalam membuat rancangan perangkat lunak dan untuk menggambarkan dan membangun, dokumen artifak dari *software intensive Sistem* (Booch et al., 2005). Dengan adanya UML, diharapkan dapat mengurangi kekacauan dalam bahasa pemodelan yang selama ini terjadi dalam lingkungan industri. UML diharapkan juga dapat menjawab masalah penotasian dan mekanisme tukar menukar model yang terjadi selama ini (Satzinger et al., 2011). Pada UML biasanya memiliki beberapa diagram yang bisa didefinisikan diantaranya *use case diagram, sequence diagram* dan *class diagram*.

1. Use Case Diagram

*Use Case Diagram* merupakan rangkaian tindakan yang dilakukan oleh sistem, aktor mewakili user atau sistem lain yang berinteraksi dengan sistem yang dimodelkan (Satzinger et al., 2010). Notasi use case diagram dapat dilihat pada tabel 2.2.

**Tabel 2.2 Contoh Simbol Use Case Diagram**

|  |  |  |
| --- | --- | --- |
| **NO** | **NAMA SIMBOL** | **KETERANGAN** |
| 1. |  | mempresentasikan kerja yang dilakukan oleh aktor yang dihubungkan ke aktor melalui suatu garis dan tidak diberikan tanda panah |
| 2. |  | Merupakan representasikan aktor dalam aplikasi |
| 3. |  | Merupakan repseresentasi komunikasi antar Aktor dengan use case |
| 4. |  | Merupakan relasi tambahan dalam use case, di mana use case yang ditambahkan dapat berdiri sendiri walaupun tampa use case tambahan tersebut. |
| 5. |  | Merupakan hubungan generalisasi antar dua buah use case di mana fungsi yang satu adalah fungsi yang lebih umum dari lainnya |
| 6. |  | Merupakan relasi tambahan dalam use case di mana use case yang ditambahkan memerlukan use case lain untuk dapat menjalankan fungsinya. |
| 7. |  | Tempat untuk menaruh setiap aktivitas sistem yang sedang berjalan. |

(Sumber: Pardigmn, 2013)

1. *Sequence Diagram*

Kegunaan utama Sequence diagram pada UML adalah untuk menggambarkan interaksi antara aktor dan objek pada sebuah sistem dan juga interaksi antara objek itu sendiri (Sommerville, 2011). *Sequence diagram*, menunjukkan sequence (urutan) interaksi yang terjadi pada sebuah use case. Notasi *sequence diagram* dapat dilihat pada Tabel 2.3.

**Tabel 2.3 Contoh Simbol Sequence Diagram**

|  |  |  |
| --- | --- | --- |
| **NO** | **GAMBAR** | **KETERANGAN** |
| 1 |  | Menggambarkan pengguna yang berinteraksi dengan sistem. |
| 2 |  | Menggambarkan hubungan yang akan dilakukan. |
| 3 |  | Menggambarkan sebuah penggambaran dari *form* |
| 4 |  | Menggambarkan penghubung antara boundary dengan tabel |
| 5 |  | Menggambarkan kondisi alternatif yang akan dijalankan dalam sebuah sistem |
| 6 | *A focus of Control & A Life Line* | Menggambarkan tempat memulai dan berakhirnya pesan (message) |
| 7 |  | Menggambarkan pengiriman pesan |

(Sumber: Pardigmn, 2013)

1. *Class Diagram*

*Class diagram* digunakan ketika mengembangkan sebuah object-oriented Sistem untuk menunjukkan kelas-kelas dalam sebuah sistem beserta asosiasi antar kelas (Sommerville, 2011). Notasi *class diagram* dapat dilihat pada Tabel 2.4.

**Tabel 2.4 Contoh Notasi Class Diagram**

|  |  |  |
| --- | --- | --- |
| **NO** | **NAMA SIMBOL** | **KETERANGAN** |
| 1 |  | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 2 |  | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama. |
| 3 |  | Hubungan di mana objek anak (*descendent*) berbagi perilaku dan struktur dari objek yang ada di atasnya |
| 4 |  | Hubungan di mana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri. |
| 5 |  | Relasi antar kelas dengan makna a *whole-part* (semua-bagian) |

(Sumber: Pardigmn, 2013)

## **2.5 Teknologi Pengembangan Sistem**

Teknologi yang digunakan dalam membangun sebuah sistem PSB meliputi **Hyper*text*  Preprocessor** (PHP), Bootstrap, Cascading Style Sheets (CSS), MySQL dan Codeigniter.

### 2.5.1 Bahasa Pemrograman PHP

**PHP (akronim dari PHP: Hyper*text*  Preprocessor) adalah salah satu bahasa pemrograman yang bertujuan untuk membuat website yang dinamis ataupun aplikasi web., sedangkan HTML hanya dapat menampilkan konten bersifat statis, PHP dapat berinteraksi dengan komponen lain seperti database, *file* dan folder, sehingga PHP mampu menampilkan konten yang bersifat dinamis dari sebuah website (Yuliano, 2007). Blog, website profil sekolah, Website Social Networking, adalah beberapa contoh aplikasi web yang dapat dibuat menggunakan PHP, PHP merupakan bahasa scripting, bukan bahasa tag-based sepertihalnya HTML. PHP masuk dalam bahasa pemrograman yang *cross-platform*, artinya PHP dapat dijalankan pada sistem operasi yang berbeda-beda (Windows, Linux, ataupun Mac). Program dengan bahasa pemrograman PHP ditulis dalam *file* *text* (teks biasa) dan diakhiri “.php” (Yuliano, 2007).**

#### ****2.5.1.1**** Codeigniter (CI)

Framework codeIgniter adalah ini adalah framework yang menggunakan model MVC (model,view dan controller) untuk membangun sebuah website yang dinamis dengan menggunakan php, framework ini adalah turunan dari php jadi kodingnya tidak jauh jauh dari php yang anda kenal sebelumnya dengan model view controller ini seorang developer akan mudah dalam membangun sebuah aplikasi website dan desain tampilan yang terstuktur sehingga dengan model ini dalam maintenance website sangat mudah. CodeIgniter merupakan salah satu framework terbaik saat ini, banyak para develop web menggunakan framework ini dalam pembuatan sistem dan aplikasi mereka. Selain kemudahan dan penggunaannya, codeIgniter juga termasuk framework yang stabil dan cepat dalam penggunaannya, karena menggunakan struktur MVC dalam penggunaan framework ini tak jarang sudah banyak website dan aplikasi menggunakan ini bahkan sudah banyak komunitas codeIgniter di mana – mana.

### ****2.5.2 Bootstrap****

Bootstrap merupakan framework untuk membangun desain web secara responsif dan cepat. Artinya, tampilan web yang dibuat oleh bootstrap akan menyesuaikan ukuran layar dari browser yang kita gunakan baik di desktop, tablet ataupun mobile device. Sehingga, user akan mendapatkan pengalaman yang lebih baik dalam berselancar tanpa mempertimbangkan perangkat apa yang harus digunakan. Sejatinya, apabila kita menggunakan bootstrap, kita tinggal menggunakan nama class (untuk css) dan library (javascript) yang sudah ditentukkan oleh bootstrap tanpa perlu menulis kode dari 0 (awal) sehingga, bisa menghemat waktu dalam pengembangan website untuk urusan UI (User Interface). Bootstrap memiliki tampilan yang indah dan dapat dicustomisasi (Tutorialspoint.com, 2005).

### 2.5.3 Cascading Style Sheets (CSS)

CSS singkatan dari Cascading Style Sheets, yaitu sebuah bahasa berbasis *text*  yang digunakan untuk memformat tampilan dari halaman web yang dibuat dengan HTML, misalnya dalam hal pewarnaan, ukuran, posisi dsb. Jika HTML adalah dasar dan batu bata yang membentuk struktur rumah, CSS adalah plester dan cat yang menghiasinya. Dengan CSS, memungkinkan kita untuk memberikan efek-efek khusus pada isi halaman web secara lebih spesifik. Memungkinkan kita mengubah tampilan beberapa halaman web sekaligus dengan sedikit kode (Mukhlasin, 2013).

### ****2.5.4 Database****

Database adalah sebuah Sistem yang dibuat untuk mengorganisasi, menyimpan dan menarik data dengan mudah. Database terdiri dari kumpulan data yang terorganisir untuk 1 atau lebih penggunaan, dalam bentuk digital. Database digital di manage menggunakan Database Management Sistem (DBMS), yang menyimpan isi database, mengizinkan pembuatan dan maintenance data dan pencarian dan akses yang lain. Beberapa Database yang ada saat ini adalah : Mysql, Sql Server, Ms.Access, Oracle, dan PostgreSql (Sofwan, 2011). Namun, dalam penelitian ini akan menggunakan Mysql sebagai penyimpanan data.

#### 2.5.4.1 MySql

MySQL adalah salah satu jenis database server yang sangat terkenal. MySQL menggunakan bahasa SQL untuk mengakses database nya. Lisensi Mysql adalah FOSS License Exception dan ada juga yang versi komersial nya. Tag Mysql adalah “The World's most popular open source database”. MySQL tersedia untuk beberapa platform, diantaranya adalah untuk versi windows dan versi linux. Untuk melakukan administrasi secara lebih mudah terhadap Mysql dapat menggunakan software tertentu, di antara nya adalah phpmyadmin dan mysql yog. Pada penelitian ini peneliti akan menggunakan phpmyadmin, yang terdapat dalam bundle xampp (Sofwan, 2011).

### 2.5.5 Midtrans

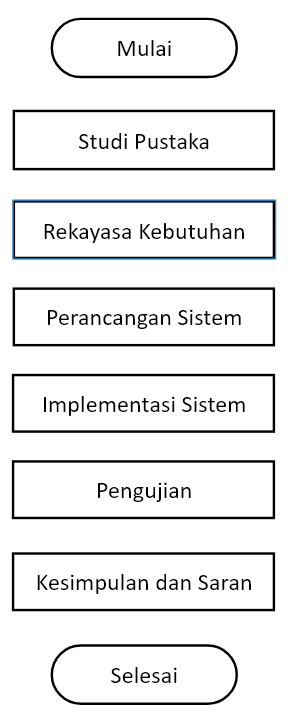
Midtrans adalah salah satu payment gateaway yang digunakan di indonesia, penggunaan midtrans sebagai payment gateaway pada toko online atau website online lainnya membuat transaksi menjadi lebih mudah dan membuat bisnis jadi lebih maju, cara kerja payment gateaway secara umum yaitu ketika pengguna telah melakukan pembelian atau terdapat tagihan pada suatu website maka data berupa pesanan dan metode pembayarannya akan diteruskan untuk diverifikasi ke bank setelah itu pengguna akan melakukan pembayaran sejumlah uang dan hingga akhirnya pengguna dan penjual akan memperoleh status dari transaksi tersebut. Keuntungan penggunaan payment gateaway pada toko online dapat menambah nilai dari sebuah toko, selain itu pada PSB sendiri dapat memudahkan dalam melakukan transaksi jarak jauh (Dewaweb, 2018).

### 2.5.6 Zenziva

Zenziva merupakan salah satu sms gateaway yang mudah dalam pengaplikasiannya, penggunaan sms gateaway dalam sebuah sistem dapat menambah dan meningkatkan kualitas dari suatu sistem. Sms gateaway biasanya digunakan pada lembaga kursus, online shop atau toko online, dll. Tanpa harus mengirim secara manual. Dengan adanya sms gateaway sangat memudahkan dalam mengirim informasi berupa jadwal, atau kode verifikasi, tausiyah, dll selain itu penggunaa sms gateaway zenziva memiliki kemudahan dalam konfigurasinya dimana pengguna tidak perlu membeli handphone atau modem untuk sms gateaway tidak perlu juga untuk membeli software sms gateaway atau bahkan install software, cukup login kedalam sistem zenziva kita sudah dapat menggunakannya untuk mengirim smsm sesuai kebutuhan kita dan bisa siakses kapan pun dan dimana pun (Zenziva, 2017).

# **METODOLOGI PENELITIAN**

Metodologi merupakan serangkaian proses yang akan dilaksanakan selama penelitian, dengan adanya tahapan metodologi, penelitian dapat dilaksanakan dengan teratur, terencana dan sistematis. Kegiatan ini akan dilaksanakan menggunakan metode *Waterfall* dan disusun dalam beberapa tahap seperti pada Gambar 3.1.



**Gambar 3.1 Diagram Alur Penelitian**

**3.1 Studi Pustaka**

Studi pustaka adalah proses pencarian terkait dasar-dasar teori yang nantinya akan digunakan selama penelitian, pencarian bersumber dari buku, jurnal, internet, dokumentasi framework serta penelitian lain yang relevan dengan penelitian ini. Teori pendukung yang menjadi bahan acuan dalam penelitian ini meliputi:

1. Kajian Pustaka
2. Sistem Informasi
3. PPDB
4. Rekayasa Perangkat Lunak

4.1 Model Pengembangan

4.2 Pendekatan Berorientasi Objek

4.3 Pemodelan Berorientasi Objek

1. Teknologi Pengembangan Sistem

5.1 Bahasa Pemrograman PHP

5.2 Bootstrap

5.3 CSS

5.4 NoSQL

5.4.1 MongoDB

5.5 Codeigniter

**3.2 Rekayasa Kebutuhan**

Rekayasa kebutuhan merupakan proses pencarian, pemodelan dan pengumpulan kabutuhan-kebutuhan yang diperlukan, tujuan dari proses ini yaitu untuk mengetahui daftar kebutuhan yang diperlukan baik fungsional maupun non fungsional. Hasil dari proses analisis kebutuhan adalah daftar kebutuhan fungsional dan non fungsional, daftar aktor yang berinteraksi dengan sistem, use case diagram dan use case scenario, mekanisme dalam proses analisis kebutuhan sebagai berikut:

1. Melakukan proses elisitasi kebutuhan, elisitasi kebutuhan dilakukan dengan cara wawancara dengan salah satu pihak sekolah yang merupakan pengurus PSB lebih tepatnya dengan Muhammad Akis.
2. Melakukan identifikasi aktor yang terlibat dengan sistem nantinya, kemudian menyusun daftar kebutuhan fungsional dan non fungsional.
3. Melakukan pemodelan kebutuhan fungsional dalam bentuk use case diagram.
4. Membuat detail langkah-langkah tiap use case dalam bentuk use case scenario.

**3.3 Perancangan Sistem**

Perancangan dilakukan dengan mengacu pada daftar kebutuhan yang telah didefinisikan pada proses analisis kebutuhan, perancangan sistem bertujuan untuk mengetahui bagaimana rancangan sistem yang akan dibuat. Pada proses ini kebutuhan sistem akan diubah dalam bentuk sequence diagram, class diagram, perancangan komponen, perancangan basisdata dan perancangan antarmuka. *Sequence diagram* menggambarkan komunikasi yang dinamis antara objek selama mengeksekusi perintah. Class diagram untuk mendefinisikan daftar class sekaligus hubungan antar class berdasarkan objek yang ada pada sequence diagram. Perancangan komponen meliputi perancang algoritma dan method yang terdapat dalam class sehingga menghasilkan pseudocode untuk tiap algoritma. Perancangan antarmuka bertujuan agar pengembang sistem dan stakeholder dapat memahami gambaran yang diberikan oleh pengembang. Perancangan basisdata digunakan untuk membentuk suatu bangunan data antara *file*-*file* yang memiliki kaitan antara satu dengan yang lain.

**3.4 Implementasi Sistem**

Implementasi sistem adalah proses mengubah hasil dari proses perancangan dalam bentuk code, tujuan dari proses ini yaitu mengembangkan sistem secara keseluruhan. Pada proses ini penulis mengimplementasikannya dengan menggunakan bahasa pemrograman Codeigniter serta basisdata MongoDB, pada tahap ini akan mengimpelementasikan antarmuka yang telah dirancang sebelumnya pada proses perancangan sistem.

**3.5 Pengujian**

Pengujian dilakukan untuk mendeteksi kesalahan pada program yang sedang dikembangkan. Pada pengembangan sistem ini, sistem diuji struktur datanya dengan menggunakan *white-box* testing dengan menggunakan pseudocode dari sistem dan flow graph untuk menguji *error* tiap jalur. Kemudian menguji tampilan sistem yang digunakan oleh pengguna atau user dengan menggunakan *black-box* testing. Sistem diuji tampilan untuk mengetahui apakah terdapat tampilan yang kurang baik, dan apakah inputan dan keluaran sesuai dengan test case yang dibuat.

**3.7 Kesimpulan dan Saran**

Tahap akhir yang dilakukan pada penelitian ini adalah menarik kesimpulan yang telah didapatkan dari hasil analisis yang diambil dari keseluruhan sistem yang telah dirancang. Selain itu, saran diberikan untuk memberi masukan atas kekurangan yang ada sebagai acuan perbaikan untuk pengembangan selanjutnya.

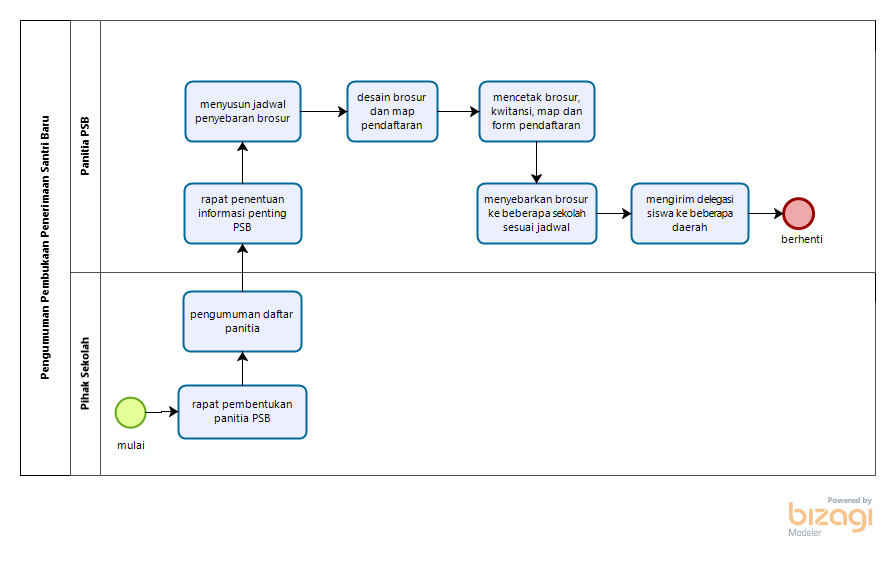
# **ANALISIS KEBUTUHAN**

Bab ini menjelaskan proses analisis kebutuhan PSB pada MAS Nurul As’adiyah Callaccu Sengkang, proses diawali dengan melakukan elisitasi kebutuhan, gambaran umum aplikasi, identifikasi aktor, spesifikasi kebutuhan dan pemodelan kebutuhan.

## **4.1 Elisitasi Kebutuhan**

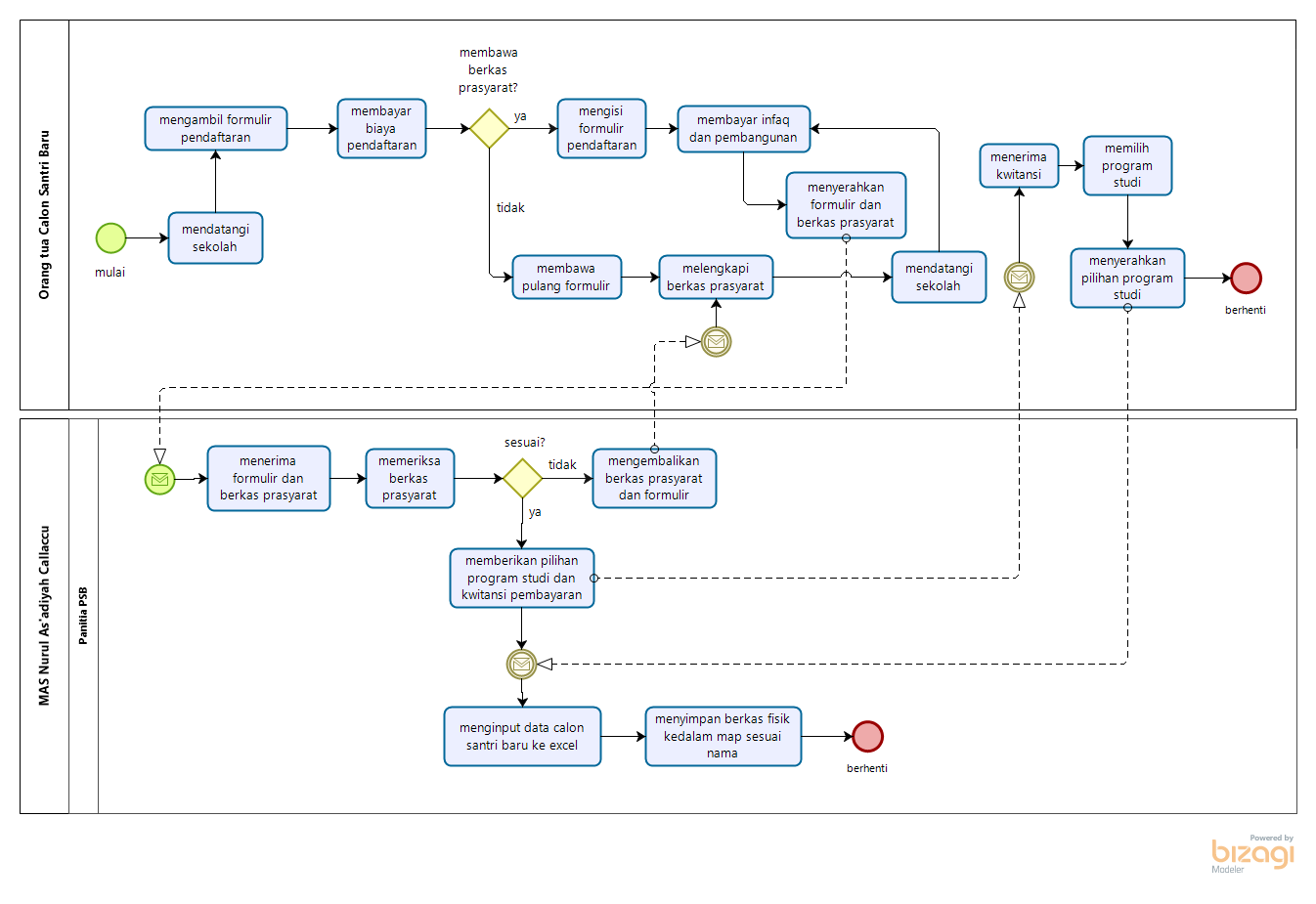
Proses elisitasi kebutuhan dilakukan untuk memperoleh daftar kebutuhan sistem serta daftar aktor yang akan berinteraksi dengan sistem, elisitasi dilakukan dengan metode wawancara terhadap salah satu pengurus PSB yaitu Bapak Muhammad Akis, wawancara dilakukan dengan topik pembahasan terkait proses bisnis penerimaan santri baru yang saat ini digunakan.

PSB pada MAS Nurul As’adiyah Callaccu memiliki lima proses tahapan diantaranya pengumuman pembukaan penerimaan santri baru, pendaftaran, seleksi penerimaan santri meliputi test tulis dan wawancara dan pengumuman. Pengumuman pembukaan penerimaan santri baru masih dilakukan dengan penyebaran brosur ke beberapa sekolah, proses pengumuman pembukaan penerimaan santri dimulai dengan pembentukan panitia PSB yang dilakukan oleh pihak sekolah. Setelah panitia PSB terbentuk, panitia mulai menentukan tanggal setiap agenda dan menyusun jadwal untuk melakukan pembagian brosur dan sosialisasi kebeberapa sekolah menengah pertama (SMP) di kota Sengkang, untuk luar daerah panitia mengirim delegasi siswa untuk melakukan penyebaran brosur disekolah mereka saat penempuh jenjang SMP. Proses pengumuman pembukaan PSB dapat dilihat pada Gambar 4.1.



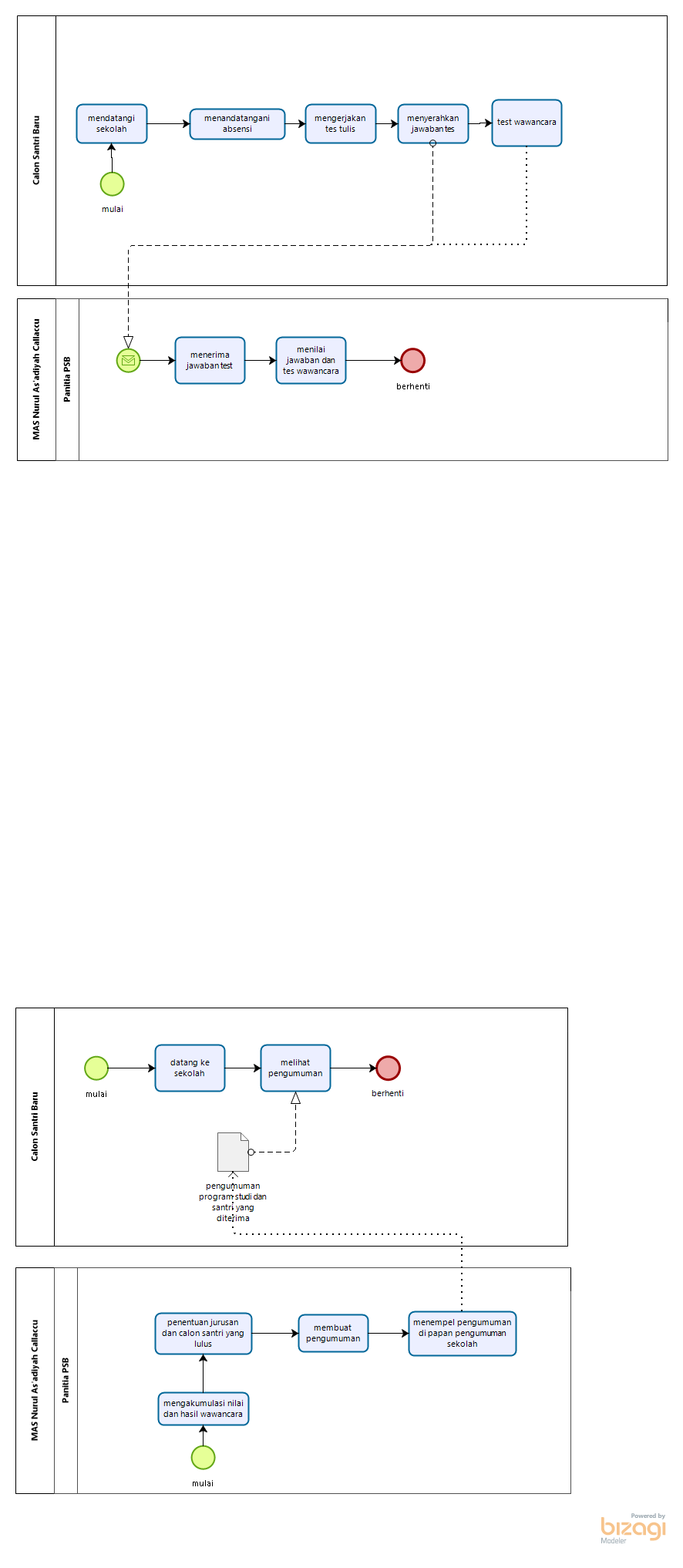
**Gambar 4.1 Pengumuman Pembukaan PSB**

Setelah pengumuman pembukaan penerimaan santri baru, dilanjutkan proses pendaftaran calon santri baru. Proses pendaftaran dilakukan dengan menggunakan media kertas dengan formulir diambil secara langsung di kantor MAS Nurul As’adiyah Callaccu pada waktu yang telah ditentukan, proses tersebut menyulitkan orang tua calon santri baru khususnya yang berada di luar daerah yaitu harus mengunjungi sekolah untuk melakukan proses pendaftaran. Proses pendaftaran dimulai ketika orang tua calon santri baru mendatangi kantor MAS Nurul As’adiyah Callaccu untuk mengambil formulir sekaligus melakukan pembayaran, kemudian formulir dibawa pulang untuk diisi sekaligus melengkapi berkas yang menjadi prasyarat pendaftaran. Panitia PSB akan mengecek kelengkapan dan kesesuain berkas yang diberikan, jika sesuai maka panitia akan memberikan pilihan program studi dan kuitansi pembayaran, setelah itu orang tua calon santri akan memilih program studi yang diinginkan dan panitia akan menginputkan keseluruhan data yang diberikan ke dalam aplikasi excel sekaligus menyimpan berkas fisik kedalam map sesuai dengan nama masing-masing calon santri baru. Penggambaran alur secara detail dapat dilihat pada Gambar 4.2.



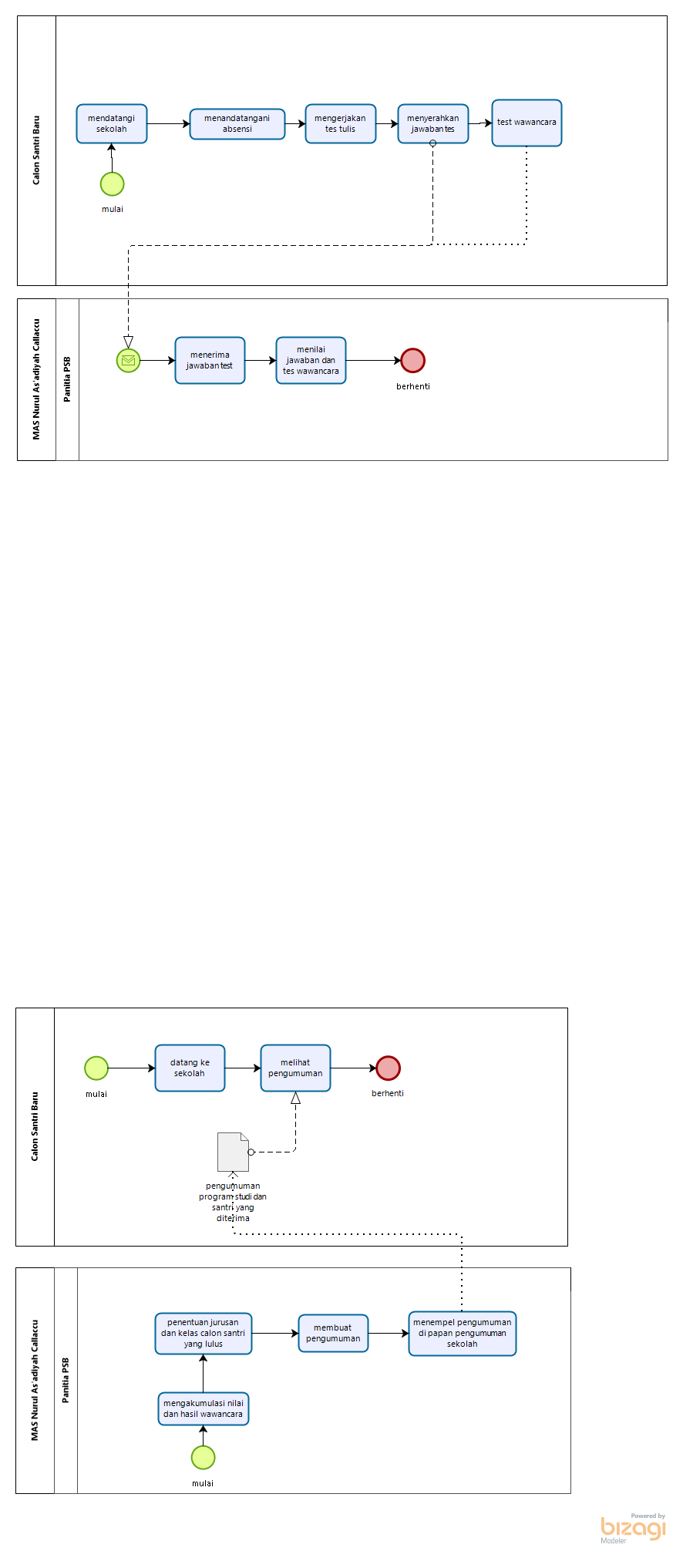
**Gambar 4.2 Alur Pendaftaran**

Setelah melakukan proses pendaftaran, calon santri akan melanjut ke proses selanjutnya yaitu proses tes seleksi yang terdiri dari tes tulis dan wawancara, proses seleksi dilakukan selama 2 hari, hari pertama untuk calon santri yang telah mengembalikan formulir dan hari kedua bagi calon santri yang terlambat mengembalikan formulir, jika selama dua hari tersebut masih terdapat calon santri yang belum melakukan pengembalian formulir maka tes seleksi akan langsung dilakukan saat mereka melakukan pengembalian formulir. Proses seleksi dimulai saat calon santri datang ke sekolah dan menandatangani absensi sesuai nama, calon santri mulai untuk mengerjakan tes tulis yang terdiri dari tes potensi dan minat santri, tes bahasa inggris dan tes bahasa arab, setelah mengerjakan jawaban diberikan ke panitia PSB, setelah itu dilanjutkan dengan tes wawancara yang membahas wawasan ke-islaman dan tes membaca Al-qur’an, setelah semua telah dilaksanakan panitia PSB akan menilai keseluruhan tes. Penggambaran alur secara detail dapat dilihat pada Gambar 4.3.



**Gambar 4.3 Seleksi Penerimaan**

Setelah tes seleksi yaitu tes tulis dan wawancara telah dilakukan, panitia PSB akan mengakumulasi total nilai keseluruhan tes untuk mengetahui siswa yang bersangkut dapat dikatakan lulus atau tidak. Selain itu dengan mengacu ada nilai tes seleksi, panitia akan menentukan program studi sekaligus kelas yang akan dimasuki oleh masing-masing siswa, untuk program studi sendiri dapat berganti apabila selama sebulan percobaan dirasa ingin berganti program studi. Setelah hasil tes dan program studi telah ditentukan maka panitia akan membuat pengumuman dan pengumuman tersebut akan ditempelkan pada papan pengumuman sekolah sesuai dengan jadwal yang telah ditentukan. Pada hari pengumuman orangtua calon santri berkunjung ke sekolah untuk melihat hasil dari proses penerimaan santri baru, pada MAS Nurul As’adiyah Callacu tidak terdapat proses pendaftaran ulang dikarenakan proses pendaftaran sudah sekaligus dianggap sebagai pendaftaran ulang, akan tetapi terdapat proses pengenalan lingkup madrasah jika calon santri telah diterima atau lulus pada proses penerimaan santri baru. Penggambaran alur secara detail dapat dilihat pada Gambar 4.4.



**Gambar 4.4 Pengumuman**

## **4.2 Deskripsi Umum Sistem**

Sistem informasi Penerimaan Santri Baru (PSB) adalah sistem perangkat lunak yang bertujuan untuk memberikan kemudahan dalam proses penerimaan santri baru pada MAS Nurul As’adiyah Callaccu, dari pengumuman pembukaan penerimaan santri baru hingga pengumuman. Sistem informasi ini akan dikembangkan dengan berbasis web sehingga memberikan akses yang luas pada masyarakat dalam memperoleh informasi.

Sistem akan menyediakan fungsi *login* untuk pengguna agar dapat memperoleh hak akses sesuai dengan fungsinya masing-masing, terdapat 3 pengguna yang akan memiliki hak akses setelah melakukan proses *login* diantaranya peserta PSB, panitia PSB dan admin. Peserta PSB merupakan calon santri yang akan melakukan pendaftaran, sehingga akan memiliki beberapa fungsi yaitu mengisi *form* data diri, orang tua dan wali, mengubah data diri/orang tua/wali, mengunduh kartu peserta, calon santri juga dapat mengerjakan test seleksi online, *upload* berkas prasyarat dan melakukan pembayaran. Selain itu, calon santri dapat mengakses pengumuman pada jadwal yang telah ditentukan dengan memasukkan Nomor Induk Nasional (NISN)/Nomor pendaftaran pada menu utama web PSB.

Setelah peserta PSB selesai melakukan pendaftaran maka panitia PSB akan mengecek berkas prasyarat dan data diri untuk diverifikasi jika dirasa telah lengkap. Panitia PSB merupakan panitia yang mengelola data peseta PSB dan mengatur seluruh kegiatan yang berhubungan dengan proses penerimaan santri baru, panitia PSB memiliki fungsi untuk dapat memverifikasi berkas peserta PSB, mengelola soal tes seleksi online, mengelola nilai tes seleksi (akademik dan wawancara), mengelola pengumuman calon santri yang diterima, mengelola data arsip, mengelola pembayaran dan mencetak nama-nama santri baru yang diterima ataupun yang mendaftar untuk menjadi laporan kepada kepala sekolah.

Dalam mengelola seluruh informasi dalam sistem informasi PSB merupakan tugas dari admin, admin merupakan petugas tata usaha yang mengelola informasi yang akan disampaikan melalui sistem informasi PSB. Admin memiliki fungsi dapat menambah pengguna, pengguna yang dimaksud adalah panitia PSB, admin juga dapat mengelola masukan dari pengunjung sistem, melihat dan mengubah profil sekolah dan mengelola informasi terkait penerimaan santri baru.

## **4.3 Identifikasi Aktor**

Identifikasi aktor diperoleh dari hasil wawancara dengan panitia PSB, di mana dalam proses PSB saat ini melibatkan panitia PSB dan peserta PSB. identifikasi aktor ini bertujuan untuk mengetahui aktor-aktor yang nantinya akan berinteraksi dengan sistem PSB, setelah wawancara dilakukan maka diperoleh 6 aktor diantaranya admin, panitia, peserta, pengunjung, sms gateaway dan payment gateaway. Tabel 4.1 menjelaskan aktor-aktor yang akan berinteraksi dengan sistem beserta deskripsinya.

**Tabel 4.1 Identifikasi Aktor**

|  |  |  |
| --- | --- | --- |
| No | Aktor | Deskripsi |
| 1 | Admin | Petugas tata usaha yang bertugas mengolah data pokok siswa dan informasi yang akan disampaikan melalui sistem. |
| 2 | Panitia | Panitia yang bertugas untuk mengelola administrasi peserta PSB. |
| 3 | Peserta | Pengguna yang telah terdaftar dalam sistem dan merupakan calon santri aktif. |
| 4 | Pengunjung | Pengguna yang belum melalui autentikasi sistem. |
| 5 | SMS Gateaway | Aktor sekunder untuk mengirim sms kepada aktor peserta dan panitia. |
| 6 | Payment Gateaway | Aktor sekunder untuk melakukan update status pembayaran dan bayar tagihan. |

## **4.4 Spesifikasi Kebutuhan**

Spesifikasi kebutuhan dalam sistem PSB akan menggunakan penomoran PSB\_F\_XX\_YY dengan penjelasan PSB berasal dari nama sistem yaitu Penerimaan Santri Baru (PSB), kemudian dilanjutkan dengan kode kebutuhan yaitu fungsional (F) atau non-fungsional (NF), kemudian (XX) merupakan kode kebutuhan dan (YY) merupakan kode spesifikasi kebutuhan, dapat dilihat pada

**Gambar 4.5 Aturan Penomoran**

**PSB\_F\_XX\_YY**

Nomor urut spesifikasi kebutuhan

Nomor urut representasi kebutuhan

Kode representasi kebutuhan:

F: kebutuhan fungsional

NF: kebutuhan non-fungsional

Singkatan dari nama sistem Penerimaan Santri Baru (PSB)

### 4.4.1 Kebutuhan Fungsional

Kebutuhan fungsional pada sistem informasi PSB didapatkan pada proses analisis kebutuhan dan dijelaskan pada Tabel 4.2 beserta deskripsinya. Tabel kebutuhan meliputi kode kebutuhan, nama kebutuhan fungsional, deskripsi kebutuhan dan aktor yang akan menjalankan fungsional tersebut.

**Tabel 4.2** **Kebutuhan Fungsional**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NO** | **Kode** | **Spesifikasi kebutuhan** | **Nama *Use Case*** | **Aktor** |
|  | PSB\_F\_01\_00 | Sistem menyediakan halaman autentikasi untuk pengunjung sebagai, admin, panitia atau peserta PSB. | *Login* | Pengunjung |
|  | PSB\_F\_01\_01 | Sistem menyediakan *form* untuk dapat memberikan hak akases sebagai peserta atau panitia yang terdiri dari data inputan Nomor Induk Nasional Siswa (NISN)/Nomor Induk Pegawai (NIP) dan *password* dan tombol masuk. |
|  | PSB\_F\_01\_02 | Sistem menyediakan fungsi ganti *password*. |
|  | PSB\_F\_01\_03 | Sistem menyediakan fungsi buat akun. |  |  |
|  |  | Sistem menyediakan halaman untuk mengisi data pendaftaran. | Isi data pendaftaran | Peserta |
|  |  | Sistem dapat menampilkan form pendaftaran dengan data inputan nama, no.tlp, nisn, jenis kelamin, email, alamat, tempat lahir, tanggal lahir, anak ke, jumlah saudara, alat transportasi, asal sekolah, alamat asal sekolah, alamat sekolah, nomor peserta UN, prodi pilihan, tinggal bersama, size baju, nama ayah, no.tlp ayah, masih hidup/tidak, email ayah, alamat ayah, pekerjaan ayah, penghasilan ayah, pendidikan ayah, nama ibu, no.tlp ibu, masih hiupu/tidak, email ibu, alamat ibu, pekerjaan ibu, penghasilan ibu, pendidikan ibu, rata-rata penghasilan, nama wali, no.tlp wali, hubungan keluarga, email wali, alamat wali dan pekerjaan wali. |  |  |
|  |  | Semua data harus diisi. |  |  |
|  | PSB\_F\_02\_00 | Sistem menyediakan halaman untuk dapat membuat akun sebagai peserta. | Buat Akun | Pengunjung |
|  | PSB\_F\_02\_01 | Sistem menyediakan *form* untuk membuat akun yang berisi data inputan NISN, nama lengkap, *email*, *password* dan *re-password* dan tombol buat akun. |
|  | PSB\_F\_02\_02 | NISN yang dimasukkan harus sejumlah 10 digit dan belum terdaftar pada sistem. |
|  | PSB\_F\_02\_03 | Email yang didaftarkan belum terdaftar dalam Sistem. |
|  | PSB\_F\_03\_00 | Sistem menyediakan *form* *reset* *password* yang berisi data inputan email. | Reset *password* | Pengunjung |
|  | PSB\_F\_03\_01 | Email yang diinputkan telah terdaftar dalam sistem. |
|  | PSB\_F\_03\_02 | Sistem menyediakan fungsi kirim email. |
|  | PSB\_F\_03\_03 | Sistem menyediakan form untuk mengganti password dengan data inputan password baru dan re-password baru. |  |  |
|  | PSB\_F\_04\_00 | Sistem menyediakan fasilitas *logout* untuk menghilangkan hak akses bagi pengguna sistem. | *Logout* | Admin, panitia dan peserta |
|  |  | Sistem menampilkan pesan konfirmasi sebelum keluar dari sistem. |
|  |  | Sistem menyediakan halaman untuk dapat melihat jadwal pelaksanaan PSB dalam bentuk tabel. | Lihat Jadwal kegiatan PSB | Admin |
|  |  | Data yang ditampilkan diantaranya nomor, kegiatan, tanggal mulai, tanggal berakhir, keterangan dan aksi. |
|  |  | Sistem menyediakan fungsi tambah, edit dan hapus kegiatan. |
|  |  | Sistem dapat menampilkan modal *form* untuk menambah kegiatan baru. | Tambah kegiatan | Admin |
|  |  | Data inputan terdiri dari nama kegiatan, tanggal mulai, tanggal berakhir dan keterangan. |
|  |  | Sistem menyediakan tombol tambah untuk menambah kegiatan. |
|  |  | Sistem dapat menampilkan modal form untuk melakukan perubahan pada kegiatan tertentu. | Edit kegiatan | Admin |
|  |  | Semua data inputan dapat diubah. |
|  |  | Sistem menyediakan tombol simpan untuk menyimpan perubahan. |
|  |  | Sistem dapat menghapus kegiatan tertentu dalam daftar kegiatan. | Hapus kegiatan | Admin |
|  |  | Sistem dapat menampilkan pesan konfirmasi sebelum menghapus data kegiatan. |
|  |  | Sistem menyediakan halaman untuk dapat melihat daftar syarat dan ketentuan PSB dalam bentuk tabel. | Lihat syarat dan ketentuan | Admin |
|  |  | Data yang ditampilkan diantaranya nomor, berkas prasyarat, ketentuan dan aksi. |
|  |  | Sistem menyediakan fungsi tambah, edit dan hapus syarat dan ketentuan. |
|  |  | Sistem dapat menampilkan modal *form* untuk menambah syarat baru. | Tambah syarat dan ketentuan | Admin |
|  |  | Data inputan terdiri dari nama berkas dan ketentuan. |
|  |  | Sistem menyediakan tombol tambah untuk menambah syarat dan ketentuan. |
|  |  | Sistem menampilkan modal form untuk melakukan perubahan syarat tertentu. | Edit syarat dan ketentuan | Admin |
|  |  | Semua data inputan dapat diubah. |  |  |
|  |  | Sistem menyediaakan tombol simpan untuk menyimpan perubahan. |
|  |  | Sistem dapat menghapus data syarat tertentu. | Hapus syarat dan ketentuan | Admin |
|  |  | Sistem dapat menampilkan pesan konfirmasi sebelum menghapus data syarat. |
|  |  | Sistem dapat menyediakan halaman untuk dapat melihat daftar biaya atau pembayaran dalam bentuk tabel. | Lihat daftar biaya | Admin |
|  |  | Data yang ditampilkan diantaranya nomor, pembayaran, nominal, keterangan dan aksi. |
|  |  | Sistem menyediakan fungsi tambah, edit dan hapus biaya atau pembayaran. |
|  |  | Sistem dapat menampilkan modal form untuk menambah pembayaran baru. | Tambah biaya | Admin |
|  |  | Data inputan terdiri dari nama pembayaran, jumlah dan keterangan. |
|  |  | Sistem menyediakan tombol tambah untuk menambah biaya. |
|  |  | Sistem dapat menampilkan modal form untuk melakukan perubahan biaya. | Edit biaya | Admin |
|  |  | Semua data inputan dapat diubah. |
|  |  | Sistem menyediakan tombol simpan untuk menyimpan perubahan data biaya. |
|  |  | Sistem dapat menghapus biaya tertentu. | Hapus biaya | Admin |
|  |  | Sistem dapat menampilkan pesan konfirmasi sebelum meng hapus data pembayaran. |
|  |  | Sistem menyediakan halaman untuk dapat melihat daftar file panduan dalam bentuk tabel. | Lihat daftar panduan | Admin |
|  |  | Data yang ditampilkan diantaranya nomor, nama file, keterangan, file dan aksi. |  |  |
|  |  | Sistem menyediakan fungsi lihat dan hapus file panduan. |  |  |
|  |  | Sistem menyediakan form unggah file panduan. | Unggah panduan | Admin |
|  |  | Data inputan terdiri dari nama berkas, keterangan dan tombol pilih file. |
|  |  | File yang diupload harus berformat pdf. |
|  |  | Sistem dapat menampilkan isi file yang telah diunggah. | Lihat panduan | Admin |
|  |  | Sistem menyediakan fungsi untuk mengunduh file panduan. | Unduh panduan | Pengunjung |
|  |  | Sistem dapat menghapus file tertentu. | Hapus panduan | Admin |
|  |  | Sistem menampilkan pesan konfirmasi sebelum menghapus file panduan. |
|  |  | Sistem menyediakan halaman untuk menampilkan informasi profil sekolah. | Lihat profil sekolah | Admin |
|  |  | Sistem menyediakan fungsi tambah, edit dan hapus data profil. |  |  |
|  |  | Sistem dapat menampilkan modal *form* untuk menambah data profil sekolah. | Tambah profil sekolah | Admin |
|  |  | Data inputan terdiri dari jenis informasi dan deskripsi. |
|  |  | Sistem menyediakan tombol tambah untuk menambah profil sekolah. |
|  |  | Sistem dapat menampilkan modal form untuk melakukan perubahan data profil sekolah. | Edit profil sekolah | Admin |
|  |  | Semua data inputan dapat diubah. |
|  |  | Sistem menyediakan tombol simpan untuk menyimpan perubahan data. |
|  |  | Sistem dapat melakukan hapus data tertentu. | Hapus profil sekolah | Admin |
|  |  | Sistem dapat menampilkan pesan konfirmasi sebelum menghapus data profil sekolah. |
|  |  | Sistem menyediakan halaman untuk dapat melihat daftar berita. | Lihat daftar berita | Admin |
|  |  | Untuk admin berita ditampilkan dalam bentuk tabel, data yang ditampilkan diantaranya nomor, judul, penullis, tanggal publish, foto dan aksi. |
|  |  | Untuk admin, sistem menyediakan fungsi tambah, edit dan hapus berita. |
|  |  | Sistem menyediakan fungsi pencarian berita tertentu. |
|  |  | Untuk pengunjung, sistem menyediakan fungsi baca berita. |
|  |  | Sistem menyediakan halaman untuk membaca berita dengan lengkap. | Baca berita | Pengunjung |
|  |  | Sistem menampilkan foto berita, isi berita, penulis dan tanggal publish. |  |  |
|  |  | Sistem menampilkan link daftar berita terbaru. |
|  |  | Sistem menyediakan halaman tambah berita baru. | Tambah berita | Admin |
|  |  | Sistem dapat menampilkan form tambah berita yang terdiri dari data judul, penulis, tanggal publish, deskripsi dan foto. |
|  |  | Sistem menyediakan tombol upload foto berita. |
|  |  | Semua data harus diisi. |
|  |  | Sistem menyediakan tombol kembali untuk ke halaman daftar berita. |
|  |  | Sistem menyediakan halaman untuk dapat mengubah berita. | Edit berita | Admin |
|  |  | Sistem dapat menampilkan form edit berita yang terdiri dari data judul, penulis, tanggal, deskripsi dan foto |
|  |  | Semua data dapat diubah. |  |  |
|  |  | Sistem menyediakan tombol untuk menyimpan perubahan.. |
|  |  | Sistem dapat melakukan hapus berita tertentu. | Hapus berita | Admin |
|  |  | Sistem dapat menampilkan pesan konfirmasi sebelum melakukan hapus berita. |
|  |  | Sistem menampilkan halaman daftar pengguna yaitu peserta dan admin. | Lihat daftar pengguna | Admin |
|  |  | Untuk data pengguna admin, data ditamplikan dalam bentuk tabel dengan data diantaranya nomor, nama, NIP, email, nomor handphone, role dan aksi. |
|  |  | Untuk data pengguna peserta, data ditamplikan dalam bentuk tabel dengan data diantaranya nomor, nama, kode daftar, email, role dan aksi. |
|  |  | Sistem menampilkan total peserta dan admin. |
|  |  | Sistem menyediakan fungsi tambah, edit, arsip dan hapus pengguna. |
|  |  | Sistem dapat menampilkan modal form untuk menambah pengguna admin. | Tambah admin | Admin |
|  |  | Data inputan terdiri dari NIP, nama, email dan role. |
|  |  | Sistem menyediakan tombol tambah untuk menambah pengguna baru. |
|  |  | Sistem dapat mengirim password akun melalui *whatsApp*. |
|  |  | Sistem menampilkan modal form untuk mengubah data. | Edit admin | Admin |
|  |  | Data yang ditampilkan meliputi NIP, nama, email, role dan password. |
|  |  | Data password tidak dapat diubah. |
|  |  | Sistem menyediakan tombol simpan untuk menyimpan perubahan data. |
|  |  | Sistem dapat melakukan hapus pengguna tertentu. | Hapus admin | Admin |
|  |  | Sistem menampilkan pesan konfirmasi sebelum menghapus pengguna. |
|  |  | Sistem dapat menyimpan data peserta sebagai data arsip. | Arsip peserta | Admin |
|  |  | Sistem menampilkan pesan konfirmasi sebelum mengarsip peserta. |  |  |
|  |  | Sistem menyediakan halaman untuk melihat data yang telah di arsip. | Lihat daftar data arsip | Admin |
|  |  | Data ditampilkan dalam bentuk tabel, diantaranya nomor, nama, kode daftar, email, tanggal daftar dan aksi. |
|  |  | Sistem menyediakan fungsi lihat detail data. |
|  |  | Sistem menyediakan fungsi pencarian data arsip. |
|  |  | Sistem menyediakan fungsi export data kedalam bentuk excel. |
|  |  | Sistem menyediakan halaman detail data arsip. | Lihat detail data arsip | Admin |
|  |  | Data yang ditampilkan diantaranya data diri, orangtua, wali dan berkas pendaftaran. |
|  |  | Sistem menyediakan fungsi kembali ke halaman daftar data arsip. |
|  |  | Sistem menydeiakan halaman untuk menampilkan daftar peserta PSB yang terlah melakukan registrasi. | Lihat daftar calon santri | Panitia |
|  |  | Data ditampilkan dalam bentuk tabel diantaranya nomor, kode daftar, nama, nisn, asal sekolah, status verifikasi dan tombol detail. |
|  |  | Sistem menyediakan fungsi pencarian data. |  |  |
|  |  | Sistem menyediakan fungsi detail untuk melihat detail data calon santri. |
|  |  | Sistem menyediakan fungsi export data dalam bentuk pdf dan excel. |
|  |  | Sistem menyediakan fungsi cetak daftar nama calon santri. | Cetak daftar calon santri | Panitia |
|  |  | Sistem menyediakan halaman untuk melihat data calon santri secara lengkap | Lihat detail data calon santri | Panitia |
|  |  | Data yang ditampilkan diantaranya data diri, orang tua, wali dan berkas prasyarat. |
|  |  | Sistem menyediakan fungsi verifikasi data. |
|  |  | Sistem menyediakan fungsi lihat pada berkas yang diupload. |
|  |  | Sistem menyediakan fungsi terima dan tolak pada berkas yang diupload. |
|  |  | Sistem dapat menampilkan modal form untuk melakukan verifikasi data. | Verifikasi data | Panitia |
|  |  | Data yang ditampilkan diantaranya nama, kode daftar, nomor telepon, status verifikasi dan pesan. |  |  |
|  |  | Data nama, kode daftar dan norom telepon hanya dapat dibaca. |
|  |  | Sistem menyediakan fungsi kirim pesan*.* |
|  |  | Sistem dapat menampilkan status verifikasi. |
|  |  | Sistem dapat menampilkan isi berkas yang diupload. | Lihat berkas | Panitia dan peserta |
|  |  |  | Lihat daftar pembayaran | Panitia |
|  |  | Sistem menyediakan halaman untuk menampilkan daftar soal tes akademik | Lihat daftar soal tes akademik | Panitia |
|  |  | Data ditampilkan dalam bentuk tabel diantaranya nomor, soal, mapel, kunci jawaban dan aksi. |
|  |  | Sistem menyediakan funsi pencarian data. |
|  |  | Sistem menyediakan fungsi export data dalam bentuk PDF dan excel. |
|  |  | Sistem menyediakan fungsi cetak daftar soal tes akademik. |
|  |  | Sistem menyediakan fungsi tambah, edit dan hapus data soal. |
|  |  | Sistem dapat menampilkan halaman tambah soal. | Tambah soal | Panitia |
|  |  | Sistem menyediakan form untuk menambah soal tes dengan data inputan diantaranya mapel, teks soal, opsi a, opsi b, opsi c, opsi d, kunci jawaban dan bobot nilai. |
|  |  | Semua data input harus diisi. |  |  |
|  |  | Sistem menyediakan fungsi untuk kembali ke halaman sebelumnya. |
|  |  | Sistem menyediakan tombol tambah untuk menambah soal baru. |
|  |  | Sistem menyediakan halaman untuk dapat melakukan edit data soal. | Edit soal | Panitia |
|  |  | Semua data input dapat diubah. |
|  |  | Sistem menyediakan tombol simpan untuk menyimpan perubahan data soal. |
|  |  | Sistem dapat menghapus soal tertentun dari daftar soal. | Hapus soal | Panitia |
|  |  | Sistem menampilkan pesan konfirmasi sebelum menghapus soal. |
|  |  | Sistem menyadiakan halaman untuk melihat daftar tes seleksi. | Lihat daftar tes | Panitia |
|  |  | Data ditampilkan dalam bentuk tabel dengan data tampilan nomor, nama tes, jumlah soal, tanggal mulai, durasi, status dan aksi. |
|  |  | Sistem menyediakan fungsi pencarian data. |  |  |
|  |  | Sistem menyediakan fungsi tambah tes seleksi. |
|  |  | Sistem menyediakan fungsi edit, hapus dan nonaktif tes seleksi. |
|  |  | Hanya boleh ada satu tes yang berstatus aktif. |  |  |
|  |  | Sistem menyediakan halaman untuk dapat membuat tes seleksi. | Buat tes | Panitia |
|  |  | Sistem menyediakan form membuat tes, dengan data inputan nama tes, jumlah soal, tanggal mulai, tanggal berakhir dan durasi. |
|  |  | Sistem menampilkan data pengguna pembuat tes diantaranya nama dan nip. |
|  |  | Sistem menampilkan petunjuk pengisian form. |
| 1. j |  | Jumlah soal tidak boleh melebihi jumlah soal di daftar soal. |
|  |  | Sistem menyediakan tombol tambah untuk menambah tes baru. |
|  |  | Sistem meyediakan form untuk mengubah tes seleksi. | Edit tes | Panitia |
|  |  | Sistem menampilkan petunjuk pengisian form edit. |
|  |  | Sistem menampilkan data pengguna pembuat tes diantaranya nama dan nip. |  |  |
|  |  | Semua data input dapat diperbaharui. |
|  |  | Sistem menyediakan tombol tambah simpan untuk menyimpan perubahan data. |
|  |  | Sistem dapat menghapus tes seleksi tertentu dari daftar tes seleksi. | Hapus tes | Panitia |
|  |  | Sistem menampilkan pesan konfirmasi sebelum menghapus tes seleksi. |
|  |  | Sistem dapat mengubah status tes dari aktif menjadi nonaktif. | Nonaktif tes | Panitia |
|  |  | Sistem menyediakan halaman untuk dapat melihat hasil tes peserta PSB. | Lihat hasil tes seleksi | Panitia |
|  |  | Data ditampilkan dalam bentuk tabel diantaranya nama, nilai dan aksi. |
|  |  | Sistem menyediakan fungsi hapus dan lihat detail jawaban. |
|  |  | Sistem menyediakan halaman untuk melihat detail jawaban dari peserta. | Lihat detail jawaban | Panitia |
|  |  | Data yang ditampilkan diantaranya jumlah benar, jumlah salah dan jawaban peserta disetiap nomor beserta kunci jawaban. |
|  |  | Sistem dapat menghapus hasil tes seleksi peserta. | Hapus hasil tes seleksi | Panitia |
|  |  | Sistem menampilkan pesan konfirmasi sebelum menghapus hasil tes. |
|  |  | Sistem menyediakan halaman untuk menilai tes wawancara peserta. | Nilai tes wawancara | Panitia |
|  |  | Sistem menyediakan fungsi pencarian data. |
|  |  | Sistem menyediakan tombol nilai untuk input nilai tes wawancara. |
|  |  | Sistem menampilkan modal form dengan data tampilan nama, kode daftar dan nilai tes wawancara. |
|  |  | Data nama dan kode daftar hanya dapat dibaca. |
|  |  | Sistem menyediakan tombol simpan untuk menyimpan nila tes wawancara. |
|  |  | Sistem menyediakan halaman untuk melihat hasil akumulasi nila tes. | Nilai kelulusan | Panitia |
|  |  | Data ditampilkan dalam bentuk tabel diantaranya nomor, kode daftar, nama, prodi pilihan, tes akademik, tes wawancara, total nilai, status penerimaan dan aksi. |
|  |  | Sistem menyediakan tombol lulus dan tidak lulus untuk menentukan hasil akhir penerimaan/status penerimaan. |  |  |
|  |  | Sistem menyediakan fungsi pencarian data. |
|  |  | Sistem dapat mengirim pesan notifikasi status pendaftaran. | Kirim pesan | Sms gateaway |
|  |  | Sistem dapat mengirim password akun panitia. |
|  |  | Sistem dapat mengirim |
|  |  | Sistem dapat menampilkan pesan pemberitahuan hasil penerimaan pada halaman dashboard peserta. | Lihat pengumuma | Peserta |
|  |  | Sistem menyediakan halaman untuk melakukan perubahan data akun. | Pengaturan akun | Peserta |
|  |  | Sistem menampilkan form pengaturan akun dengan data inputan nomor pendaftaran, NISN, nama lengkap dan email. |
|  |  | Sistem menyediakan tombol simpan untuk menyimpan perubahan akun. |
|  |  | Data yang dapat diubah hanya nama dan email. |  |  |
|  |  | Sistem menyediakan fungsi mencetak kartu peserta. | Cetak kartu peserta | Peserta |
|  |  | Sistem menyediakan halaman untuk dapat melakukan perubahan foto, data diri, orangtua dan wali. | Edit data | Peserta |
|  |  | Nomor pendaftaran tidak dapat diubah. |
|  |  | Semua form input wajib diisi. |
|  |  | Sistem menyediakan tombol pilih file untuk mengubah file foto. |
|  |  | Sistem menyediakan tombol simpan untuk menyimpan perubahan data. |
|  |  | Sistem menyediakan tombol kembali ke halaman data diri. |
|  |  | Sistem menyediakan halaman untuk melakukan upload berkas. | Upload berkas | Peserta |
|  |  | Sistem dapat menampilkan daftar berkas prasyarat dan ketentuannya. |
|  |  | Sistem dapat melakukan upload banyak berkas sekaligus. |  |  |
|  |  | Sistem dapat menampilkan daftar berkas yang telah diupload. |
|  |  | Daftar berkas ditampilkan dalam bentuk tabel dengan data tampilan nama file, status dan aksi. |
|  |  | Sistem menyediakan fungsi lihat dan hapus berkas. |
|  |  | Sistem dapat menghapus berkas tertentu dari daftar berkas. | Hapus berkas | Peserta |
|  |  | Sistem menampilkan pesan konfirmasi sebelum menghapus berkas. |
|  |  | Sistem menyediakan halaman untuk melakukan pembayaran. | Bayar tagihan | Peserta dan payment gateaway |
|  |  | Sistem dapat menampilkan daftar tagihan dan keterangan sesuai jenis kelamin peserta. |
|  |  | Sistem dapat menampilkan riwayat pembayaran dalam bentuk tabel dengan data tampilan tanggal, virtual account, media, total pembayaran dan status pembayaran. |
|  |  | Sistem dapat menampilkan modal form input total pembayaran yang ingin dibayar. |  |  |
|  |  | Sistem dapat menyediakan pembayaran melalui antarbank dan indomaret. |
|  | PSB\_F\_04\_00 | Sistem dapat mengupdate status pembayaran peserta. | Update status pembayaran | Payment gateaway |
|  | PSB\_F\_04\_01 | Status yang ditampilkan diantaranya proses, expired dan selesai. |  |  |
|  |  | Sistem menyediakan halaman tes seleksi. | Lihat tes seleksi | Peserta |
|  |  | Sistem dapat menampilkan infomasi syarat untuk mengikuti tes seleksi. |
|  |  | Sistem dapat menampilkan informasi akun diantaranya nama, kode pendaftaran, email dan prodi pilihan. |
|  |  | Sistem dapat menampilkan informasi tes diantaranya nama tes, jumlah soal, durasi, tanggal mulai dan tanggal berakhir. |
|  |  | Sistem menyediakan fungsi mulai tes. |
|  |  | Sistem dapat menampilkan hasil tes diantaranya nilai, jumlah salah dan jumlah benar jika telah mengikuti tes seleksi. |
|  |  | Sistem menyediakan halamam untuk mengerjakan tes seleksi. | Kerjakan tes seleksi | Peserta |
|  |  | Sistem dapat menampilkan informasi data diri diantaranya nama, ko datar dan prodi pilihan. |
|  |  | Sistem dapat menampilkan soal, pilihan jawaban dan countdown. |
|  |  | Sistem dapat menampilkan daftar soal dalam bentuk tombol dengan angka. |

### 4.4.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan daftar batasan layanan yang akan diberikan sistem atau yang harus dipenuhi oleh sistem, kebutuhan non-fungsional pada sistem PSB yang dijelaskan secara detail pada Tabel 4.3.

**Tabel 4.3 Kebutuhan Non-Fungsional**

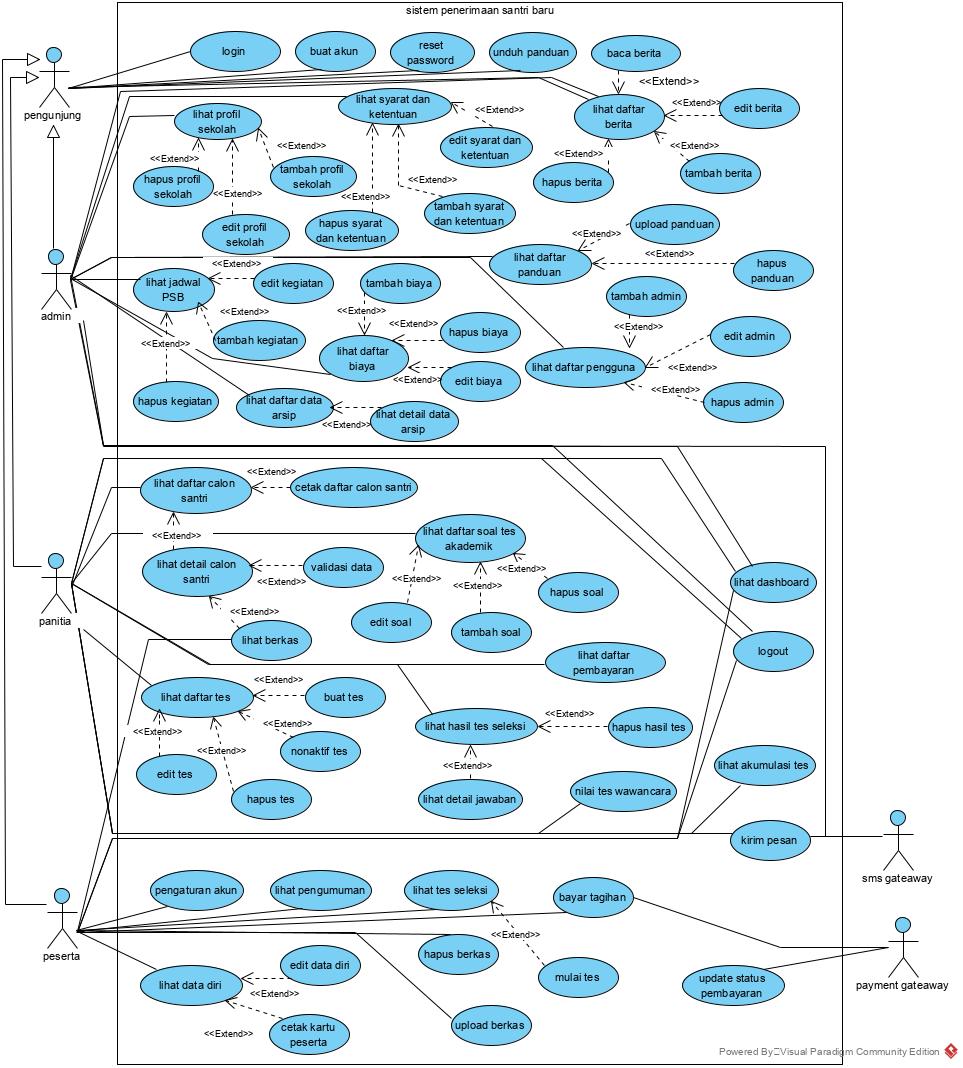
|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **Kode** | **Parameter** | **Deskripsi** |
| 1 | PSB\_NF\_01 | *Compatibility* | Sistem dapat dijalankan diberbagai web pencarian diantaranya chrome, internet explore. |

## **4.5 Pemodelan Kebutuhan**

Kumpulan kebutuhan fungsional pada sistem PSB yang diperoleh dari spesifikasi kebutuhan selanjutnya digambarkan dengan use case diagram dan use case scenario.

### 4.5.1 Use Case Diagram

Use case diagram menyatakan visualisasi interaksi antara aktor atau pengguna sistem dengan sistem, dengan adanya use case diagram batasan dari sebuah sistem dapat terlihat jelas, lebih lengkapnya dapat dilihat pada Gambar 4.6.



**Gambar 4.6 Use Case Diagram**

### 4.5.2 Use Case Scenario

Use case scenario menggambarkan secara tekstual kumpulan skenario interaksi, tiap skenario akan mendeskripsikan urutan langkah yang akan dilakukan oleh aktor saat berinteraksi dengan sistem. Use case scenario dari tiap use case dapat dilihat lebih detail pada tabel berikut.

**Tabel 4.4 Use Case Scenario Login**

|  |  |
| --- | --- |
| *Login* (PSB\_F\_01\_00) | |
| ***Aktor*** | Pengunjung |
| ***Objective*** | Fungsi untuk dapat masuk kedalam sistem |
| ***Preconditions*** | Halaman *login* sudah ditampilkan |
| ***Main Flows*** | * + - 1. Sistem menampilkan form *login* dengan data inputan NIP/NISN dan *password*.       2. Aktor mengisi data dengan lengkap dan menekan tombol masuk.       3. Sistem melakukan pencocokan data dalam basisdata.       4. Sistem menampilkan halaman dashboard. |
| ***Alternative Flows*** | 3.1 Jika data akun belum ada dalam basisdata pendaftaran, maka sistem akan menampilkan form pendaftaran PSB.  4.1. Jika data inputan form login kosong, maka Sistem menampilkan pesan kesalahan “NISN harus diisi” dan “Password harus diisi”.  4.2. Jika data inputan NISN kurang atau lebih dari 10 digit, maka sistem menampilkan pesan kesalahan “NISN harus berjumlah 10 digit”.  4.3. Jika data password tidak sesuai dalam basisdata, maka sistem menampilkan kembali form login berikut pesan kesalahan “Password tidak sesuai”.  4.4. Jika data NIP/NISN tidak ditemukan dalam basisdata, maka sistem menampilkan kembali form login berikut pesan kesalahan “Akun tidak ditemukan, silahkan buat akun”. |
| ***Post Conditions*** | Aktor sudah teridentifikasi sebagai peserta, panitia atau admin. |

**Tabel 4.5 Use Case Scenario Registrasi PSB**

|  |  |
| --- | --- |
| Registrasi PSB (PSB\_F\_02\_00) | |
| ***Aktor*** | Peserta |
| ***Objective*** | Fungsi untuk mengisi data pendaftaran |
| ***Preconditions*** | Aktor terdefinisi sebagai peserta |
| ***Main Flows*** | Sistem menampilkan form pendaftaran PSB dengan data nama, no.tlp, nisn, jenis kelamin, email, alamat, tempat lahir, tanggal lahir, anak ke, jumlah saudara, alat transportasi, asal sekolah, alamat asal sekolah, alamat sekolah, nomor peserta UN, prodi pilihan, tinggal bersama, size baju, nama ayah, no.tlp ayah, masih hidup/tidak, email ayah, alamat ayah, pekerjaan ayah, penghasilan ayah, pendidikan ayah, nama ibu, no.tlp ibu, masih hiupu/tidak, email ibu, alamat ibu, pekerjaan ibu, penghasilan ibu, pendidikan ibu, rata-rata penghasilan, nama wali, no.tlp wali, hubungan keluarga, email wali, alamat wali dan pekerjaan wali.  Aktor mengisi data dengan lengkap dan menekan tombol daftar.  Sistem melakukan validasi dan menyimpan data.  Sistem menampilkan halaman dashboard serta pesan berhasil “Registrasi PSB berhasil dilakukan, silahkan lengkapi berkas pada menu Data Peserta”. |
| ***Alternative Flows*** | - |
| ***Post Conditions*** | Halaman dashboard serta pesan berhasil “Registrasi PSB berhasil dilakukan, silahkan lengkapi berkas pada menu Data Peserta” berhasil ditampilkan. |

**Tabel 4.6 Use Case Scenario Buat Akun**

|  |  |
| --- | --- |
| Buat akun (PSB\_F\_02\_00) | |
| ***Aktor*** | Pengunjung |
| ***Objective*** | Fungsi untuk terdaftar sebagai peserta dalam sistem |
| ***Preconditions*** | Halaman buat akun telah ditampilkan |
| ***Main Flows*** | Sistem menampilkan form buat akun dengan data inputan nama lengkap, NISN, *email* dan *password*.  Aktor mengisi data dengan lengkap dan menekan tombol buat akun.  Sistem melakukan validasi dan menyimpan data.  Sistem menampilkan halaman *login* berikut pesan berhasil “Akun berhasil dibuat, silahkan login”. |
| ***Alternative Flows*** | 3.1 Jika data inputan form buat akun kosong, maka sistem menampilkan pesan kesalahan “NISN harus diisi”, “nama harus diisi”, “email harus diisi” dan “password harus diisi”.  3.2 Jika data inputan NISN kurang atau lebih dari 10 digit, maka sistem menampilkan kembali form buat akun beserta pesan kesalahan “NISN harus berjumlah 10 digit”.  3.3 Jika data inputan re-password tidak sama dengan data inputan password, maka sistem menampilkan kembali form buat akun beserta pesan kesalahan “kata sandi tidak cocok”.  3.4 Jika data inputan email telah terdaftar dalam sistem, maka sistem menampilkan kembali form buat akun berikut pesan kesalahan “email telah terdaftar”.  3.5 Jika data inputan NISN telah terdaftar dalam sistem, maka sistem menampilkan kembali form buat akun berikut pesan kesalahan “NISN telah terdaftar”.  3.6 Jika data inputan password kurang dari 4 digit, maka, sistem menampilkan kembali form buat akun berikut pesan kesalahan “kata sandi terlalu pendek”. |
| ***Post Conditions*** | Aktor berhasil terdaftar sebagai peserta dan sistem menampilkan halaman *login* berikut pesan berhasil “Akun berhasil dibuat, silahkan login”. |

**Tabel 4.7 Use Case Scenario Ganti Password**

|  |  |
| --- | --- |
| Reset *password* (PSB\_F\_03\_00) | |
| ***Aktor*** | Pengunjung |
| ***Objective*** | Fungsi untuk me-reset *password* akun peserta |
| ***Preconditions*** | Halaman ganti *password* telah ditampilkan |
| ***Main Flows*** | 1. Sistem menampilkan form reset password dengan data inputan email. 2. Aktor mengisi data dengan lengkap dan menekan tombol reset password. 3. Sistem melakukan pencocokan data email dalam basis data. 4. Sistem mengirim link reset password melalui email. 5. Sistem menampilkan pesan pemberitahuan “*silahkan cek email anda*”. 6. Aktor menekan link reset password di email. 7. Sistem menampilkan form ganti password dengan data inputan password baru dan re-password baru. 8. Aktor mengisi data dengan lengkap dan menekan tombol simpan. 9. Sistem melakukan validasi dan menyimpan data. 10. Sistem menampilkan halaman *login* berikut pesan berhasil “password berhasil diperbaharui, silahkan login”. |
| ***Alternative Flows*** | 3.1 Jika data inputan email tidak ditemukan dalam basisdata, maka sistem menampilkan kembali form reset password berikut pesan kesalahan “email tidak terdaftar”.  9.1 Jika data inputan re-password baru tidak sama dengan data inputan password, maka sistem menampilkan kembali form ganti password beserta pesan kesalahan “kata sandi tidak cocok”.  9.2 Jika data inputan password kurang dari 4 digit, maka, sistem menampilkan kembali form ganti password berikut pesan kesalahan “kata sandi terlalu pendek”. |
| ***Post Conditions*** | Halaman login berhasil ditampilkan beserta pesan berhasil “*Password* berhasil diperbaharui, silahkan login”. |

**Tabel 4.8 Use Case Scenario Logout**

|  |  |
| --- | --- |
| *Logout* (PSB\_F\_04\_00) | |
| ***Aktor*** | Admin, peserta dan panitia PSB |
| ***Objective*** | Fungi untuk dapat menghilangkan hak akses pengguna |
| ***Preconditions*** | Aktor sudah teridentifikasi oleh sistem |
| ***Main Flows*** | Aktor memencet tombol *logout*.  Sistem menampilkan pesan konfirmasi “yakin untuk keluar?”.  Aktor menekan tombol ya.  Sistem menghapus data akun dari session.  Sistem menampilkan halaman utama. |
| ***Alternative Flows*** | 3.1 Jika aktor menekan tombol tidak, maka sistem akan mengembalikan halaman yang sedang diakses. |
| ***Post Conditions*** | Hak akses aktor berhasil dihilangkan dari session. |

**Tabel 4.9 Use Case Scenario Lihat Jadwal Kegiatan PSB**

|  |  |
| --- | --- |
| *Lihat jadwal kegiatan PSB* (PSB\_F\_04\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk menampilkan jadwal kegiatan yang sudah diinput |
| ***Preconditions*** | Aktor sudah teridentifikasi sebagai admin |
| ***Main Flows*** | Aktor menekan tombol PSB.  Sistem menampilkan halaman jadwal kegiatan dengan data tampilan nomor, kegiatan, tanggal mulai, tanggal berakhir, keterangan dan aksi, serta tombol hapus dan edit pada kolom aksi.  Sistem menampilkan menu jadwal, syarat dan ketentuan, biaya dan panduan.  Sistem menampilkan tombol tambah kegiatan. |
| ***Alternative Flows*** | 1.1 Jika belum ada data dalam basisdata, maka sistem menampilkan pesan pemberitahuan “No data available in table”. |
| ***Post Conditions*** | Halaman jadwal kegiatan PSB berhasil ditampilkan. |

**Tabel 4.10 Use Case Scenario Tambah Kegiatan**

|  |  |
| --- | --- |
| Tambah kegiatan (PSB\_F\_05\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk menambah kegiatan baru |
| ***Preconditions*** | Halaman jadwal kegiatan PSB telah ditampilkan |
| ***Main Flows*** | Aktor menekan tombol tambah kegiatan.  Sistem menampilkan form tambah kegiatan dengan data inputan nama kegiatan, tanggal mulai, tanggal berakhir, dan keterangan.  Aktor mengisi data secara lengkap dan menekan tombol tambah.  Sistem melakukan validasi dan menyimpan data.  Sistem menampilkan halaman jadwal kegiatan PSB serta pesan berhasil “kegiatan berhasil ditambah” |
| ***Alternative Flows*** | 3.1 Jika terdapat data inputan yang kosong, maka sistem akan menampilkan pesan kesalahan “Harap isi bidang ini”.  3.2 Jika aktor menekan tombol batal, maka sistem akan mengembalikan halaman jadwal kegiatan PSB. |
| ***Post Conditions*** | Halaman jadwal kegiatan PSB berhasil ditampilkan beserta pesan berhasil “kegiatan berhasil ditambah”. |

**Tabel 4.11** **Use Case Scenario Edit Kegiatan**

|  |  |
| --- | --- |
| Edit kegiatan (PSB\_F\_06\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk mengubah data kegiatan |
| ***Preconditions*** | Halaman jadwal kegiatan PSB telah ditampilkan |
| ***Main Flows*** | Aktor menekan tombol edit pada kolom aksi.  Sistem menampilkan form edit yang berisi data nama kegiatan, tanggal mulai, tanggal berakhir, dan keterangan.  Aktor mengisi data yang ingin diubah dan menekan tombol simpan.  Sistem melakukan validasi dan menyimpan perubahan data.  Sistem menampilkan halaman jadwal kegiatan PSB serta pesan berhasil “kegiatan berhasil diperbaharui” |
| ***Alternative Flows*** | 3.1 Jika terdapat data inputan yang kosong, maka sistem akan menampilkan pesan kesalahan “Harap isi bidang ini”.  3.2 Jika aktor menekan tombol batal, maka sistem akan mengembalikan halaman jadwal kegiatan PSB. |
| ***Post Conditions*** | Halaman jadwal kegiatan PSB berhasil ditampilkan beserta pesan berhasil “kegiatan berhasil diperbaharui”. |

**Tabel 4.12 Use Case Scenario Hapus Kegiatan**

|  |  |
| --- | --- |
| Hapus kegiatan (PSB\_F\_07\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk menghapus kegiatan dari daftar kegiatan |
| ***Preconditions*** | Halaman jadwal kegiatan PSB telah ditampilkan |
| ***Main Flows*** | Aktor memencet tombol hapus pada kolom aksi.  Sistem menampilkan pesan konfirmasi “Yakin untuk menghapus kegiatan?”.  Aktor memencet tombol ya.  Sistem menghapus data kegiatan dalam jadwal kegiatan.  Sistem menampilkan halaman jadwal kegiatan PSB serta pesan berhasil “kegiatan berhasil dihapus”. |
| ***Alternative Flows*** | 3.1 Jika aktor menekan tombol tidak, maka sistem akan mengembalikan halaman jadwal kegiatan PSB. |
| ***Post Conditions*** | Halaman jadwal kegiatan PSB berhasil ditampilkan beserta pesan berhasil “kegiatan berhasil dihapus”. |

**Tabel 4.13 Use Case Scenario Lihat Syarat dan Ketentuan**

|  |  |
| --- | --- |
| Lihat syarat dan ketentuan (PSB\_F\_08\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk menampilkan daftar syarat dan ketentuan PSB |
| ***Preconditions*** | Aktor sudah teridentifikasi sebagai admin |
| ***Main Flows*** | 1. Aktor menekan tombol syarat dan ketentuan. 2. Sistem menampilkan halaman syarat dan ketentuan dengan data tampilan nomor, berkas prasyarat, ketentuan dan aksi, serta tombol hapus dan edit pada kolom aksi. 3. Sistem menampilkan menu jadwal, syarat dan ketentuan, biaya dan panduan. 4. Sistem menampilkan tombol tambah syarat dan ketentuan. |
| ***Alternative Flows*** | 1.1 Jika belum ada data dalam basisdata, maka sistem menampilkan pesan pemberitahuan “No data available in table”. |
| ***Post Conditions*** | Halaman daftar syarat dan ketentuan berhasil ditampilkan. |

**Tabel 4.14 Use Case Scenario Tambah Syarat dan Ketentuan**

|  |  |
| --- | --- |
| Tambah syarat dan ketentuan (PSB\_F\_09\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk menambah syarat dan ketentuan |
| ***Preconditions*** | Halaman daftar syarat dan ketentuan sudah ditampilkan |
| ***Main Flows*** | * + - 1. Aktor memencet tombol tambah syarat dan ketentuan.       2. Sistem menampilkan form tambah syarat dan ketentuan dengan data inputan nama berkas dan ketentuan.       3. Aktor mengisi data secara lengkap dan menekan tombol tambah.       4. Sistem melakukan validasi dan menyimpan data.       5. Sistem menampilkan halaman daftar syarat dan ketentuan serta pesan berhasil “syarat dan ketentuan berhasil ditambah” |
| ***Alternative Flows*** | * 1. Jika terdapat data inputan yang kosong, maka sistem akan menampilkan pesan kesalahan “Harap isi bidang ini”.   2. Jika aktor menekan tombol batal, maka sistem akan mengembalikan halaman daftar syarat dan ketentuan. |
| ***Post Conditions*** | Halaman daftar syarat dan ketentuan berhasil ditampilkan beserta pesan berhasil “syarat dan ketentuan berhasil ditambah”. |

**Tabel 4.15 Use Case Scenario Edit Syarat dan Ketentuan**

|  |  |
| --- | --- |
| Edit syarat dan ketentuan (PSB\_F\_10\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk mengubah data syarat dan ketentuan |
| ***Preconditions*** | Halaman daftar syarat dan ketentuan sudah ditampilkan |
| ***Main Flows*** | * + - 1. Aktor menekan tombol edit dalam kolom aksi.  1. Sistem menampilkan form edit yang berisi data nama berkas dan ketentuan. 2. Aktor mengisi data yang ingin diubah dan menekan tombol simpan. 3. Sistem melakukan validasi dan menyimpan perubahan data. 4. Sistem menampilkan halaman daftar syarat dan ketentuan serta pesan berhasil “syarat dan ketentuan berhasil diperbaharui”. |
| ***Alternative Flows*** | * 1. Jika terdapat data inputan yang kosong, maka sistem akan menampilkan pesan kesalahan “Harap isi bidang ini”.   2. Jika aktor memencet tombol batal, maka sistem akan menyembalikan halaman daftar syarat dan ketentuan. |
| ***Post Conditions*** | Halaman daftar syarat dan ketentuan berhasil ditampilkan beserta pesan berhasil “syarat dan ketentuan berhasil diperbaharui”. |

**Tabel 4.16 Use Case Scenario Hapus Syarat dan Ketentuan**

|  |  |
| --- | --- |
| Hapus syarat dan ketentuan (PSB\_F\_11\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk menghapus data syarat dan ketentuan |
| ***Preconditions*** | Halaman daftar syarat dan ketentuan sudah ditampilkan |
| ***Main Flows*** | * 1. Aktor menekan tombol hapus dalam kolom aksi.   2. Sistem menampilkan pesan konfirmasi “Yakin untuk menghapus data sebagai syarat dan ketentuan?”.   3. Aktor memencet tombol ya.   4. Sistem menghapus data syarat dan ketentuan.   5. Sistem menampilkan halaman daftar syarat dan ketentuan serta pesan berhasil “syarat dan ketentuan berhasil dihapus”. |
| ***Alternative Flows*** | 3.1 Jika aktor memencet tombol tidak, maka sistem akan mengembalikan halaman daftar syarat dan ketentuan. |
| ***Post Conditions*** | Halaman daftar syarat dan ketentuan berhasil ditampilkan beserta pesan berhasil “syarat dan ketentuan berhasil dihapus”. |

**Tabel 4.17 Use Case Scenario Lihat Daftar Biaya**

|  |  |
| --- | --- |
| Lihat daftar biaya (PSB\_F\_12\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk menampilkan daftar biaya PSB |
| ***Preconditions*** | Aktor sudah terdefinisi sebagai admin |
| ***Main Flows*** | * + - 1. Aktor menekan tombol biaya.       2. Sistem menampilkan halaman biaya dengan data tampilan nomor, jenis pembayaran, nominal, keterangan, pembayaran dan aksi, serta tombol hapus dan edit pada kolom aksi.       3. Sistem menampilkan menu jadwal, syarat dan ketentuan, biaya dan panduan.       4. Sistem menampilkan tombol tambah pembayaran. |
| ***Alternative Flows*** | 1.1 Jika belum ada data dalam basisdata, maka sistem menampilkan pesan pemberitahuan “No data available in table”. |
| ***Post Conditions*** | Halaman daftar biaya berhasil ditampilkan. |

**Tabel 4.18 Use Case Scenario Tambah Biaya**

|  |  |
| --- | --- |
| Tambah biaya (PSB\_F\_13\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk menambah jenis pembayaran peserta |
| ***Preconditions*** | Halaman daftar biaya sudah ditampilkan |
| ***Main Flows*** | Aktor menekan tombol tambah pembayaran.  Sistem menampilkan form tambah pembayaran dengan data inputan nama pembayaran, jumlah, keterangan dan pembayaran untuk.  Aktor mengisi data secara lengkap dan menekan tombol tambah.  Sistem melakukan validasi dan menyimpan data.  Sistem menampilkan halaman daftar biaya serta pesan berhasil “pembayaran berhasil ditambah”. |
| ***Alternative Flows*** | * 1. Jika belum ada data dalam basisdata, maka sistem menampilkan pesan pemberitahuan “No data available in table”.   3.1 Jika aktor menekan tombol batal, maka sistem akan mengembalikan halaman daftar biaya. |
| ***Post Conditions*** | Halaman daftar biaya berhasil ditampilkan beserta pesan berhasil “pembayaran berhasil ditambah”. |

**Tabel 4.19 Use Case Scenario Edit Biaya**

|  |  |
| --- | --- |
| Edit biaya (PSB\_F\_14\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk mengubah data pembayaran |
| ***Preconditions*** | Halaman daftar biaya sudah ditampilkan |
| ***Main Flows*** | Aktor menekan tombol edit dalam kolom aksi.  Sistem menampilkan form edit pembayaran tang berisi data nama pembayaran, jumlah, keterangan dan pembayaran untuk.  Aktor mengubah data yang ingin diubah dan menekan tombol simpan.  Sistem melakukan validasi dan menyimpan perubahan data.  Sistem menampilkan halaman daftar biaya serta pesan berhasil “pembayaran berhasil diperbaharui”. |
| ***Alternative Flows*** | * 1. Jika terdapat data inputan yang kosong, maka sistem akan menampilkan pesan kesalahan “Harap isi bidang ini”.   2. Jika aktor menekan tombol batal, maka sistem akan mengembalikan halaman daftar biaya. |
| ***Post Conditions*** | Halaman daftar biaya berhasil ditampilkan beserta pesan berhasil “pembayaran berhasil diperbaharui”. |

**Tabel 4.20 Use Case Scenario Hapus Biaya**

|  |  |
| --- | --- |
| Hapus biaya (PSB\_F\_15\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk menghapus data pembayaran |
| ***Preconditions*** | Halaman daftar biaya sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol hapus dalam kolom aksi. 2. Sistem menampilkan pesan konfirmasi “Yakin untuk menghapus pembayaran?”. 3. Aktor menekan tombol ya. 4. Sistem menghapus data pembayaran. 5. Sistem menampilkan halaman daftar biaya serta pesan berhasil “Pembayaran berhasil dihapus”. |
| ***Alternative Flows*** | 3.1 Jika aktor menekan tombol tidak, maka sistem akan mengemballikan halaman daftar biaya. |
| ***Post Conditions*** | Halaman daftar biaya berhasil ditampilkan serta pesan berhasil “Pembayaran berhasil dihapus”. |

**Tabel 4.21 Use Case Scenario Lihat Daftar Panduan**

|  |  |
| --- | --- |
| Lihat daftar panduan (PSB\_F\_16\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk melihat daftar file panduan PSB |
| ***Preconditions*** | Aktor terdefinisi sebagai admin |
| ***Main Flows*** | * 1. Aktor menekan tombol panduan.   2. Sistem menampilkan halaman daftar panduan dengan data tampilan nomor, nama file, keteranganfile dan aksi, serta tombol lihat dan hapus dalam kolom aksi.   3. Sistem menampilkan menu jadwal, syarat dan ketentuan, biaya dan panduan.   4. Sistem menampilkan form unggah panduan dengan data inputan nama berkas, keterangan dan tombol pilih file. |
| ***Alternative Flows*** | 1.1 Jika belum ada data dalam basisdata, maka sistem menampilkan pesan pemberitahuan “No data available in table”. |
| ***Post Conditions*** | Halaman daftar panduan berhasil ditampilkan. |

**Tabel 4.22 Use Case Scenario Unggah Panduan**

|  |  |
| --- | --- |
| Unggah panduan (PSB\_F\_16\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk mengunggah file panduan PSB |
| ***Preconditions*** | Halaman daftar panduan sudah ditampilkan |
| ***Main Flows*** | * 1. Aktor mengisi data dengan lengkap dan menekan tombol unggah.   2. Sistem melakukan validasi dan menyimpan data unggahan.   3. Sistem menampilkan kembali form unggah panduan serta pesan berhasil “Panduan berhasil diunggah”. |
| ***Alternative Flows*** | 2.1 Jika file belum dipilih, maka sistem akan menampilkan pesan kesalahan “gagal, silahkan pilih berkas”.  2.2 Jika file yang diunggah buka format pdf, maka Sistem akan menampilkan pesan kesalahan “berkas harus berupa pdf”.  2.3. Jika terdapat data inputan yang kosong, maka sistem akan menampilkan pesan kesalahan “harap isi bidang ini”. |
| ***Post Conditions*** | Pemberitahuan “Panduan berhasil diunggah” berhasil ditampilkan. |

**Tabel 4.23 Use Case Scenario Lihat Panduan**

|  |  |
| --- | --- |
| Lihat Panduan (PSB\_F\_17\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk melihat isi file panduan |
| ***Preconditions*** | Halaman daftar panduan sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol lihat pada kolom aksi.  2. Sistem menampilkan isi file panduan |
| ***Alternative Flows*** | - |
| ***Post Conditions*** | Isi file panduan berhasil ditampilkan. |

**Tabel 4.24 Use Case Unduh Panduan**

|  |  |
| --- | --- |
| Unduh panduan (PSB\_F\_18\_00) | |
| ***Aktor*** | Pengunjung |
| ***Objective*** | Fungsi untuk mengunduh file panduan PSB |
| ***Preconditions*** | Aktor sudah keluar dari sistem |
| ***Main Flows*** | 1. Aktor menekan tombol informasi PSB. 2. Sistem menampilkan halaman informasi PSB dengan menu pilihan jadwal, biaya, syarat dan ketentuan dan panduan. 3. Aktor menekan tombol panduan. 4. Sistem menampilkan tombol download. 5. Aktor menekan tombol download. 6. Sistem mulai memproses pengunduhan file panduan. |
| ***Alternative Flows*** | - |
| ***Post Conditions*** | File panduan berhasil diunduh. |

**Tabel 4.25 Use Case Scenario Hapus Panduan**

|  |  |
| --- | --- |
| Hapus panduan (PSB\_F\_18\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk menghapus file panduan |
| ***Preconditions*** | Halaman daftar panduan sudah ditampilkan |
| ***Main Flows*** | * + - 1. Aktor menekan tombol hapus dalam kolom aksi.       2. Sistem menampilkan pesan konfirmasi “Yakin untuk menghapus file?”.       3. Aktor menekan tombol ya.       4. Sistem memproses hapus file panduan.       5. Sistem menampilkan pesan berhasil “file berhasil dihapus”. |
| ***Alternative Flows*** | 3.1 Jika aktor menekan tombol tidak, maka sistem akan mengembalikan halaman daftar panduan. |
| ***Post Conditions*** | File panduan berhasil dihapus dan pemberitahuan “file berhasil dihapus” berhasil ditampilkan. |

**Tabel 4.26 Use Case Scenario Lihat Profil Sekolah**

|  |  |
| --- | --- |
| Lihat Profil Sekolah (PSB\_F\_19\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk melihat profil sekolah |
| ***Preconditions*** | Aktor terdefinisi sebagai admin |
| ***Main Flows*** | Aktor menekan tombol profil sekolah.   1. Sistem menampilkan halaman profil sekolah dengan data tampilan jenis informasi, deskripsi dan aksi, serta tombol edit dan hapus dalam kolom aksi. 2. Sistem menampilkan tombol tambah. |
| ***Alternative Flows*** | - |
| ***Post Conditions*** | Halaman profil sekolah berhasil ditampilkan. |

**Tabel 4.27 Use Case Scenario Tambah Profil Sekolah**

|  |  |
| --- | --- |
| Tambah profil sekolah (PSB\_F\_20\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk menambah data profil sekolah |
| ***Preconditions*** | Halaman profil sekolah sudah ditampilkan |
| ***Main Flows*** | * 1. Aktor menekan tombol tambah.   2. Sistem menampilkan form tambah informasi dengan data inputan jenis informasi dan deskripsi.   3. Aktor mengisi data dengan lengkap dan menekan tombol tambah.   4. Sistem melakukan validasi dan menyimpan data.   5. Sistem menampilkan halaman profil sekolah serta pesan berhasil “informasi baru berhasil ditambah”. |
| ***Alternative Flows*** | 3.1 Jika terdapat data inputan yang kosong, maka sistem akan menampilkan pesan kesalahan “harap isi bidang ini”.  3.2 Jika aktor menekan tombol batal, maka sistem akan mengembalikan halaman profil sekolah. |
| ***Post Conditions*** | Halaman profil sekolah dan pesan berhasil “informasi baru berhasil ditambah” berhasil ditampilkan. |

**Tabel 4.28 Use Case Scenario Edit Profil Sekolah**

|  |  |
| --- | --- |
| Edit profil sekolah (PSB\_F\_21\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk mengubah informasi profil sekolah |
| ***Preconditions*** | Halaman profil sekolah sudah ditampilkan |
| ***Main Flows*** | * 1. Aktor menekan tombol edit dalam kolom aksi.   2. Sistem menampilkan form edit profil sekolah yang berisi data jenis informasi dan deskripsi.   3. Aktor mengubah data yang ingin diperbaharui dan menekan tombol simpan.   4. Sistem melakukan validasi dan menyimpan data perubahan.   5. Sistem menampilkan halaman profil sekolah beserta pesan berhasil “profil sekolah berhasil diperbaharui”. |
| ***Alternative Flows*** | 3.1 Jika aktor menekan tombol batal, maka sistem akan mengembalikan halaman profil sekolah. |
| ***Post Conditions*** | Halaman profil sekolah berhasil dan pesan berhasil “profil sekolah berhasil diperbaharui” berhasil ditampilkan. |

**Tabel 4.29 Use Case Scenario Hapus Profil Sekolah**

|  |  |
| --- | --- |
| Hapus profil sekolah (PSB\_F\_22\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk menghapus informasi profil sekolah |
| ***Preconditions*** | Halaman profil sekolah sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol hapus dalam kolom aksi. 2. Sistem menampilkan pesan konfirmasi “Yakin untuk menghapus informasi?”. 3. Aktor menekan tombol ya. 4. Sistem memproses penghapusan data. 5. Sistem menampilan halaman profil sekolah seerta pesan berhasil “Informasi berhasil dihapus. |
| ***Alternative Flows*** | 3.1 Jika aktor menekan tombol tidak, maka sistem akan menampilkan kembali halaman profil sekolah. |
| ***Post Conditions*** | Pemberitahuan “Kartu peserta berhasil diunduh” berhasil ditampilkan. |

**Tabel 4.30 Use Case Scenario Lihat Daftar Berita**

|  |  |
| --- | --- |
| Lihat daftar berita (PSB\_F\_23\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk melihat daftar berita yang telah dirilis |
| ***Preconditions*** | Aktor terdefinisi sebagai admin |
| ***Main Flows*** | * 1. Aktor menekan tombol berita.   2. Sistem menampilkan halaman daftar berita dengan data tampilan nomor, judul, penulis, tanggal publish, foto dan aksi, serta tombol edit dan hapus dalam kolom aksi.   3. Sistem menampilkan kolom pencarian data.   4. Sistem menampilkan tombol tambah berita. |
| ***Alternative Flows*** | 1.1 Jika belum ada data dalam basisdata, maka sistem akan menampilkan pemberitahuan “No data available in table”. |
| ***Post Conditions*** | Halaman daftar berita berhasil ditampilkan. |

**Tabel 4.31 Use Case Scenario Baca Berita**

|  |  |
| --- | --- |
| Baca berita (PSB\_F\_24\_00) | |
| ***Aktor*** | Pengunjung |
| ***Objective*** | Fungsi untuk membaca berita secara detail |
| ***Preconditions*** | Aktor berada pada halaman utama |
| ***Main Flows*** | 1. Aktor menekan tombol berita.  2. Sistem menampilkan halaman berita.  3. Aktor menekan tombol baca berita pada berita yang ingin dibaca secara lanjut.  4. Sistem menampilkan halaman detail berita dengan data tampilan judul berita, tanggal publish, penulis, isi berita dan foto berita. |
| ***Alternative Flows*** | - |
| ***Post Conditions*** | Halaman detail berita berhasil ditampilkan. |

**Tabel 4.32 Use Case Scenario Tambah Berita**

|  |  |
| --- | --- |
| Tambah berita (PSB\_F\_25\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk menambah berita baru |
| ***Preconditions*** | Aktor terdefinisi sebagai admin |
| ***Main Flows*** | * 1. Aktor menekan tombol tambah berita   2. Sistem menampilkan halaman tambah berita dengan data inputan judul, penulis, tanggal publish, isi berita dan tombol pilih file.   3. Aktor mengisi data secara lengkap dan menekan tombol tambah.   4. Sistem melakukan validasi dan menyimpan data.   5. Sistem menampilkan halaman daftar berita serta pesan berhasil “Berita berhasil ditambah”. |
| ***Alternative Flows*** | * 1. Jika terdapat data inputan yang kosong, maka sistem akan menampilkan pemberitahuan “Harap isi bidang ini”.   2. Jika aktor memilih file dengan format yang tidak sesuai, maka sistem akan menampilkan kembali halaman daftar berita serta pesan kesalahan “berita gagal ditambah, format file tidak sah”. |
| ***Post Conditions*** | Halaman daftar berita dan pesan berhasil “Berita berhasil ditambah” berhasil ditampilkan. |

**Tabel 4.33 Use Case Scenario Edit Berita**

|  |  |
| --- | --- |
| Edit berita (PSB\_F\_26\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk mengubah data berita |
| ***Preconditions*** | Halahalaman daftar berita sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol edit dalam kolom aksi. 2. Sistem menampilkan halaman edit berita yang berisi data judul, penulis, tanggal publish, isi berita dan foto berita. 3. Aktor mengubah data yang ingin diperbaharui dan menekan tombol simpan. 4. Sistem melakukan validasi dan menyimpan perubahan data. 5. Sistem menampilkan halaman daftar berita serta pesanberhasil “berita berhasil diperbaharui”. |
| ***Alternative Flows*** | 4.1 Jika aktor memilih file dengan format yang tidak sesuai, maka sistem akan menampilkan kembali halaman daftar berita serta pesan kesalahan “berita gagal diperbaharui, format file tidak sah”. |
| ***Post Conditions*** | Halaman daftar berita dan pesan berhasil ”berita berhasil diperbaharui” berhasil ditampilkan. |

**Tabel 4.34 Use Case Scenario Hapus Berita**

|  |  |
| --- | --- |
| Hapus berita (PSB\_F\_27\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk menghapus berita dari daftar berita |
| ***Preconditions*** | Halaman daftar berita sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan rombol hapus dalam kolom aksi. 2. Sistem menampilkan pesan konfirmasi “yakinuntuk menghapus berita?”. 3. Aktor menekan tombol ya. 4. Sistem melakukan proses hapus data. 5. Sistem menampilkan halaman daftar berita serta pesan berhasil “berita berhasil dihapus”. |
| ***Alternative Flows*** | 3.1 Jika aktor menekan tombol tidak, maka sistem akan menampilkan kembali halaman daftar berita. |
| ***Post Conditions*** | Halaman daftar berita dan pesan berhasil “berita berhasil dihapus” berhasil ditampilkan. |

**Tabel 4.35 Use Case Scenario Lihat Daftar Pengguna**

|  |  |
| --- | --- |
| Lihat daftar pengguna (PSB\_F\_28\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk melihat daftar pengguna sistem |
| ***Preconditions*** | Aktor terdefinisi sebagai admin |
| ***Main Flows*** | 1. Aktor menekan tombol pengguna. 2. Sistem menampilkan halaman daftar pengguna diantaranya admin dan peserta, dengan data tampilan untuk admin nomor, nama, NIP, email, nomor handphone, role dan aksi, serta tombol edit dan hapus dalam kolom aksi, selain itu untuk peserta sistem menampilkan data tampilan nomor, nama, kode daftar, email, role dan aksi, serta tombol arsip dalam kolom aksi. 3. Sistem menampilkan kolom pencarian. 4. Sistem menampilkan tombol tambah admin. |
| ***Alternative Flows*** | 1.1 Jika data belum ada dalam basisdata, maka sistem menampilkan pemberitahuan “No data available in table”. |
| ***Post Conditions*** | Halaman daftar pengguna berhasil ditampilkan. |

**Tabel 4.36 Use Case Scenario Tambah Admin**

|  |  |
| --- | --- |
| Tambah admin (PSB\_F\_29\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk menambah pengguna baru dengan role admin atau panitia |
| ***Preconditions*** | Halaman daftar pengguna sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol tambah admin. 2. Sistem menampilkan form tambah admin dengan data input NIP, nama, email, nomor handphone dan role. 3. Aktor mengisi data secara lengkap dan menekan tombol tambah. 4. Sistem melakukan validasi dan meyimpan data. 5. Sistem mengirim pesan kata sandi akun melalui email dan whatsapp. 6. Sistem menampilkan halaman daftar penggun serta pesan berhasil “password telah dikirim, pengguna berhasil ditambahkan”. |
| ***Alternative Flows*** | 3.1 Jika terdapat data inputan yang kosong, maka sistem akan menampilkan pemberitahuan “Harap isi bidang ini”.  3.2 Jika aktor menekan tombol batal, maka sistem akan emngembalikan halaman daftar pengguna. |
| ***Post Conditions*** | Halaman daftar pengguna dan pesan berhasil “password telah dikirim, pengguna berhasil ditambahkan” berhasil ditampilkan. |

**Tabel 4.37 Use Case Scenario Edit Admin**

|  |  |
| --- | --- |
| Edit admin (PSB\_F\_30\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk mengubah data pengguna dengan role admin atau panitia |
| ***Preconditions*** | Halaman daftar pengguna sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol edit dalam kolom aksi. 2. Sistem menampilkan form edit akun 3. Aktor mengubah data yang ingin diubah dan menekan tombol simpan. 4. Sistem melakukan validasi dan menyimpan perubahan data. 5. Sistem menampilkan halaman daftar pengguna serta pesan berhasil “akun berhasil diperbaharui”. |
| ***Alternative Flows*** | 3.1 Jika aktor menekan tombol batal, maka sistem akan mengembalikan halaman daftar pengguna. |
| ***Post Conditions*** | Halaman daftar pengguna serta pesan berhasil “akun berhasil diperbaharui” berhasil ditampilkan. |

**Tabel 4.38 Use Case Scenario Hapus Admin**

|  |  |
| --- | --- |
| Hapus admin (PSB\_F\_31\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk menghapus akun dengan role admin atau panitia |
| ***Preconditions*** | Halaman daftar pengguna sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol hapus dalam kolom aksi. 2. Sistem menampilkan pesan konfirmasi “yakin untuk menghapus admin dengan nama ... ?”. 3. Aktor menekan tombol ya. 4. Sistem melakukan proses hapus data. 5. Sistem menampilkan halaman daftar pengguna serta pesan berhasil “Pengguna berhasil dihapus”. |
| ***Alternative Flows*** | 3.1 Jika aktor menekan tombol tidak, maka sistem akan mengembalikan halaman daftar pengguna. |
| ***Post Conditions*** | Halaman daftar penggun serta pesan berhasil “pengguna berhasil dihapus” berhasil ditampilkan. |

**Tabel 4.39 Use Case Scenario Arsip Peserta**

|  |  |
| --- | --- |
| Arsip peserta (PSB\_F\_32\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk mengarsip data peserta PSB |
| ***Preconditions*** | Halaman daftar pengguna sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol arsip dalam kolom aksi. 2. Sistem menampilkan pesan konfirmasi “yakin untuk mengarsip data?”. 3. Aktor menekan tombol ya. 4. Sitem melakukan proses update status aktif data peserta menjadi tidak aktif. 5. Sistem menampilkan halaman daftar pengguna serta pesan berhasil “Pengguna berhasil diarsipkan”. |
| ***Alternative Flows*** | 3.1 Jika aktor menekan tombol tidak, maka sistem akan mengambalikan halaman daftar pengguna. |
| ***Post Conditions*** | Update status aktif data peserta berhasil dilakukan. |

**Tabel 4.40 Use Case Scenario Lihat Daftar Data Arsip**

|  |  |
| --- | --- |
| Lihat daftar data arsip (PSB\_F\_33\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk melihat daftar peserta yang diarsip |
| ***Preconditions*** | Aktor terdefinisi sebagai admin |
| ***Main Flows*** | 1. Aktor menekan tombol data arsip 2. Sistem menampilkan halaman daftar data arsip dengan data tampilan nomor, nama, kode daftar, email, tanggal pendaftaran dan tombol detail dalam kolom aksi. 3. Sistem menampilkan kolom pencarian dan tombol excel. |
| ***Alternative Flows*** | 1.1 Jika belum ada data dalam basisdata, makasistem akan menampilkan pemberitahuan “No data available in table”. |
| ***Post Conditions*** | Halaman daftar data arsip berhasil ditampilkan. |

**Tabel 4.41 Use Case Scenario Lihat Detail Data Arsip**

|  |  |
| --- | --- |
| Lihat detail data arsip (PSB\_F\_34\_00) | |
| ***Aktor*** | Admin |
| ***Objective*** | Fungsi untuk melihat data arsip secara detail |
| ***Preconditions*** | Halaman daftar data arsip sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol detail dalam kolom aksi.  2. Sistem menampilkan halaman detail data arsip dengan data tampilan data diri, orangtua, wali dan berkas prasyarat.  3. Sistem menampilkan tombol lihat dan kembali. |
| ***Alternative Flows*** | - |
| ***Post Conditions*** | Halaman detail data arsip berhasil ditampilkan. |

**Tabel 4.42 Use Case Scenario Lihat Daftar Calon Santri**

|  |  |
| --- | --- |
| Lihat daftar calon santri (PSB\_F\_35\_00) | |
| ***Aktor*** | Panitia |
| ***Objective*** | Fungi untuk melihat daftar pendaftar PSB |
| ***Preconditions*** | Aktor terdefinisi sebagai panitia |
| ***Main Flows*** | 1. Aktor menkan tombol calon santri. 2. Sistem menampilkan halaman daftar calon santri dengan data tampilan nomor, kode daftar, nama, nisn, asal sekolah, status verifikasi dan aksi, serta tombol detail dalam kolom aksi. 3. Sistem menampilkan kolom pencarian. 4. Sistem menampilkan tombol cetak daftar peserta, pdf dan excel. |
| ***Alternative Flows*** | 1.1 Jika belum ada data dalam basisdata, maka sistem akan menampilkan pemberitahuan “No data available in table”. |
| ***Post Conditions*** | Halaman daftar calon santri berhasil ditampilkan. |

**Tabel 4.43 Use Case Scenario Cetak Daftar Calon Santri**

|  |  |
| --- | --- |
| Cetak daftar calon santri (PSB\_F\_36\_00) | |
| ***Aktor*** | Panitia |
| ***Objective*** | Fungsi untuk mencetak daftar pendaftar PSB |
| ***Preconditions*** | Halaman daftar calon santri sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol cetak daftar peserta. 2. Sistem melakukan proses cetak data. |
| ***Alternative Flows*** | 1.1 Jika aktor menekan tombol cetak dan data belum ada dalam basisdata, maka sistem akan menampilkan kembali halaman daftar calon santri serta pesan kesalahan “belum ada data yang dapat dicetak”. |
| ***Post Conditions*** | Daftar calon santri berhasil dicetak. |

**Tabel 4.44 Use Case Scenario Lihat Detail Data Calon Santri**

|  |  |
| --- | --- |
| Lihat detail data calon santri (PSB\_F\_37\_00) | |
| ***Aktor*** | Panitia |
| ***Objective*** | Fungsi untuk melihat data calon santri secara detail |
| ***Preconditions*** | Halaman daftar calon santri sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol detail dalam kolom aksi. 2. Sistem menampilkanhalaman detail data calon santri dengan data tampilan data diri, orangtua, wali dan berkas prasyarat. 3. Sistem menampilkan tombol verifikasi data. 4. Sistem menampilkan tombol lihat, terima dan tolak pada data tampilan berkas prasyarat. |
| ***Alternative Flows*** | - |
| ***Post Conditions*** | Halaman detail data calon santri berhasil ditampilkan. |

**Tabel 4.45 Use Case Scenario Verifikasi Data**

|  |  |
| --- | --- |
| Verifikasi data (PSB\_F\_38\_00) | |
| ***Aktor*** | Panitia |
| ***Objective*** | Fungsi untuk melakukan verifikasi pada data calon santri |
| ***Preconditions*** | Halaman detail data calon santri sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol verifikasi data. 2. Sistem menampilkan form pesan verifikasi dengan data inputan nama, kode daftar, nomor telepon, status verifikasi dan pesan. 3. Aktor mengisi data dengan lengkap dan menekan tombol kirim. 4. Sistem melakukan validasi dan menyimpan data. 5. Sistem melakukan kirim pesan verifikasi melalui whatsapp. 6. Sistem menampilkan halaman detail data calon santri serta pesan berhasil “pesan telah dikirim, status verifikasi berhasil diperbaharui”. |
| ***Alternative Flows*** | * 1. jika aktor menekan tombol batal, maka sistem akan mengembalikan halaman detail data calon santri.   2. Jik terdapat data input yang kosong, maka sistem akan menampilkan pesan pemberitahuan “Harap isi bidang ini”. |
| ***Post Conditions*** | Data calon santri berhasil diverifikasi. |

**Tabel 4.46 Use Case Scenario Lihat Berkas**

|  |  |
| --- | --- |
| Lihat berkas (PSB\_F\_39\_00) | |
| ***Aktor*** | Panitia dan peserta |
| ***Objective*** | Fungsi untuk melihat isi berkas yang diunggah calon santri |
| ***Preconditions*** | Halaman detail data calon santri sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol lihat dalam data berkas prasyarat. 2. Sistem menampilkan isi berkas calon santri. |
| ***Alternative Flows*** | - |
| ***Post Conditions*** | Isi berkas calon santri berhasil ditampilkan. |

**Tabel 4.47 Use Case Scenario Lihat Daftar Pembayaran**

|  |  |
| --- | --- |
| Lihat daftar jawaban (PSB\_F\_40\_00) | |
| ***Aktor*** | Panitia PSB |
| ***Objective*** | Dapat melihat daftar jawaban dari peserta PSB |
| ***Preconditions*** | Halaman tes akademik sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet link tab jawaban.  2. Sistem menampilkan halaman jawaban denga data meliputi waktu selesai, nama, NISN, jawaban dan tombol lihat detail pada data aksi.  3. Sistem menampilkan daftar jawaban dalam bentuk tabel. |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “Belum terdapat jawaban”, jika aktor memencet link tab jawaban namun belum terdapat jawbaan dari peserta PSB. |
| ***Post Conditions*** | Daftar jawaban peserta PSB berhasil ditampilkan. |

**Tabel 4.48 Use Case Scenario Lihat Daftar Soal Tes**

|  |  |
| --- | --- |
| Lihat daftar soal tes (PSB\_F\_41\_00) | |
| ***Aktor*** | Panitia |
| ***Objective*** | Fungsi untuk melihat daftar soal tes |
| ***Preconditions*** | Aktor terdefinisi sebagai panitia |
| ***Main Flows*** | 1. Aktor menekan tombol akademik 2. Sistem menampilkan halaman daftar soal tes dengan data tampilan nomor, soal, mapel, kunci jawaban dan aksi, serta tombol edit dan hapus dalam kolom aksi. 3. Sistem menampilkan menu soal, tes seleksi dan hasil tes. 4. Sistem menampilkan kolom pencarian. 5. Sistem menampilkan tombol tambah soal. 6. Sistem menampilkan tombol export file dalam format pdf atau excel dan tombol print untuk melakukan cetak daftar soal. |
| ***Alternative Flows*** | 1.1 Jika belum ada data dalam basisdata, maka sistem akan menampilkan pesan pemberitauhan “No data available in table”. |
| ***Post Conditions*** | Halaman daftar soal tes berhasil ditampilkan. |

**Tabel 4.49 Use Case Scenario Tambah Soal**

|  |  |
| --- | --- |
| Tambah soal (PSB\_F\_42\_00) | |
| ***Aktor*** | Panitia |
| ***Objective*** | Fungsi untuk menambah soal tes |
| ***Preconditions*** | Halaman daftar soal tes sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol tambah soal. 2. Sistem menampilkan halaman tambah soal dengan data inputan mapel, soal, opsi a, opsi b, opsi c, opsi d, kunci jawaban dan bobot nilai. 3. Aktor mengisi data secara lengkap dan menekan tombol tambah. 4. Sistem melakukan validasi dan menyimpan data. 5. Sistem menampilkan halaman daftar soal tes serta pesan berhasil “soal berhasil ditambah”. |
| ***Alternative Flows*** | 3.1 Jika terdapat data inputan yang kosong, maka sistem akan menampilkan pesan pemberitahuan “harap isi bidang ini”. |
| ***Post Conditions*** | Halaman daftar soal serta pesan berhasil “soal berhasil ditambah” berhasil ditampilkan. |

**Tabel 4.50 Use Case Scenario Edit Soal**

|  |  |
| --- | --- |
| Edit soal (PSB\_F\_43\_00) | |
| ***Aktor*** | Panitia |
| ***Objective*** | Fungsi untuk mengubah soal tes |
| ***Preconditions*** | Halaman daftar soal tes sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol edit dalam kolom aksi. 2. Sistem menampilkan halaman edit soal yang berisi data mapel, soal, opsi a, opsi b, opsi c, opsi d, kunci jawaban dan bobot nilai. 3. Aktor mngubah data yang ingin diubah. 4. Sistem melakukan validasi dan menyimpan data perubahan. 5. Sistem menampilkan halaman daftar soal serta pesan berhasil “soal berhasil diperbaharui”. |
| ***Alternative Flows*** | - |
| ***Post Conditions*** | Halaman daftar soal tes serta pesan berhasil “soal berhasil diperbaharui” berhasil ditampilkan. |

**Tabel 4.51 Use Case Scenario Hapus Soal**

|  |  |
| --- | --- |
| Hapus soal (PSB\_F\_44\_00) | |
| ***Aktor*** | Panitia |
| ***Objective*** | Fungsi untuk menghapus soal tes |
| ***Preconditions*** | Halaman daftar soal sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol hapus dalam kolom aksi. 2. Sistem menampilkan pesan konfirmasi “yakin untuk menghapus soal?” 3. Aktor menekan tombol ya. 4. Sistem melakukan proses hapus data. 5. Sistem menampilkan halaman daftar soal tes serta pesan berhasil “soal berhasil dihapus”. |
| ***Alternative Flows*** | 3.1 Jika aktor menekan tombol tidak, maka sistem akan mengembalikan halaman daftar soal tes. |
| ***Post Conditions*** | Halaman daftar soal tes serta pesan berhasil “soal berhasil dihapus” berhasil ditampilkan. |

**Tabel 4.52 Use Case Scenario Lihat Daftar Tes**

|  |  |
| --- | --- |
| Lihat daftar tes (PSB\_F\_45\_00) | |
| ***Aktor*** | Panitia |
| ***Objective*** | Fungsi unttuk melihat daftar tes seleksi |
| ***Preconditions*** | Aktor terdefinisi sebagai panitia |
| ***Main Flows*** | 1. Aktor menekan tombol tes seleksi. 2. Sistem menampilkan halaman daftar tes seleksi dengan data tampilan nomor, nama tes, jumlah soal, tanggal mulai, durasi, status dan aksi, serta tombol edit, hapus dan nonaktif dalam kolom aksi. 3. Sistem menampilkan menu soal, tes seleksi dan hasil tes. 4. Sistem menampilkan tombol buat tes dan kolom pencarian. |
| ***Alternative Flows*** | 1.1 Jika belum ada data dalam database, maka sistem akan menampilkan pesan pemberitahuan “No data available in table”. |
| ***Post Conditions*** | Halaman daftar tes berhasil ditampilkan. |

**Tabel 4.53 Use Case Scenario Buat Tes**

|  |  |
| --- | --- |
| Buat tes (PSB\_F\_46\_00) | |
| ***Aktor*** | Panitia |
| ***Objective*** | Fungsi untuk menambah tes seleksi |
| ***Preconditions*** | Halaman daftar tes sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol buat tes. 2. Sistem menampilkan halaman buat tes dengan data inputan nama tes, jumlah soal, tanggal mulai, tanggal berakhir dan durasi. 3. Sistem menampilkan petunjuk pengisian data dan data pembuat tes. 4. Aktor mengisi data secara lengkap dan menekan tombol buat. 5. Sistem melakukan validasi dan menyimpan data. 6. Sistem menampilkan halaman daftar tes serta pesan berhasil “tes berhasil ditambahkan”. |
| ***Alternative Flows*** | * 1. Jika jumlah soal melebihinjumlah soal dari daftar soal, maka sistem akan mengembalikan halaman daftar tes serta pesan kesalahan “gagal menambah, jumlah soal melebihi jumlah soal yang ada. Total jumlah soal sejumlah ...”.   2. Jika aktor menekan tombol batal, maka sistem akan mengembalikan halaman daftar tes. |
| ***Post Conditions*** | Halaman daftar tes serta pesan berhasil “tes berhasil ditambah” berhasil ditampilkan. |

**Tabel 4.54 Use Case Scenario Edit Tes**

|  |  |
| --- | --- |
| Edit tes (PSB\_F\_47\_00) | |
| ***Aktor*** | Panitia |
| ***Objective*** | Fungsi untuk mengubah data tes seleksi |
| ***Preconditions*** | Halaman daftar tes sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol edit dalam kolom aksi. 2. Sistem menampilkan halaman buat tes yang berisi data nama tes, jumlah soal, tanggal mulai, tanggal berakhir dan durasi. 3. Aktor mengubah data yang diinginkan dan menekan tombol simpan. 4. Sistem melakukan validasi data dan menyimpan perubahan data. 5. Sistem menampilkan halaman daftar tes serta pesan berhasil “Tes berhasil diperbaharui”. |
| ***Alternative Flows*** | 3.1 Jika aktor menekan tombol batal, maka sistem akan menampilkan kembali halaman daftar tes. |
| ***Post Conditions*** | Halaman daftar tes serta pesan berhasil “tes berhasil diperbaharui” berhasil ditampilkan. |

**Tabel 4.55 Use Case Scenario Hapus Tes**

|  |  |
| --- | --- |
| Hapus tes (PSB\_F\_48\_00) | |
| ***Aktor*** | Panitia |
| ***Objective*** | Fungsi untuk menghapus data tes |
| ***Preconditions*** | Halaman daftar tes sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol hapus dalam kolom aksi. 2. Sistem menampilkan pesan konfirmasi “yakin untuk menghapus tes?”. 3. Aktor menekan tombol ya. 4. Sistem melakukan proses hapus data. 5. Sistem menampilkan halaman daftar tes serta pesan berhasil “tes berhasil dihapus”. |
| ***Alternative Flows*** | 3.1 Jika aktor menekan tombol batal, maka sistem akan mengembalikan halaman daftar tes. |
| ***Post Conditions*** | Tes seleksi berhasil dihapus serta pesan berhasil “tes berhasil dihapus” berhasil ditampilkan. |

**Tabel 4.56 Use Case Scenario Nonaktif Tes**

|  |  |
| --- | --- |
| Nonaktif tes (PSB\_F\_49\_00) | |
| ***Aktor*** | Panitia |
| ***Objective*** | Fungsi untuk mengubah status tes menjadi tidak aktif |
| ***Preconditions*** | Halaman daftar tes udah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol nonaktif dalam kolom aksi.  2. Sistem melakukan proses update status tes seleksi menjadi nonaktif. |
| ***Alternative Flows*** | - |
| ***Post Conditions*** | Status tes seleksi berhasil menjadi nonaktif. |

**Tabel 4.57 Use Case Scenario Lihat Hasil Tes Seleksi**

|  |  |
| --- | --- |
| Lihat hasil tes seleksi (PSB\_F\_50\_00) | |
| ***Aktor*** | Panitia |
| ***Objective*** | Fungsi untuk melihat hasil tes tiap calon santri |
| ***Preconditions*** | Aktor terdefinisi sebagai panitia |
| ***Main Flows*** | 1. Aktor menekan tombol hasil tes 2. Sistem menampilkan halaman hasil tes dengan data tampilan nama, nilai dan aksi, serta tombol detail dan hapus dalam kolom aksi. 3. Sistem menampilkan kolom pencarian |
| ***Alternative Flows*** | 1.1 Jika belum ada data dalam basisdata, maka sistem menampilkan pesan pemberitahuan “No data available in table”. |
| ***Post Conditions*** | Halaman hasil tes berhasil ditampilkan. |

**Tabel 4.58 Use Case Scenario Lihat Detail Jawaban**

|  |  |
| --- | --- |
| Lihat detail jawaban (PSB\_F\_51\_00) | |
| ***Aktor*** | Panitia |
| ***Objective*** | Fungsi untuk melihat detail jawaban calon santri |
| ***Preconditions*** | Halaman hasil tes sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol detail dalam kolom aksi. 2. Sistem menampilkan halaman detail jawaban dengan data tampilan soal, jawaban dan kunci soal. 3. Sistem menampilkan jumlah jawaban yang benar dan salah. |
| ***Alternative Flows*** | 1.1 Jika belum ada data dalam basisdata, maka sistem menampilkan pesan pemberitahuan “No data available in table”. |
| ***Post Conditions*** | Halaman detail jawaban berhasil ditampilkan. |

**Tabel 4.59 Use Case Scenario Hapus Hasil Tes**

|  |  |
| --- | --- |
| Hapus hasil tes (PSB\_F\_52\_00) | |
| ***Aktor*** | Panitia |
| ***Objective*** | Fungsi untuk menghapus data hasil tes calon santri |
| ***Preconditions*** | Halaman hasil tes sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol hapus dalam kolom aksi. 2. Sistem menampilkan pesan konfirmasi “yakin untuk menghapus hasil tes?”. 3. Aktor menekan tombol ya. 4. Sistem melakukan proses hapus data. 5. Sistem menampilkan halaman hasil tes serta pesan berhasil “hasil tes berhasil dihapus”. |
| ***Alternative Flows*** | 3.1 Jika aktor menekan tombol tidak, maka sistem akan mengembalikan halaman hasil tes. |
| ***Post Conditions*** | Halaman hasil tes serta pesan berhasil “hasil tes berhasil diapus” berhasil ditampilkan. |

**Tabel 4.60 Use Case Scenario Nilai Tes Wawancara**

|  |  |
| --- | --- |
| Nilai tes wawancara (PSB\_F\_53\_00) | |
| ***Aktor*** | Panitia |
| ***Objective*** | Fungsi untuk menginput nilai tes wawancara calon santri |
| ***Preconditions*** | Aktor terdefinisi sebagai panitia |
| ***Main Flows*** | 1. Aktor menekan tombol wawancara. 2. Sistem menampilkan halaman tes wawancara dengan data tampilan nomor, kode daftar, nama, program studi dan nilai serta tombol nilai dalam kolom aksi. 3. Sistem menampilkan kolom pencarian data. 4. Aktor menekan tombol nilai dalam kolom aksi. 5. Sistem menampilkan form penilaian dengan data input nama, kode daftar dan nilai wawancara. 6. Aktor mengidi data secara lengkap dan menekan tombol simpan. 7. Sistem melakukan validasi data dan menyimpan data. 8. Sistem menampilkan halaman tes wawancara serta pesan berhasil “nilai berhasil diinput”. |
| ***Alternative Flows*** | * 1. Jika belum ada data dalam basisdata, maka sistem menampilkan pesan pemberitahuan “No data available in table”.   6.1 Jika aktor menekan tombol batal, maka sistem akan mengembalikan halaman tes wawancara. |
| ***Post Conditions*** | Nilai tes wawancara berhasip diinput. |

**Tabel 4.61 Use Case Scenario Nilai Kelulusan**

|  |  |
| --- | --- |
| Nilai kelulusan (PSB\_F\_54\_00) | |
| ***Aktor*** | Panitia |
| ***Objective*** | Fungsi untuk menentukan hasil akhir penerimaan |
| ***Preconditions*** | Aktor terdefinisi sebagai panitia |
| ***Main Flows*** | 1. Aktor menekan tombol akumulasi. 2. Sistem menampilkan halaman akumulasi dengan data tampilan nomor, kode daftar, nama, prodi pilihan, tes akademik, tes wawancara, total nilai, status penerimaan dan aksi, serta tombol lulus dan tidak lulus dalam kolom aksi. 3. Aktor menekan tombol lulus atau tidak lulus. 4. Sistem melakukan proses update data status penerimaan. |
| ***Alternative Flows*** | 1.1 Jika belum ada data dalam basisdata, maka sistem menampilkan pesan pemberitahuan “No data available in table”. |
| ***Post Conditions*** | Kelulusan calon santri berhasil dinilai. |

**Tabel 4.62 Use Case Scenario Lihat Kirim Pesen**

|  |  |
| --- | --- |
| Kirim pesan (PSB\_F\_55\_00) | |
| ***Aktor*** | SMS gateaway |
| ***Objective*** | Fungsi untuk mengirim pesan melaui *whatsApp* |
| ***Preconditions*** | Nomor handphone dan pesan telah didapatkan dari jalannya fungsi verifikasi data dan tambah admin |
| ***Main Flows*** | 1. Aktor mengirim pesan dengan nomor hp tujuan penerima serta isi pesan yang telah dituliskan. 2. Sistem menampilkan pesan berhasil “password telah dikirim, pengguna berhasil ditambahkan” saat fungsi tambah admin dijalankan dan “pesan telah dikirim, status verifikasi berhasil diperbaharui” saat fungsi verifikasi data dijalankan. |
| ***Alternative Flows*** | - |
| ***Post Conditions*** | Pesan berhasil dikirim ke nomor hp penerima. |

**Tabel 4.63 Use Case Scenario Lihat Pengumuman**

|  |  |
| --- | --- |
| Lihat pengumuman (PSB\_F\_55\_00) | |
| ***Aktor*** | Peserta |
| ***Objective*** | Fungsi untuk melihat hasil akhir proses penerimaan santri baru |
| ***Preconditions*** | Aktor terdefinisi sebagai peserta |
| ***Main Flows*** | 1. Aktor menekan tombol dashboard. 2. Sistem menampilkan halaman dashboar serta pesan kelulusan “selamat, anda dinyatakan lulus dan diterima menjadi siswa mas nurul as'adiyah callaccu tahun ... ”. |
| ***Alternative Flows*** | 1.1 Jika aktor dinayatakan tidak lulus, maka sistem akan menampilkan pesan kegagalan “mohon maaf, anda dinyatakan tidak lulus menjadi siswa mas nurul as'adiyah callaccu tahun ... “. |
| ***Post Conditions*** | Pesan kelulusan berhasil ditampilkan. |

**Tabel 4.64 Use Case Scenarion Pengaturan Akun**

|  |  |
| --- | --- |
| Pengaturan akun (PSB\_F\_55\_00) | |
| ***Aktor*** | Peserta |
| ***Objective*** | Fungsi untuk mengubah data akun |
| ***Preconditions*** | Aktor terdefinisi sebagai peserta |
| ***Main Flows*** | 1. Aktor menekan foto akun. 2. Sistem menampilkan menu pengaturan aku dan keluar. 3. Aktor menekan tombol pengaturan akun. 4. Sistem menampilkan halaman pengaturan aku yang berisi data nomor pendaftaran, NISN, nama lengkap dan email. 5. Aktor mengubah data yang diinginkan dan menekan tombol simpan. 6. Sistem melakukan validasi dan menyimpan perubahan data. 7. Sistem menampilkan halaman dashboard serta pesan berhasil “perubahan berhasil dilakukan”. |
| ***Alternative Flows*** | 1.1 Jika terdapat data yang kosong, maka sistem akan menampilkan kembali halaman dashboard dengan pesan kesalahan “perubahan gagal dilakukan”. |
| ***Post Conditions*** | Halaman dashboard serta pesan berhasil “perubahan berhasil dilakukan” berhasil ditampikan. |

|  |  |
| --- | --- |
| Cetak kartu peserta (PSB\_F\_55\_00) | |
| ***Aktor*** | Peserta |
| ***Objective*** | Fungsi untuk mencetak kartu peserta PSB |
| ***Preconditions*** | 1. Aktor terdefinisi sebagai peserta 2. Halaman data peserta sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol cetak kartu peserta. 2. Sistem menampilkan preview kartu yang akan dicetak, dengan data yang dicetak diantaranya foto, nomor pendaftaran, nama, NISN, asal sekolah dan program studi. 3. Aktor menekan tombol cetak. 4. Sistem melakukan proses cetak kartu peserta. |
| ***Alternative Flows*** | - |
| ***Post Conditions*** | Kartu peserta berhasil dicetak. |

**Tabel 4.65 Use Case Scenario Edit Data**

|  |  |
| --- | --- |
| Edit data (PSB\_F\_55\_00) | |
| ***Aktor*** | Peserta |
| ***Objective*** | Fungsi untuk mengubah data pendaftaran |
| ***Preconditions*** | Halaman data peserta sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol edit. 2. Sistem menampilkan halaman edit data yang berisi data diri, orangtua dan wali. 3. Aktor mengubah data yang diinginkan dan menekan tombol simpan. 4. Sistem melakukan validasi dan menyimpan perubahan data. 5. Sistem menampilkan halaman data peserta serta pesan berhasil “data berhasil diperbaharui” |
| ***Alternative Flows*** | * 1. Jika aktor mengunggah foto lebih dari 2MB atau fengan format file selain gambar, maka sistem akan menampilkan halaman data peserta serta pesan kesalahan “file harus berupa gambar denga ukuran 2MB, silahkan unggah ulang foto”.   2. Jika aktor menekan tombol batal, maka sistem akan menampilkan kembali halaman data peserta. |
| ***Post Conditions*** | Halaman data peserta serta pesan berhasil “data berhasil diperbaharui” berhasil ditampilkan. |

**Tabel 4.66 Use Case Scenario Upload Berkas**

|  |  |
| --- | --- |
| Upload berkas (PSB\_F\_55\_00) | |
| ***Aktor*** | Peserta |
| ***Objective*** | Fungsi untuk mengunggah berkas prasyarat PSB |
| ***Preconditions*** | Halaman data peserta sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol berkas prasyarat. 2. Sistem halaman daftar berkas prasyarat dengan data tampilan nama berkas, status dan aksi, serta tombol hapus dan lihat dalam kolom aksi. 3. Sistem menampilkan form unggah berkas 4. Aktor menekan tombol pilih file. 5. Aktor memilih berkas yang ingin diunggah dan menekan tombol unggah. 6. Sistem melakukan validasi dan menyimpan berkas. 7. Sistem menampilkan form unggah berkas serta pesan berhasil “berkas berhasil diunggah”. |
| ***Alternative Flows*** | 5.1 Jika aktor mengunggah berkas dengan format selain pdf atau ukuran berkas melebihi 2MB, maka sistem akan menampilkan pesan kesalahan “file harus berupa pdf dengan ukuran max 2MB, silahkan unggah ulang file”. |
| ***Post Conditions*** | Pesan berhasil “berkas berhasil diunggah” berhasil ditampilkan. |

|  |  |
| --- | --- |
| Hapus berkas (PSB\_F\_55\_00) | |
| ***Aktor*** | Peserta |
| ***Objective*** | Fungsi untuk menghapus berkas yang sudah diunggah |
| ***Preconditions*** | Halaman daftar berkas prasyarat sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol hapus dalam kolom aksi. 2. Sistem menampilkan pesan konfirmasi “yakin untuk menghapus berkas?”. 3. Aktor menekan tombol ya. 4. Sistem melakukan proses hapus data. 5. Sistem menampilkan kembali halaman daftar berkas prasyarat serta pesan berhasil “berkas berhasil dihapus”. |
| ***Alternative Flows*** | 3.1 Jika aktor menekan tombol batal, maka Sistem akan mengembalikan halaman daftar berkas prasyarat. |
| ***Post Conditions*** | Halaman daftar berkas prasyarat serta pesan berhasil “berkas berhasil dihapus” berhasil ditampilkan. |

**Tabel 4.67use Case Scenario Bayar Tagihan**

|  |  |
| --- | --- |
| Bayar tagihan (PSB\_F\_55\_00) | |
| ***Aktor*** | Peserta dan payment gateaway |
| ***Objective*** | Fungsi untuk melakukan pembayaran tagihan PSB |
| ***Preconditions*** | Aktor terdefinisi sebagai peserta |
| ***Main Flows*** | 1. Aktor menekan tombol pembayaran. 2. Sistem menampilkan daftar tagihan pembayaran dan riwayat pembayaran. 3. Aktor menekan tombol bayar. 4. Sistem menampilkan form pembayaran dengan data inputan nominal yang ingin dibayar. 5. Aktor menekan tombol bayar sekarang. 6. Sistem menampilkan detail pembayaran diantaranya total, nama dan email. 7. Aktor menekan tombol continue. 8. Aktor memilih metode pembayaran bank transfer atau indomaret. 9. Sistem menampilkan pilihan bank. 10. Aktor memilih bank yang diinginkan. 11. Sistem menampilkan total pembayaran serta tata cara pembayaran. 12. Aktor menekan tombol see account number. 13. Sistem menampilkan nomor virtual account. 14. Aktor menekan tombol please complement payment. 15. Sistem menyimpan data pembayaran. 16. Sistem menampilkan daftar tagihan serta pesan berhasil “Berhasil, silahkan lakukan pembayaran sebelum 24jam”. |
| ***Alternative Flows*** | 5.1 Jika terjadi kesalahan dalam proses pembayaran, maka Sistem menampilkan pesan kesalahan, kami tidak menemukan transaksi anda. |
| ***Post Conditions*** | Halaman daftar tagihan serta pesan berhasil “Berhasil, silahkan lakukan pembayaran sebelum 24jam” berhasil ditampilkan. |

**Tabel 4.68 Use Case Scenario Update Status Pembayaran**

|  |  |
| --- | --- |
| Update status pembayaran (PSB\_F\_55\_00) | |
| ***Aktor*** | Payment gateaway |
| ***Objective*** | Fungsi untuk mengubah status pembayaran |
| ***Preconditions*** | Aktor peserta sudah melakukan proses pembayaran |
| ***Main Flows*** | 1. Sistem mengirim api request 2. Aktor memberi api response 3. Aktor memproses data dan merespon dengan status pembayaran. 4. Sistem menampilkan halaman daftar tagihan serta status pembayaran. |
| ***Alternative Flows*** | - |
| ***Post Conditions*** | Status pembayaran berhasil diupdate |

**Tabel 4.69 Use Case Scenario Lihat Tes Seleksi**

|  |  |
| --- | --- |
| Lihat tes seleksi (PSB\_F\_55\_00) | |
| ***Aktor*** | Peserta |
| ***Objective*** | Fungsi untuk melihat informasi tes seleksi |
| ***Preconditions*** | Aktor terdefinisi sebagai peserta |
| ***Main Flows*** | 1. Aktor menekan tombol tes seleksi 2. Sistem menampilkan halaman tes seleksi serta pesan peringatan syarat untuk mengikuti tes. 3. Sistem menampilkan informasi data diri diantaranya nama, kode pendaftaran, email dan prodi pilihan, serta informasi tes diantaranya nama tes, jumlah soal, durasi, tanggal mulai dan tanggal berakhir. 4. Sistem menampilkan countdown mulai tes. 5. Sistem menampilkan tombol mulai |
| ***Alternative Flows*** | * 1. Jika belum ada tes, maka Sistem akan menampilkan pesan pemberitahuan “belum ada tes”.   2. Jika tes sudah dimulai, maka Sistem akan menampilkan pemberitahuan “tes dimulai sekarang”.   3. Jika Aktor sudah mengikuti tes, maka Sistem akan menampilkan pemberitahuan “anda telah mengikuti tes seleksi”.   4. Jika tes seleksi berakhir, maka Sistem akan menampilkan pemberitahuan “Tes Berakhir”.   5. Jika data diri belum berstatus terima, maka Sistem akan menampilkan pemberitahuan “Anda belum memenuhi syarat untuk mengikuti tes”. |
| ***Post Conditions*** | Status pembayaran berhasil diupdate |

**Tabel 4.70 Use Case Scenario Kerjakan Tes**

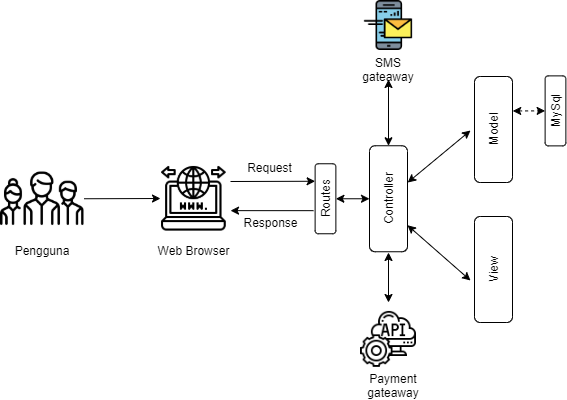
|  |  |
| --- | --- |
| Kerjakan tes (PSB\_F\_55\_00) | |
| ***Aktor*** | Peserta |
| ***Objective*** | Fungsi untuk menjawab soal tes seleksi |
| ***Preconditions*** | Halaman tes seleksi sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol mulai. 2. Sistem menampilkan halaman soal. 3. Aktor memilih jawaban yang diinginkan dan menekan tombol selanjutnya. 4. Sistem menampilkan halaman soal selanjutnya, seterusnya hingga soal terakhir. 5. Sistem menampilkan halaman tes seleksi serta hasil tes diantaranya nilai, jumlah salah dan jumlah benar. |
| ***Alternative Flows*** | 1.1 Jika Aktor menekan tombol mulai saat countdown masih berjalan, maka Sistem menampilkan pemberitahuan “tes seleksi belum dimulai”. |
| ***Post Conditions*** | Hasil tes seleksi berhasil ditampilkan. |

# **PERANCANGAN DAN IMPLEMENTASI**

## **5.1 Perancangan Sistem**

Perancangan adalah tahap untuk menggambarkan arsitektur sistem yang akan dibangun, dengan berpedoman pada daftar kebutuhan yang telah di definisikan pada proses sebelumnya yaitu analisis kebutuhan. Adapun rangkaian tahap yang akan dilakukan yaitu: perancangan arsitektur, perancangan database, perancangan komponen dan perancangan antarmuka pengguna.

### 5.1.1 Perancangan Arsitektur



**Gambar 5.1 Perancangan Arsitektur Sistem**

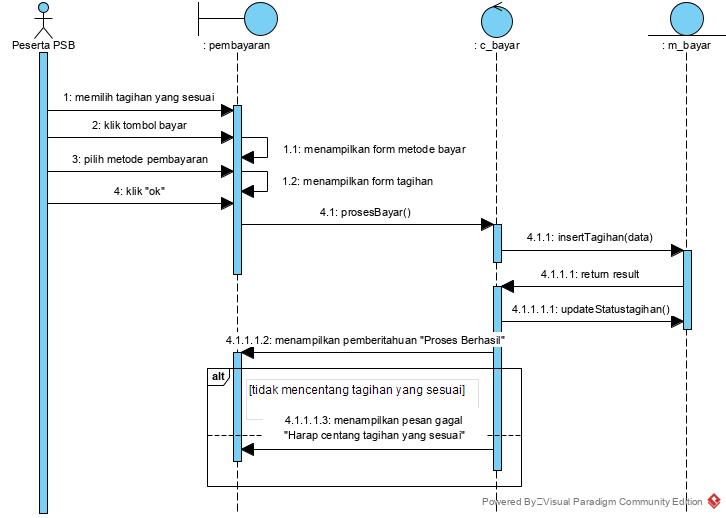
Perancangan arsitektur merupakan proses untuk mengambarkan hubungan antar komponen dalam sistem yang akan dibangun, dalam penelitian ini sistem akan dikembangkan dengan berbasis website dengan menggunakan *framework codeigniter* yang dapat membuat performa website jadi lebih baik. *Codeigniter* menganut konsep MVC (Model, View Controller) yang memisahkan 3 komponen utama yaitu model, view dan controller sehingga struktur kode dapat menjadi lebih baik dan teratur selain itu *maintenance code* juga dapat lebih muda.

Saat pengguna mengakses url melalui web browser, maka hal tersebut dinamakan membuat atau mengirim request ke Sistem dan routes adalah komponen pertama yang akan di akses, routes adalah komponen yang menentukan controller mana yang akan dijalankan. Kemudian, request akan diteruskan ke controller, controller adalah komponen utama yang akan mengendalikan alur proses dalam sistem yang dibangun dan juga sebagai penghubung antara model dan view. Controller akan menentukan apakah akan menampilkan view atau data dari model (database), view merupakan komponen yang berfungsi untuk menampilkan halaman atau mengatur tampilan halam dari Sistem yang dibangun dan model merupakan komponen yang digunakan ketika akan berinteraksi dengan data yang terdapat dalam database. Setelah itu, view atau model akan memberikan response dan diteruskan melalui controller lagi dan ditampilkan kepada pengguna.

### 5.1.2 Perancangan *Sequence Diagram*

Pemodelan *sequence diagram* menggambarkan hubungan antar objek dalam sistem, pemodelan ini dirancang dengan berpedoman pada use case scenario yang telah dibuat pada analisis kebutuhan sebelumnya. Jumlah *sequence diagram* akan sama dengan jumlah *use case secanario* yang telah di definisikan, tetapi dalam dokumentasi ini hanya akan menampilkan 3 *sequence diagram* yaitu proses pembayaran, verifikasi berkas dan proses tes seleksi sebagai sampel.

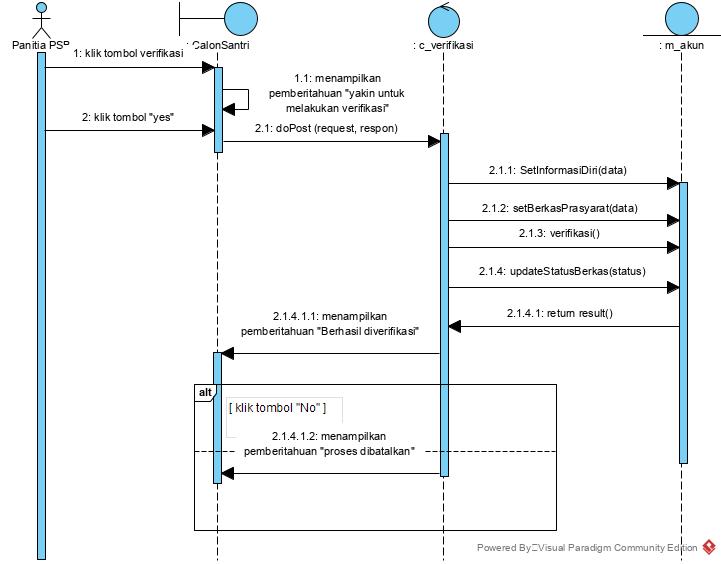
1. *Sequence diagram* Pembayaran



**Gambar 5.2 Sequence diagram Pembayaran**

Proses pembayaran diawali dengan aktor memilih tagihan pembayaran yang sesuai dengan gender yang terdapat pada *boundary* pembayaran, lalu sistem akan menghitung total tagihan yang harus dibayar dan aktor memencet tombol bayar. Lalu sistem akan menampilkan form metode bayar dan aktor memilih metode pembayaran yang diingikan, setelah itu sistem menampilkan form total tagihan beserta rekening tujuan kemudian aktor memencet tombol ok. Sistem akan menjalankan fungsi prosesBayar() pada *controller* c\_bayar , kemudian c\_bayar akan memerintahkan *model* m\_pembayaran untuk menjalankan fungsi insertTagihan. Kemudian m\_pembayaran akan mengembalikan hasil query ke c\_bayar, lalu memerintahkan m\_pembayaran untuk menjalankan fungsi updateStatusTagihan(). Sistem akan menampilkan status pembayaran jika telah melakukan proses bayar dan akan menampilkan pesan gagal jika belum memilih tagihan pembayaran. Hasil perancangan *sequence diagram* pembayaran dapat dilihat pada Gambar 5.1.

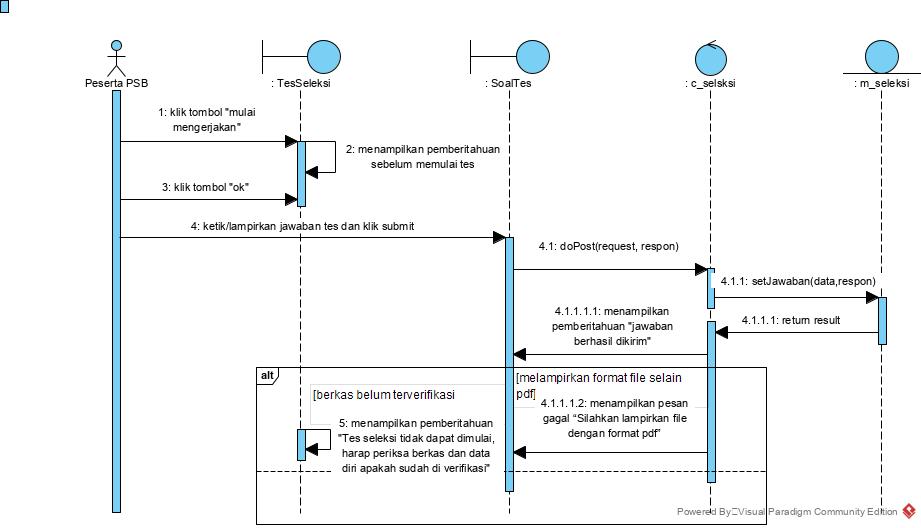
2. *Sequence diagram* Verifikasi Berkas



**Gambar 5.3 Sequence diagram Verifikasi Berkas**

Setelah peserta PSB mengunggah berkas prasyarat maka saatnya panitia PSB untuk melakukan verifikasi berkas, proses dimulai saat aktor memencet tombol verifikasi pada *boundary* CalonSantri kemudian sistem menampilkan pemberitahuan “Yakin untuk melakukan verifikasi?” dan aktor memencet tombol “yes”, maka *controller* c\_dataCalonSantri akan menjalankan fungsi doPost dengan parameter request dan respon untuk memproses verifikasi, kemudian c\_dataCalonSantri akan memerintahkan *model* m\_calonSantri untuk melakukan setInformasiDiri(data), setBerkasPrasyarat(data), verifikasi() dan updateStatusBerkas(status) akan mengubah status pendaftaran calon santri, setelah dijalankan m\_calonSantri akan mengembalikan respon kepada c\_dataCalonSantri berupa pemberitahuan bahwa verifikasi berhasil dilakukan. Akan tetapi jika aktor memencet tombol “no” maka akan ditampilkan pemberitahuan “proses dibatalkan”. Hasil perancangan *sequence diagram* Verifikasi Berkas dapat dilihat pada Gambar 5.2.

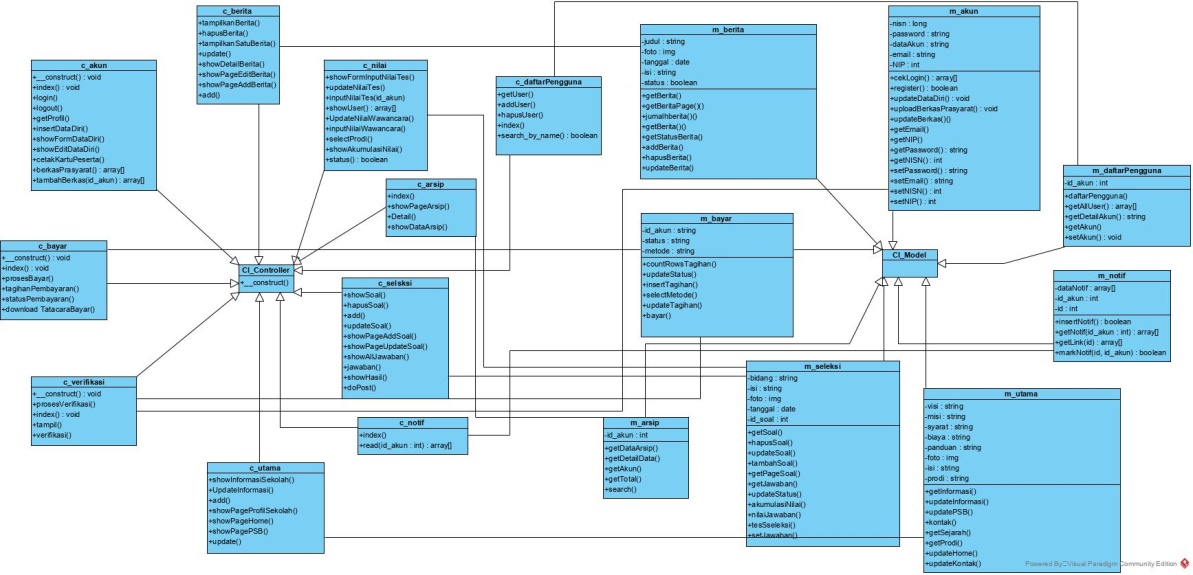
3. *Sequence diagram* Tes seleksi



**Gambar 5.4 Sequence diagram Tes Seleksi**

Tes seleksi akan dimulai saat aktor mencet tombol “Mulai Mengerjakan” yang terdapat pada *boundary* TesSeleksi lalu sistem menampilkan pemberitahuan penting terkait tes dan aktor memencet tombol “ok”. Akan ditampilkan *boundary* SoalTes, aktor mengetikkan jawaban atau melampirkan file atas jawaban dari pertanyaan tes, setelah itu *controller* c\_tesAkademik akan menjalankan fungsi doPost(request, respon) dan memerintahkan *model* m\_SeleksiAkademik untuk menjalankan fungsi setJawaban(data,respon), setelah model dijalankan maka akan mengembalikan respon kepada c\_tesAkademik berupa pemberitahuan “Jawaban berhasil dikirim”. Saat aktor melampirkan file selain format pdf c\_doPost akan dijalankan dan menampilkan pemberitahuan “silahkan lampirkan file dengan format pdf”, namun jika berkas belum terverifikasi diawal maka akan ditampilkan pemberitahuan “Tes seleksi tidak dapat dimulai harap periksa berkas dan data diri apakah sudah diverifikasi”. Hasil perancangan *sequence diagram* tes seleksi dapat dilihat pada Gambar 5.3.

### 5.1.3 Perancangan *Class Diagram*



**Gambar 5.5 Perancangan Class Diagram**

Class Diagram dirancang untuk mengetahui kelas-kelas apa yang nantinya terdapat dalam sistem, class diagram juga berfungsi untuk dapat mengetahui relasi antar kelas. Pada pengembangan PSB Online ini menggunakan *framework* Codeigniter yang menganut konsep MVC (Model, View Controller), di mana controller yang bertugas sebagai penghubung antara model dan view, jika pengguna ingin menampilkan sebuah halaman sistem akan melakukan pemanggilan ke controller kemudian diteruskan ke model kemudian melalui controller barulah ditampilkan ke pengguna.

CI\_Controller pada sistem terdiri dari c\_notif, c\_utama, c\_verifikasi, c\_daftarPengguna, c\_bayar, c\_soal, c\_arsip, c\_nilai, c\_berita dan c\_akun, tiap controller akan meng-*extends* kelas yang terdapat pada CI\_Model, kelas yang terdapat pada CI\_Model diantaranya m\_utama, m\_notif, m\_arsip, m\_bayar, m\_soal, m\_berita, m\_daftarPengguna dan m\_akun. CI\_controller akan berasosiasi dengan CI\_Model untuk mendapatkan data yang diinginkan, setelah didapatkan data tersebut akan dibawa ke controller untuk ditampilkan pada *view*. Hasil dari perancangan *Class Diagram* dapat dilihat pada Gambar 5.4.

### 5.1.4 Perancangan Komponen

Perancangan komponen adalah proses mendefinisikan alur pada algoritma yang terdapat dalam sebuah metode dalam sistem yang dikembangkan, dalam proses perancangan ini akan menghasilkan alur algoritma dalam bentuk verifikasi(), bayar() dan tesSeleksi().

#### 5.1.4.1 Perancangan Komponen verifikasi()

Nama Kelas : c\_verifikasi

Nama Fungsi : verifikasi()

Deskripsi : verifikasi berkas prasayarat dan pembayaran

**Tabel 5.1 Perancangan verifikasi()**

|  |  |
| --- | --- |
| No | Perancangan Komponen verifikasi() |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | START  Var Verifikasi = array(  'foto' = berkas foto peserta PSB,  'ijazah' = berkas ijazah peserta PSB,  'SHUN' = berkas SHUN peserta PSB,  'KK' = berkas KK peserta PSB,  'KTP\_ortu' = berkas KTP\_ortu peserta PSB,  'akta\_kelahran' = berkas akta\_kelahiran peserta PSB,  'ket\_lulus' = berkas ket\_lulus peserta PSB,  'k\_NISN' = berkas NISN peserta PSB,  'KIP' = berkas KIP peserta PSB,  'KPS' = berkas KPS peserta PSB,  'SKKM' = berkas SKKM peserta PSB)  Var query = menambahkan nilai dari array verifikasi pada tabel tb\_verifikasi  IF (verifikasi berhasil)  Var insert\_id = nilai id verifikasi yang baru saja dimasukkan  Return insert\_id  ELSE  Return FALSE  END  Cetak “Verifikasi error” end if |

#### 5.1.3.2 Perancangan Komponen prosesBayar()

Nama Kelas : c\_bayar

Nama Fungsi : prosesBayar()

Deskripsi : melakukan pembayaran tagihan sekolah

**Tabel 5.2 Perancangan Komponen prosesBayar()**

|  |  |
| --- | --- |
| No | Perancangan Komponen prosesBayar() |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | Start  Inisialisasi set rules form\_validation untuk norek = required  Inisialisasi set rules form\_validation untuk atasnama = required  Inisialisasi set rules form\_validation untuk jumlah = required Inisialisasi set rules form\_validation untuk bukti = required    Inisialisasi variabel data = post form  if (form\_validation berjalan) then Inisialisasi variabel result = insertTagihan(data) ke m\_bayar  Inisialisasi variabel result2 = insertBuktiTransfer(data) ke m\_bayar  If (result bernilai > 0 dan result2 bernilai > 0) then  publishSolenoidOff(data[idUser])  updateStatus(data[idUser], ‘terkunci’, ‘transfer’) ke  m\_bayar    resetFlow(data[idUser]) ke m\_bayar  Cetak “Proses berhasil”  else  Cetak “Harap centang tagihan yang sesuai” end if  else  Cetak “Validation error” end if |

#### 5.1.3.3 Perancangan Komponen tesSeleksi()

Nama Kelas : m\_seleksi

Nama Fungsi : tesSeleksi()

Deskripsi : mengerjakan tes seleksi akademik

**Tabel 5.3 Perancangan Komponen tesSeleksi()**

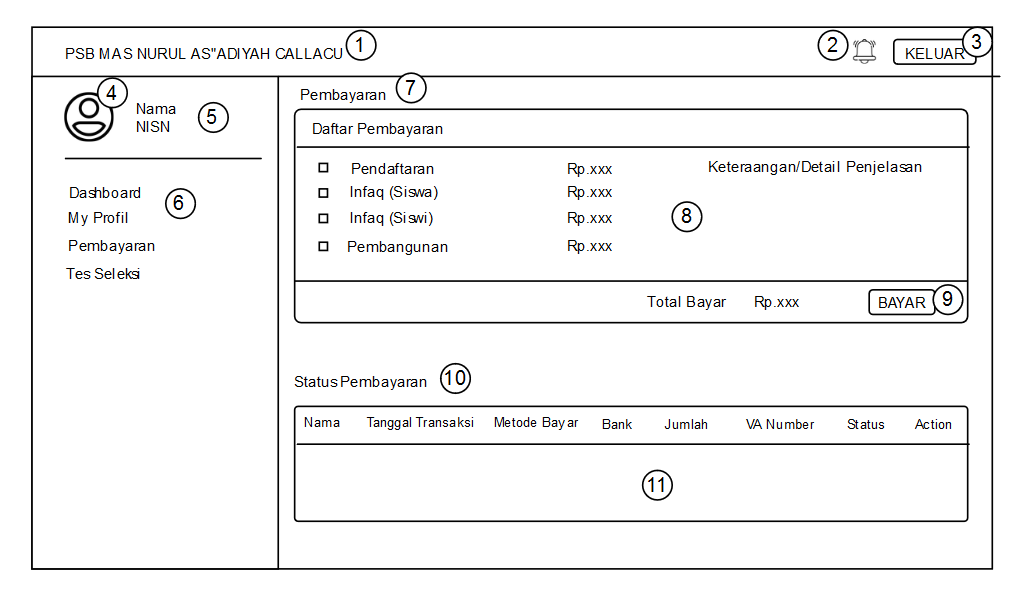
|  |  |
| --- | --- |
| No | Perancangan Komponen tesSeleksi() |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | Inisialisasi set rules form\_validation untuk norek = required  Inisialisasi set rules form\_validation untuk atasnama = required  Inisialisasi set rules form\_validation untuk jumlah = required Inisialisasi set rules form\_validation untuk bukti = required    Inisialisasi variabel data = post form  if (form\_validation berjalan) then Inisialisasi variabel result = insertTagihan(data) ke m\_bayar  Inisialisasi variabel result2 = insertBuktiTransfer(data) ke m\_bayar  If (result bernilai > 0 dan result2 bernilai > 0) then  publishSolenoidOff(data[idUser])  updateStatus(data[idUser], ‘terkunci’, ‘transfer’) ke  m\_bayar    resetFlow(data[idUser]) ke m\_bayar  Cetak “Proses berhasil”  else  Cetak “Harap centang tagihan yang sesuai” end if  else  Cetak “Validation error” end if |

### 5.1.5 Perancangan Basis Data

### 5.1.6 Perancangan Antarmuka

Perancangan antarmuka merpakan proses untuk mendesain tampilan pada sistem yang akan dikembangkan, sehingga memudahkan dalam proses implementasi kedalam bentuk kode nantinya. Berikut terdapat 3 contoh desain diantaranya halaman pembayaran, verifikasi berkas dan tes seleksi.

#### 5.1.6.1 Perancangan Antarmuka Pembayaran

****

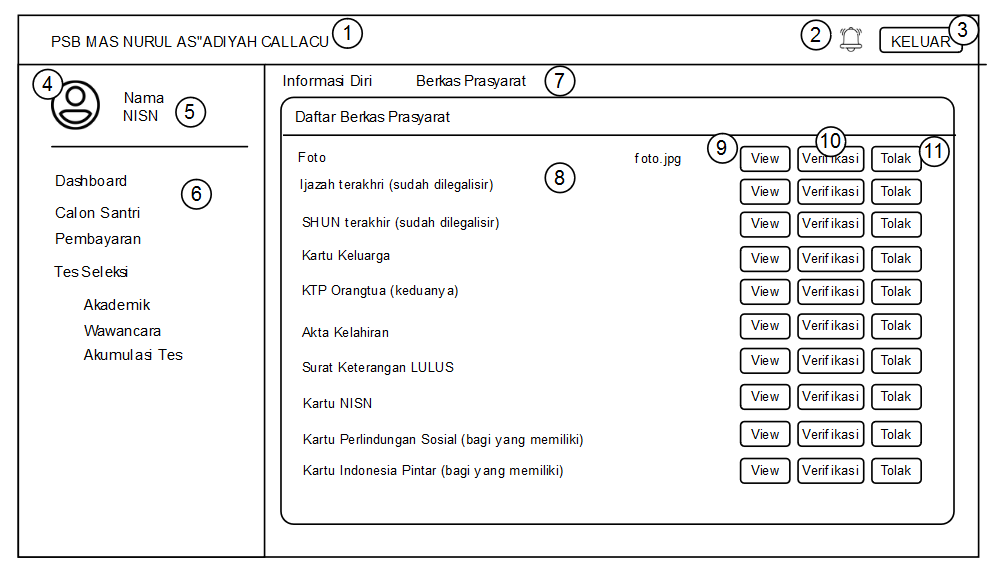
**Gambar 5.6 Perancangan Antarmuka Pembayaran**

Pada Gambar 5.4 adalah hasil rancangan antarmuka proses pembayaran, detail penjelasan terkait komponen penyusun yang terdapat didalamnya dijelaskan pada Tabel 5.1.

**Tabel 5.4 Penjelasan Antarmuka Pembayaran**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Objek** | **Tipe** | **Keterangan** |
| 1 | Judul sistem | *Label* | Nama sistem |
| 2 | Notifikasi | *Button* | Menampilkan dan menyembunyikan notifikasi |
| 3 | Keluar | *Button* | Menghilangkan hak akses pengguna pada sistem |
| 4 | Foto pengguna | *Image rounded* | Menampilkan foto profil pengguna |
| 5 | Data pengguna | *Label* | Menampilkan data penggguna berupa nama dan NISN |
| 6 | Daftar menu | *Sidebar menu* | Menampilkan daftar menu yang terdapat dalam sistem |
| 7 | Judul halaman | *Label* | Menampilkan nama halaman yang diakses |
| 8 | Tabel pembayaran | *Table* | Menampilkan daftar jenis pembayaran yang harus dibayar |
| 9 | Bayar | *Button* | Media untuk melanjutkan proses pembayaran |
| 10 | Judul table | *Lable* | Menampilkan judul tabel |
| 11 | Table status pembayaran | *Table* | Menampilkan status pembayaran terkini |

#### 5.1.6.2 Perancangan Antarmuka Verifikasi Berkas



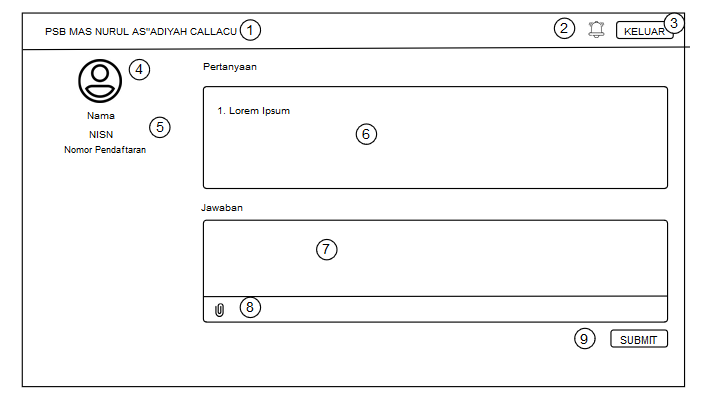
**Gambar 5.7 Perancangan Antar Muka Verifikasi Berkas**

Pada Gambar 5.6 adalah hasil rancangan antarmuka proses verifikasi berkas, detail penjelasan terkait komponen penyusun yang terdapat didalamnya dijelaskan pada Tabel 5.2.

**Tabel 5.5 Penjelasan Antarmuka Verifikasi Berkas**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Objek** | **Tipe** | **Keterangan** |
| 1 | Judul sistem | *Label* | Nama sistem |
| 2 | Notifikasi | *Button* | Menampilkan dan menyembunyikan notifikasi |
| 3 | Keluar | *Button* | Menghilangkan hak akses pengguna pada sistem |
| 4 | Foto pengguna | *Image rounded* | Menampilkan foto profil pengguna |
| 5 | Data pengguna | *Label* | Menampilkan data penggguna berupa nama dan NISN |
| 6 | Daftar menu | *Sidebar menu* | Menampilkan daftar menu yang terdapat dalam sistem |
| 7 | Tombol informasi data pendaftaran | *Button* | Menampilkan data sesuai judul sub menu |
| 8 | Tabel daftar berkas prasyarat | *Table* | Menampilkan daftar berkas prasyarat |
| 9 | View | *Button* | Media untuk melihat isi berkas |
| 10 | Verifikasi | *Button* | Media untuk menyetujui berkas |
| 11 | Tolak | *Button* | Media untuk menolak berkas |

#### 5.1.6.3 Perancangan Antarmuka Tes Seleksi



**Gambar 5.8 Perancangan Antarmuka Tes Seleksi**

Pada Gambar 5.7 adalah hasil rancangan antarmuka proses verifikasi berkas, detail penjelasan terkait komponen penyusun yang terdapat didalamnya dijelaskan pada Tabel 5.3.

**Tabel 5.6 Penjelasan Antarmuka Tes Seleksi**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Objek** | **Tipe** | **Keterangan** |
| 1 | Judul sistem | *Label* | Nama sistem |
| 2 | Notifikasi | *Button* | Menampilkan dan menyembunyikan notifikasi |
| 3 | Keluar | *Button* | Menghilangkan hak akses pengguna pada sistem |
| 4 | Foto pengguna | *Image rounded* | Menampilkan foto profil pengguna |
| 5 | Data pengguna | *Label* | Menampilkan data penggguna berupa nama, NISN dan nomor pendaftaran |
| 6 | Pertanyaan | *Label* | Menampilkan daftar pertanyaan tes |
| 7 | Jawaban | *Input text* | Input yang berfungsi untuk mengetikkan jawaban |
| 8 | File | *Button* | Media untuk melampirkan file jawaban |
| 9 | Submit | *Button* | Media untuk mengirim jawaban tes seleksi |

## **5.2 Implementasi Sistem**

### 5.2.1 Spesifikasi Sistem

### 5.2.2 Implementasi Kode Program

### 5.2.3 Implementasi Basis Data

### 5.2.4 Implementasi Antarmuka

# **PENGUJIAN**

**6.1 Pengujian Unit**

**6.2 Pengujian Integrasi**

**6.3 Pengujian Validasi**

**6.4 Pengujian *Compatibility***

# **PENUTUP**

**DAFTAR PUSTAKA**

Alshamrani, A., & Bahattab, A. (2015). A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model. *IJCSI International Journal of Computer Science Issues*, *12*(1), 106–111.

Bhakti, D. D., Putri, S. M., Nasrulloh, I., Tetep, & Nurkamilah, S. (2019). The development of ppdb (admission of new students) application to develop the quality of new students’ recapitulation administration in vocational high school bumi cikajang. *Journal of Physics: Conference Series*, *1280*(3). https://doi.org/10.1088/1742-6596/1280/3/032041

Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *The Unified Modeling Language User Guide* (2nd ed.). Addison-Wesley.

Dewaweb. (2018). *Midtrans Payment Gateway: Panduan Dasar*. https://www.dewaweb.com/blog/midtrans-payment-gateway/

Fandhilah, Rindina, Ferdiansyah, & Ishaq, 2019. (2019). *Implementasi Metode Waterfall Pada Pengembangan Sistem Informasi Penerimaan Siswa Baru Berbasis Web Pada SMK Negeri 2 Adiwerna*. *5*(1), 104–113.

Kebudayaan, M. P. dan. (2018). Permendikbud Nomor 51 Tahun 2018 Tentang Penerimaan Peserta Didik Baru Pada Taman Kanak-Kanak, Sekolah Dasar, Sekolah Menengah Pertama, Sekolah Menengah Atas, Dan Sekolah Menengah Kejuruan. *Permendikbud*, 1–26.

Mukhlasin, H. (2013). Kupas Tuntas CSS. *Kupas Tuntas CSS*, *1999*(December 1996), 1–61.

Mutiarin, D., & Wijaya, J. H. (2017). *E Valuasi P Enerapan S Iap -P Pdb O Nline D Alam M Eningkatkan M Utu O Nline Siap-Ppdb Evaluation Toward Educational Services Quality*.

Pardigmn, V. (2013). *Visual Paradigm Quick Start Users Guide*. 1–15.

Satzinger, J., Jackson, R., & Burd, S. (2010). *Sistem Analysis & Design In a Changing World* (5th ed.). Cengage Learning. https://doi.org/10.1109/TMAG.2015.2421281

Satzinger, J., Jackson, R., & Burd, S. (2012). *Introduction to Sistems Analysis and Design* (6th ed.). Cengage Learning.

Silalahi, M. (2018). Perbandingan Performansi Database Mongodb Dan Mysql Dalam Aplikasi File Multimedia Berbasis Web. *Computer Based Information Sistem Journal*, *6*(1), 63. https://doi.org/10.33884/cbis.v6i1.574

Singh, Y. (2010). Software testing. *Software Testing*, *9781107012*, 1–626. https://doi.org/10.1017/CBO9781139196185

Sofwan, A. (2011). Belajar Mysql dengan Phpmyadmin Pendahuluan. *Modul Kuliah Graphical User Interface I (GUI) Di Perguruan Tinggi Raharja*, 1–29.

Sommerville. (2011). *SOFTWARE ENGINEERING* (9th ed.). https://doi.org/10.1136/bmj.1.5802.756-b

Sommerville, I. (2003). *Software Engineering* (H. Wibi Hardani (Ed.); 6th ed.). Erlangga.

Sutabri, T. (2012a). *Analisis Sistem Informasi* (C. Putri (Ed.)). C.V ANDI.

Sutabri, T. (2012b). *Konsep Sistem Informasi* (I. Nastiti (Ed.)). C.V ANDI.

Tutorialspoint.com. (2005). Bootstrap tutorial. *Mathematica Journal*, *9*(4), 768–775. https://doi.org/10.1198/tech.2005.s292

Yuliano, T. (2007). Pengenalan PHP. *Ilmiu Komputer*, 1–9.

Zenziva. (2017). *MANFAAT DAN KEUNTUNGAN MEMILIKI SMS GATEWAY*. https://www.zenziva.id/manfaat-dan-keuntungan-memiliki-sms-gateway/