**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 IN*FORM*ATIKA

JURUSAN TEKNIK IN*FORM*ATIKA

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 Sengkang 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 memelihaara 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 lendan 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 permasalah 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 resiko 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 yan gingin diperoleh dari penelitian ini sebagai berikutyang 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. Menguhemat 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 penilitian 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 MongoDB.

**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 kedalam 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 penerimana 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 MA 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 deskrip 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 per*form*asi 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 dimana 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 didefenisikan terlehin dahulu sebelum di design, meminimalkan planning overhead, dll. Dalam kegunaannya waterfall coock untuk sistem yang mementingkan kualitas daripada biaya, kebutuhan sistem sangat jelas dan tetap dan porting produk yang ada ke *platform* baru. Spiral model cock digunakan pada proyek-proyek berisiko tinggi, ketika perubahan yang signifikan diharapkan dan kebutuhan pengguna belum diketahui dengan pasti. Sedangkan untuk incremental/iterative model digunakaan untuk sistem dengan resiko menengah, memungkinkan pengguna untuk menyesuaikan diri denga sistem jika pada ptoyek menggunakan teknologi baru dan juga ketika berisiko tinggi untuk mengembangkan seluruh sistem sekaligus.

Kajian pustaka kelima dilakukan oleh (Bhakti, Putri, Nasrulloh, Tetep, & Nurkamilah, 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 pelamardan komite PPDB dengan jumlah sampel 20 calon pelamar dan 5 komite PPDB, hasil pengujian menyatakan sistem dalam kategori sangat layak denga 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 adalah data yang telah diproses atau data yang memiliki arti (Sutabri, 2012b). Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata(Sutabri, 2012b). Sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu(Sutabri, 2012a). Sebuah sistem dapat berupa sistem terbuka dan tertutup, sistem informasi biasanya adalah sistem terbuka, yang berarti bahwa sistem tersebut dapat menerima berbagai masukan dari lingkungan sekitarnya(Sutabri, 2012a).

Sistem Informasi adalah Suatu kumpulan dari komponen yang saling berhubungan yang mengumpulkan, memproses, menyimpan, dan menyediakan output dari suatu informasi yang dibutuhkan untuk menyelesaikan tugas bisnis(Satzinger, Jackson, & Burd, 2012). Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan laporan-laporan yang diperlukan oleh puhak luar tertentu (Sutabri, 2012a).

## **2.3 PPDB**

Penerimaan Santri baru Baru, yang selanjutnya disingkat PPDB, adalah penerimaan santri baru baru pada TK dan Sekolah. PPDB dilakukan berdasarkan nondiskriminatif, objektif, transparan, akuntabel dan berkeadilan (Kebudayaan, 2018). Berdasarkan Permendikbud Nomor 51 tahun 2018 BAB II Pasal 4 tetang 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 disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal spesifikasi sistem sampai pemeliharaan sistem setelah digunakan. Selain itu, menurut Pressman (2012:15) merupakan aplikasi dari suatu pendekatan yang sistematik, disiplin dan dapat diukur pada pengembangan, operasi dan perawatan perangkat lunak. Sehingga, secara umum rekayasa perangkat lunak memakai pendekatan yang sistematis dan terorganisir terhadap pekerjaan mereka karena cara ini seringkali paling efektif untuk menghasilkan perangkat lunak berkualitas tinggi (Sommerville, 2003)

Pada rekayasa perangkat lunak, terdapat beberapa model pengembangan yang dapat digunakan, masing-masing model memiliki alur pengembangan yang digunakan sesuai 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 acuan selama proses pengembangan dimana didalamnya terdapat petunjuk yang mendefinisikan seluruh alur aktivitas yang akan dilakukan selama pengembangan perangkat lunak. *SDLC (Systems Development Life Cycle*) merupakan seluruh proses ruang lingkup sistem yang dimulai pada tahap Pengembangan (*building*), menyebarkan (*deploying*), menggunakan (*using*) dan memperbarui (*updating*) sistem informasi (Satzinger, Jackson, & Burd, 2010). SDLC Model memiliki beberapa model yang akan merepresentasikan pendekatan yang beda dalam tahapan pengembangan perangkat lunak (Sommerville, 2011).

*Waterfall* merupakan salah satu contoh SDLC dengan metode pengembangan yang didasarkan dengan prinsip sekuensial, pada prinsipnya hasil dari setiap fase merupakan satu atau lebih dokumen yang disetujui artinya dalam tiap fase harus terjadi proses validasi dan verifikasi karena fase berikutnya tidak dapat dimulai sebelum fase sebelumnya selesai (Sommerville, 2003). 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 tahap awal sehingga akan menghasilkan perancangan yang baik tanpa kecacatan desain (Sommerville, 2011). Metode *Waterfall* memiliki tahapan utama dari Waterfall model yang mencerminkan aktifitas pengembangan dasar. Pada Gambar 2.1 merupakan gambar metode pengembangan *Waterfall*, terdapat 5 (lima) tahapan pada metode W*aterfall*, 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

System and software design

Implementation and unit testing

Integration and system 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 diskusi dengan user sistem, persyaratan ini kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem. Spesifikasi perangkat lunak ditujukan untuk menetapkan layanan apa yang dituntut dari sistem dan batasan pada operasi dan pengembangan sistem, hasil dari tahap ini yaitu dokument persyaratan yang merupakan spesifikasi sistem. Pengembangan sistem memerlukan spesifikasi sistem yang lebih rinci, ada 4 fase utama padaproses rekyasa kebutuhan yaitu:

1. *Studi Kelayakan*, dibuat perkiraan mengenai apakah user yang didefinisikan puas menggunakan perangkat lunak dan teknologi perangkat keras yang dipakai pada saat ini.
2. *Eisitasi dan Analisis rekayasa kebutuhan*, ini merupakan proses penurunan persyaratan sistem melaui observasi sistem yang ada, diskusi dengan use yang akan memakai dan yang akan mengadakan, analisis pekerjaan, dll. Hasil fase ini akan membantu analis memahami sistem yang akan dispesifikasi.
3. *Spesifikasi Kebutuhan*, kegiatan menerjemahkan informasi yang dikumpulkan pada kegiatan analisis kebutuhan menjadi dokumen yang mendefinisikan serangkaian kebutuhan.
4. *Validasi Kebutuhan*, tahap ini memeriksa apakah persyaratan dapat direalisasikan, konsisten dan lengkap. Selain itu, kesalahan pada dokument akan ditemukan pada tahap ini kemudian dimodifikasi untuk menyelesaikan masalahnya.
   * + 1. Perancangan sistem dan perangkat lunak

Proses perancangan sistem membagi persyaratan dalam sistem perangkat keras atau perangkat lunak. Kegiatan ini menetukan 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 danbatas 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 opersi subsistem.
4. Perancangan Komponen, layanandialokasikan 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 system di perlukan dalam membangun ataupun menganalisa sebuah system apakah sudah berjalan sebagai mana mestinya atau tidak. Hal ini perlu dilakukan mengingat white box testing merupakan pengecekan system secara lebih detail dan komplek ketimbang system black box testing, di white box testing system tidak hanya di cek dari fungsional luarnya saja "interface" melainkan juga dari berbagai aspek system, termasuk diagram alur system dan berbagai komponent lainnya yang ada dalam system (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, setalah 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 komleksitas 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 didunia 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 dimana komponen-komponen fundamentalpada design mempresentasikan objek dengan status pribadinya sendiri dan operasi ketimbang fungsi. *Object Oriented Programming* merupakan proses untuk merubah rancangan yang telah dibuat dalam bentuk bahasa pemrograman. Pemrograman berorientasi objek menggunakan beberapa konsep dalam proses pemrogramannya, yaitu encapsulation, in*form*ation 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 system* (Booch, Rumbaugh, & Jacobson, 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 actor dengan use case |
| 4. |  | Merupakan relasi tambahan dalam use case, dimana use case yang ditambahkan dapat berdiri sendiri walaupun tampa use case tambahan tersebut. |
| 5. |  | Merupakan hubungan generalisasi antar dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya |
| 6. |  | Merupakan relasi tambahan dalam use case dimana 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 diagrampada UML adalah untuk menggambarkan interaksi antara aktor dan objek pada sebuah sistem dan juga interaksi antara objek itu sendiri (Sommerville, 2011). Sequencediagram, 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 system 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 dimana objek anak (*descendent*) berbagi perilaku dan struktur dari objek yang ada di atasnya |
| 4 |  | Hubungan dimana 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), NoSQL dan Codeigniter.

### 2.5.1 Bahasa Pemrograman PHP

**PHP (akronim dari PHP: Hyper*text*  Preprocessor) adalah bahasa pemrograman yang berfungsi untuk membuat website dinamis maupun aplikasi web. Berbeda dengan HTML yang hanya bisa menampilkan konten statis, PHP bisa berinteraksi dengan database, *file* dan folder, sehingga membuat PHP bisa menampilkan konten yang dinamis dari sebuah website (Yuliano, 2007). Blog, Toko Online, CMS, Forum, dan Website Social Networking adalah contoh aplikasi web yang bisa dibuat oleh PHP. PHP adalah bahasa scripting, bukan bahasa tag-based seperti HTML. PHP termasuk bahasa yang *cross-platform*, ini artinya PHP bisa berjalan pada sistem operasi yang berbeda-beda (Windows, Linux, ataupun Mac). Program PHP ditulis dalam *file* plain *text*  (teks biasa) dan mempunyai akhiran “.php” (Yuliano, 2007).**

### ****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 di customisasi (Tutorialspoint.com, 2005).

### 2.5.3 Cascading Style Sheets (CSS)

CSS singkatan dari Cascading Style Sheets, yaitu sebuah bahasa berbasis *text*  yang digunakan untuk mem*form*at 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 NoSQL****

Istilah NoSQL diciptakan oleh Carlo Strozzi pada tahun 1998 dan mengacu pada database non-relasional, pada tahun 2009 Eric Evans memperkenalkan kembali istilah NoSQL. Baru-baru ini, istilah ini memiliki makna lain, yaitu "Not Only SQL", istilah yang lebih baik dari sebelumnya yang lebih dikenal dengan “anti-relasional” (No & Junaidi, 2016). Perbedaan antara SQL dan NoSQL adalah SQL memiliki skema yang kaku sementara database NoSQL menawarkan desain yang fleksibel yang dapat diubah tanpa downtime atau gangguan layanan (No & Junaidi, 2016).

NoSQL menjadi solusi dalam penanganan data dalam jumlah besar yang berkembang pesat saat ini. Data ini biasanya non-terstruktur, kompleks dan tidak cocok digunakan dalam model relasional. Contoh data yang bisa kita yang bisa kita rasakan adalah data yang berasal dari smartphone yang mencatat lokasi broadcast setiap saat, video dan kamera bahkan halaman halaman website yang berisi banyak informasi serta dokumen (No & Junaidi, 2016).

#### 2.5.4.1 MongoDB

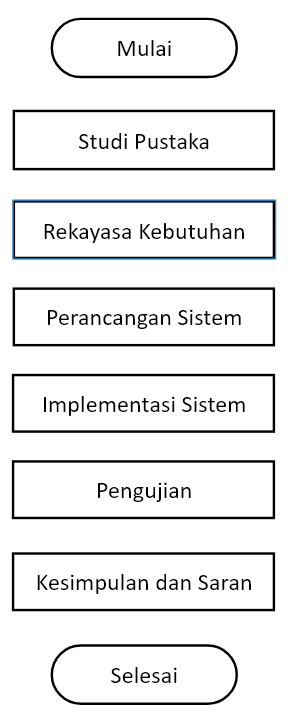
MongoDB adalah sebuah database yang berorientasi dokumen lintas *platform*. MongoDB dikelompokkan kedalam NoSQL database, memberikan kemampuan untuk query pada bidang apapun dalam dokumen. MongoDB is a *cross-platform* document-oriented database system. Classified as a "NoSQL" database, MongoDB bukan database relasional database, MongoDB support untuk dokumen JSON yang membuat integrasi data dalam beberapa jenis aplikasi lebih mudah dan lebih cepat. MongoDB di release dibawah kombinasi lisensi GNU Affero General Public License dan, MongoDB free untuk digunakan dan opensource, ditulis dalam bahasa C++ dan disupport 10gen (No & Junaidi, 2016).

### 2.5.5 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 dimana – mana.

# **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 analisi 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 kedalam 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 kedalam 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 kedalam 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 di kembangkan. Pada pengembangan sistem ini, sistem di uji 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 di gunakan oleh pengguna atau user dengan menggunakan *black-box* testing. Sistem di uji tampilan untuk mengetahui apakah terdapat tampilan yang kurang baik, dan apakah inputan dan keluaran sesuai dengan test case yang di buat.

**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.

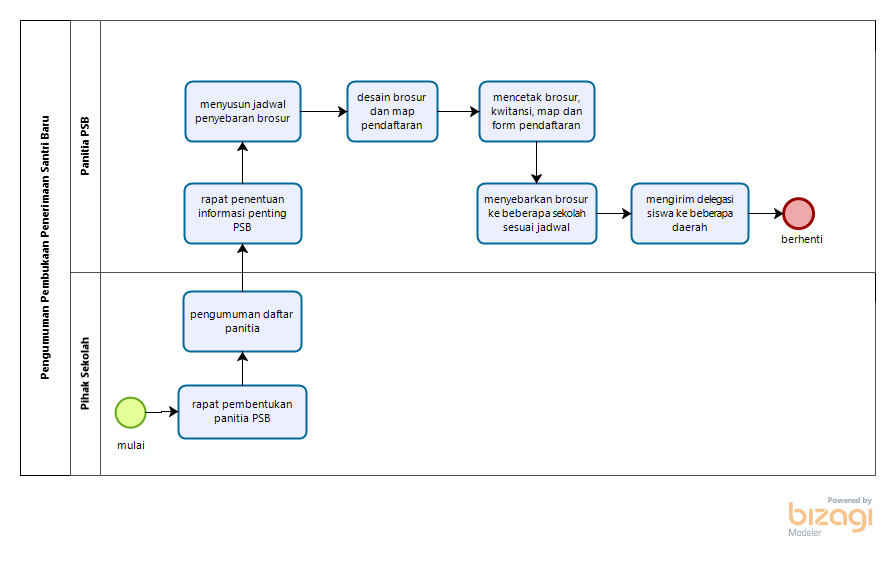
# **ANALISIS KEBUTUHAN**

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**

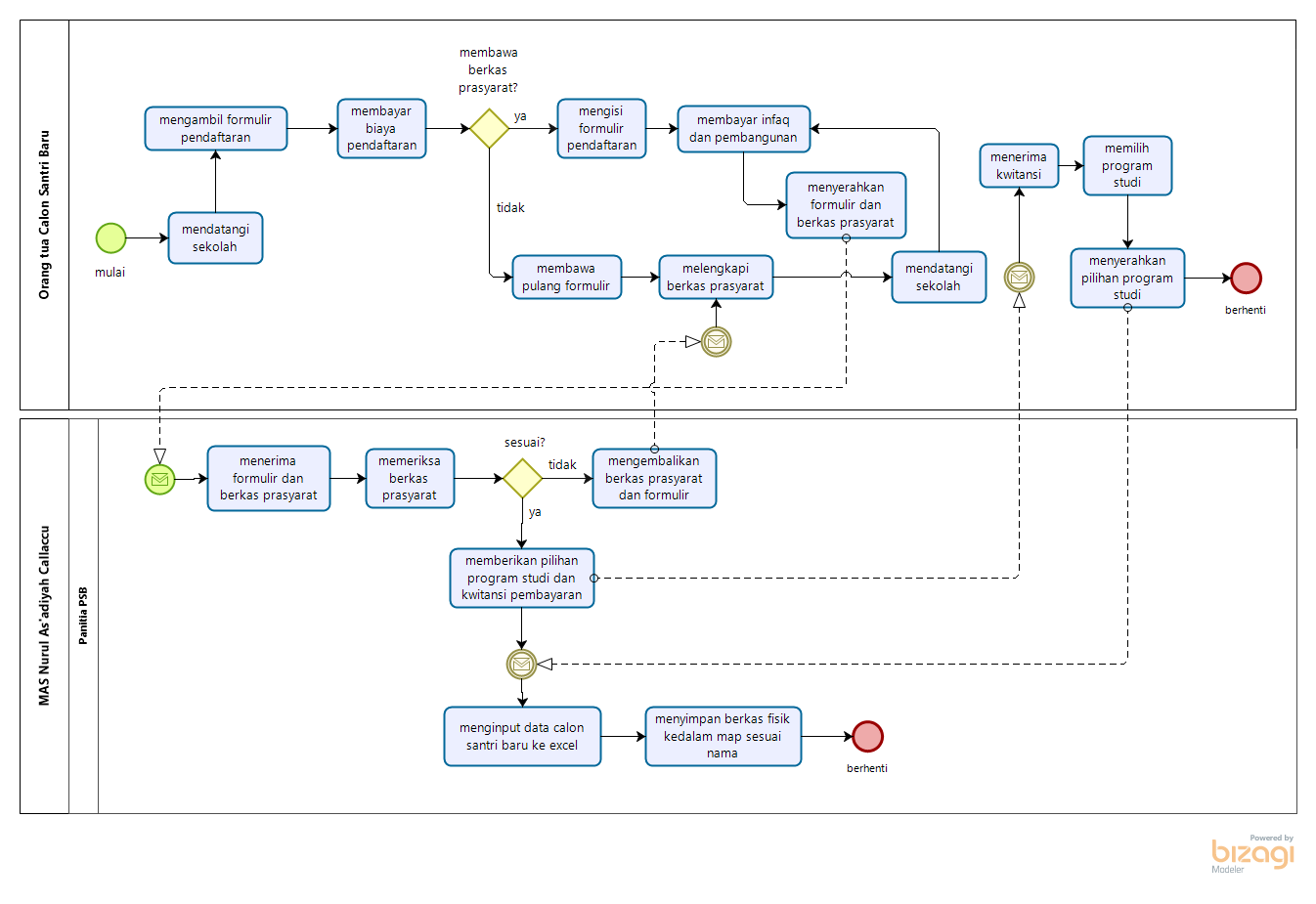
Proses elisitasi kebutuhan dilakukan untuk memperoleh daftar kebutuhan sistem serta daftar aktor yang akan berinteraksi dengan sistem, elisitasi dilakukan dengan metode wawancara terhadap salah satu pengurus PSB yaitu Bapak Muhammad Akis, wawancara dilakukan dengan topik pembahasan terkait proses bisnis 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, pengumuman dan pendaftaran ulang. 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.



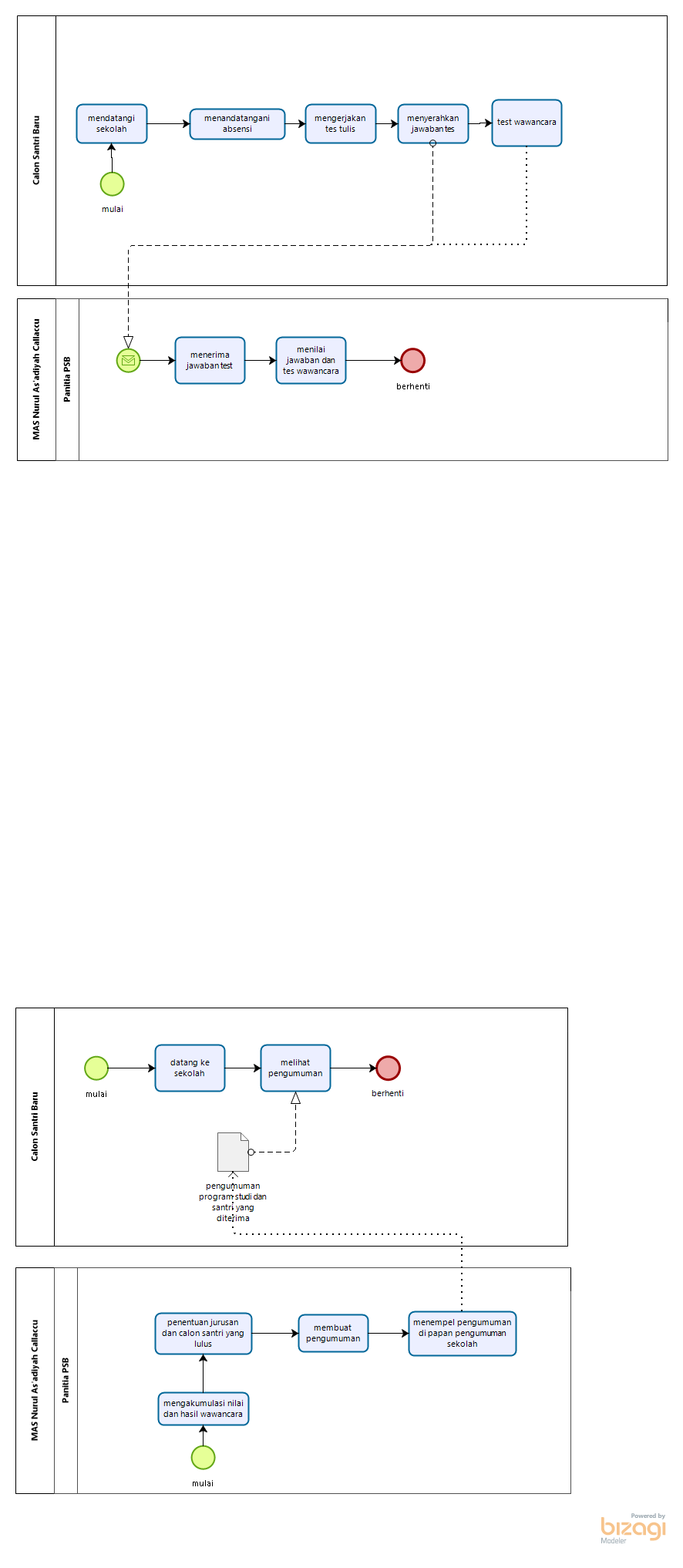
**Gambar 4.1 Pengumuman Pembukaan PSB**

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, proses tersebut menyulitkan orang tua calon santri baru khususnya yang berada diluar 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, namun ada beberapa orang tua calon santri baru yang telah membawa berkas prasyarat sehingga cukup mengisi formulir dan menyerahkan berkas prasyarat. Panitia PSB akan mengecek kelengkapan dan kesesuain berkas yang diberikan, jika sesuai maka panitian akan memberikan pilihan program studi dan kwitansi pembayaran, setelah itu orang tua calon santri akan memilih program studi yang diinginkan dan panitia akan menginputkan keseluruhan data yang diberikan kedalam 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.



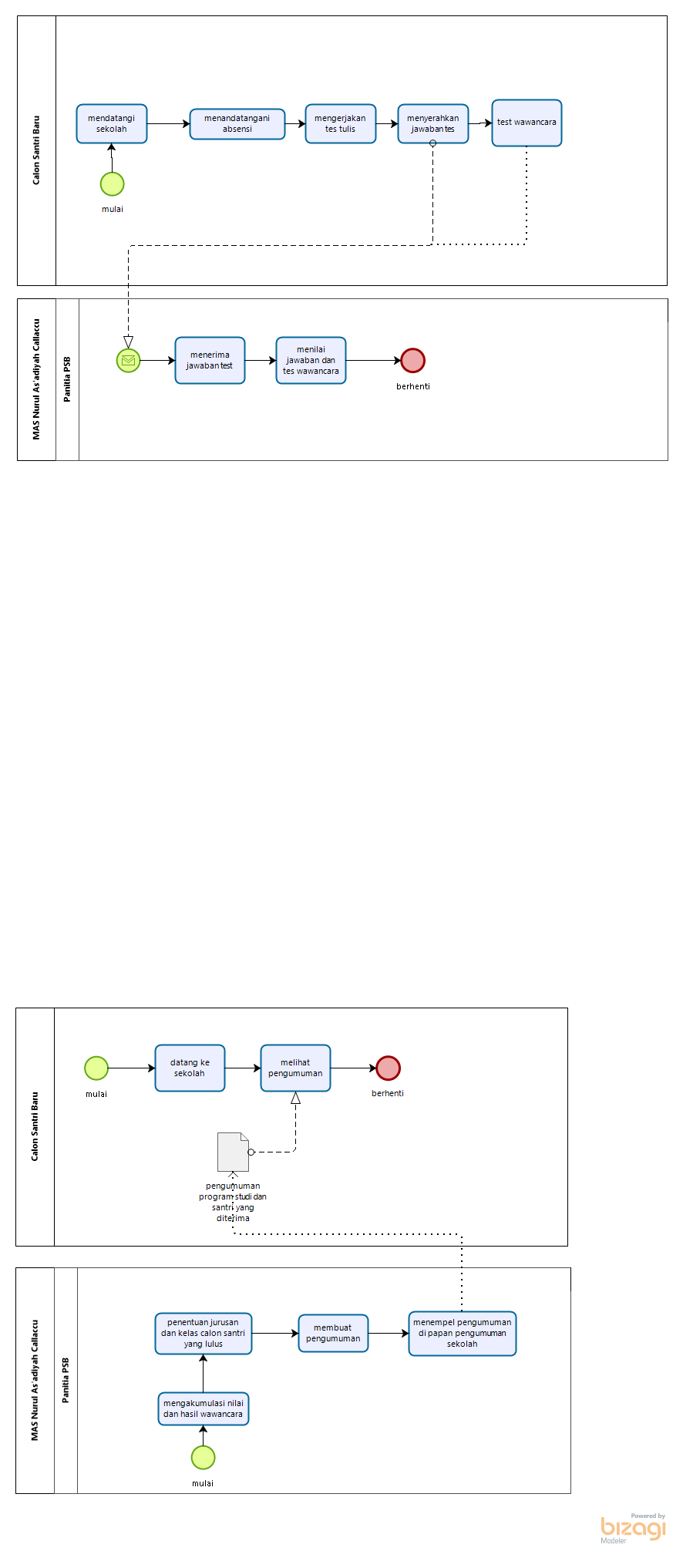
**Gambar 4.2 Alur Pendaftaran**

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. Dihari 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 medrasah 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 siswa aktif, 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 dan mengunduh kartu peserta pada menu my profil, calon santri juga dapat mengerjakan test seleksi online, *upload* bukti pembayaran dan berkas prasyarat yang telah ditentukan. Cetak kartu peserta dapat dilakukan jika semua berkas dan pembayaran telah diverifikasi oleh panitia PSB, selain itu calon santri dapat mengakses pengumuman pada jadwal yang telah ditentukkan dengan memasukkan Nomor Induk Nasional (NISN)/Nomor pendaftaran pada web PSB.

Setelah peserta PSB selesai melakukan pendaftaran maka panitia PSB akan mengecek berkas dan bukti pembayaran dan akan memverifikasi jika semuanya telah sesuai. Panitia PSB merupakan panitia yang mengelolah data peseta PSB dan mengatur seluruh kegiatan yang berhubungan dengan proses penerimaan santri baru, panitia PSB memiliki fugnsi untuk dapat memverifikasi berkas peserta PSB, mengelolah soal tes seleksi online, mengelolah nilai tes seleksi (akademik dan wawancara), mengelolah pengumuman calon santri yang diterima, mengelolah data arsip, mengelolah pembayaran SPP siswa aktif dan mencetak nama-nama santri baru yang diterima ataupun yang mendaftar untuk menjadi laporan ke kepala sekolah.

Setelah pengumuman keluar dan santri dinyatakan lulus, maka santri akan berstatus siswa aktif dan dapat melakukan pembayaran SPP atau uang madrasah tiap semesternya, uang madrasah dapat diangsur dengan batas angsuran sebanyak empat kali. Siswa aktif memiliki fungsi melihat status pembayaran, mengubah data diri, upload bukti pembayaran SPP, melihat nominal tagihan pembayaran dan melihat data diri.

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

## **4.3 Identifikasi Aktor**

Identifikasi aktor diperoleh dari hasil wawancara dengan panitia PSB, dimana 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 4 aktor diantaranya admin, panitia PSB, peserta PSB dan pengunjung. 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 PSB | Panitia PSB yang bertugas untuk mengelolah administrasi peserta PSB. |
| 3 | Peserta PSB | Pengguna yang telah terdaftar dalam sistem dan merupakan calon santri aktif. |
| 4 | Pengunjung | Pengguna yang belum melalui autentikasi sistem. |
| 5 | Santri Aktif | Peserta PSB yang dinyatakan lulus pada proses penerimaan santri baru. |

## **4.4 Spesifikasi Kebutuhan**

Spesifikasi kebutuhan dalam sistem PSB akan menggunakan penomoran PSB\_F\_XX\_YY dengan penjelasann 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.

**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 santri aktif, admin, panitia PSB atau peserta PSB. | *Login* | Pengunjung |
|  | PSB\_F\_01\_01 | Sistem menyediakan *form* untuk masuk kedalam sistem yang berisi data NIS/NISN/Nomor Induk Pegawai (NIP) dan *password*. |
|  | PSB\_F\_01\_02 | Sistem menyediakan fasilitas ganti *password*. |
|  | PSB\_F\_02\_00 | Sistem menyediakan halaman untuk dapat membuat akun bagi pengunjung sebagai pesera PSB. | Buat Akun | Pengunjung |
|  | PSB\_F\_02\_01 | Sistem menyediakan *form* untuk membuat akun yang berisi data nama lengkap, NISN, *email* dan *passwor*. |
|  | PSB\_F\_02\_02 | NISN yang dimasukkan harus sejumlah 10 digit dan belum terdaftar pada sistem. |
|  | PSB\_F\_03\_00 | Sistem menyediakan halaman untuk dapat mengganti *password* jika lupa. | Ganti *password* | Pengunjung |
|  | PSB\_F\_03\_01 | Sistem menyediakan *form* ganti *password* yang berisi data NISN, *password* baru dan ulang *password*. |
|  | PSB\_F\_03\_02 | NISN yang dimasukkan harus sejumlah 10 digit dan telah terdaftar pada sistem. |
|  | PSB\_F\_04\_00 | Sistem menyediakan fasilitas *logout* untuk menghilangkan hak bagi pengguna sistem. | *Logout* | Admin, santri aktif, panitia PSB dan peserta PSB |
|  | PSB\_F\_05\_00 | Sistem menyediakan fasilitas untuk dapat unduh panduan pendaftaran PSB. | Unduh panduan | Pengunjung |
|  | PSB\_F\_06\_00 | Sistem menyediakan halaman untuk dapat melihat hasil tes. | Lihat hasil tes | Pengunjung |
|  | PSB\_F\_06\_01 | Hasil dapat dilihat denan menginputkan NISN peserta PSB. |
|  | PSB\_F\_06\_02 | Sistem akan menampilkan keterangan lulus atau tidak lulus sekaligus program studi yang terpilih. |
|  | PSB\_F\_06\_03 | NISN yang diinputkan merupakan NISN yang terdaftar dalam sistem dan berjumlah 10 digit. |
|  | PSB\_F\_07\_00 | Sistem menyediakan halaman untuk dapat mengirim keluhan dan pertanyaan kepada admin. | Hubungi admin | Pengunjung |
|  | PSB\_F\_07\_01 | Sistem menyediakan *form* untuk dapat menghubungi admin yang berisi data nama, *e-mail*, tentang dan isi pesan. |
|  | PSB\_F\_08\_00 | Sistem menyediakan halaman untuk dapat melihat informasi sekolah yang tercantum pada web PSB. | Lihat informasi sekolah | Admin |
|  | PSB\_F\_08\_01 | Sistem menampilkan informasi nama sekolah, akreditasi, alamat, kodepos, nomor telepon, email, status sekolah, struktur organisasi sekolah, visi, misi, ekstrakurikuler sekolah dan sejarah singkat sekolah. |  |
|  | PSB\_F\_08\_01 | Sistem menyediakan fasilitas untuk dapat mengubah informasi sekolah. |
|  | PSB\_F\_09\_00 | Sistem menyediakan halam untuk dapat mengubah profil sekolah. | Ubah informasi sekolah | Admin |
|  | PSB\_F\_09\_01 | Semua informasi dapat diubah dan tidak boleh kosong. |
|  | PSB\_F\_10\_00 | Sistem menyediakan halaman untuk dapat melihat informasi PSB yang tercantum pada web PSB. | Lihat informasi PSB | Admin |
|  | PSB\_F\_10\_01 | Sistem menampilkan Informasi PSB dalam sub menu meliputi jadwal PSB, biaya, syarat pendaftaran yang berada pada sub menu. |
|  | PSB\_F\_10\_02 | Sistem menyediakan fasilitas untuk dapat megubah informasi PSB. |
|  | PSB\_F\_11\_00 | Sistem menyediakan halaman untuk dapat mengubah informasi PSB. | Ubah informasi PSB | Admin |
|  | PSB\_F\_11\_01 | Semua informasi PSB dapat diubah dan tidak boleh kosong. |
|  | PSB\_F\_12\_00 | Sistem menyediakan halaman untuk dapat melihat daftar pesan dari pengunjung. | Lihat daftar pesan | Admin |
|  | PSB\_F\_12\_01 | Pesan ditampilkan dalam bentuk tabel yang berisi nama, *email*, tentang dan isi pesan. |
|  | PSB\_F\_12\_02 | Sistem menyediakan fungsi pencarian pesan. |
|  | PSB\_F\_13\_00 | Sistem menyediakan fasilitas untuk dapat menghapus pesan dari pengunjung. | Hapus pesan | Admin |
|  | PSB\_F\_14\_00 | Sistem menyediakan fasilitas untuk membalas pesan. | Balas pesan | Admin |
|  | PSB\_F\_14\_01 | Sistem menyediakan *form* untuk membalas pesan yang berisi data *email*, pesan dari pengunjung dan isi balasan pesan. |
|  | PSB\_F\_15\_00 | Sistem menyediakan halaman untuk dapat melihat daftar pengguna sistem. | Lihat daftar pengguna | Admin |
|  | PSB\_F\_15\_01 | Daftar pengguna yang ditampilkan adalah santri aktif, panitia dan peserta PSB. |
|  | PSB\_F\_15\_02 | Sistem menyediakan fasilitas untuk dapat menambah pengguna. |
|  | PSB\_F\_16\_00 | Sistem menyediakan halaman untuk dapat menambah pengguna. | Tambah pengguna | Admin |
|  | PSB\_F\_16\_01 | Sistem menyediakan *form* untuk dapat menambah pengguna yang berisi NIP, *email* dan *password* bagi panitia dan NISN, NIS, email dan password untuk santri aktif. |
|  | PSB\_F\_16\_02 | Pengguna yang dapat ditambahkan hanya panitia PSB dan santri aktif. |
|  | PSB\_F\_16\_03 | NIP yang didaftar harus berjumlah 18 digit. |
|  | PSB\_F\_16\_04 | NISN yang didaftarkan harus berjumlah 10 digit. |
|  | PSB\_F\_17\_00 | Sistem menyediakan fasilitas untuk dapat menghapus pengguna sistem. | Hapus pengguna | Admin |
|  | PSB\_F\_18\_00 | Sistem menyediakan halaman untuk dapat mengisi data pendaftaran. | Isi *form* pendaftaran | Peserta PSB |
|  | PSB\_F\_18\_01 | Sistem menyediakan *form* pendaftaran yang berisikan foto, data diri, orang tua, wali, prestasi dan organisasi. |
|  | PSB\_F\_18\_02 | Semua data harus diisi. |
|  | PSB\_F\_19\_00 | Sistem menyediakan halaman untuk dapat melihat data profil yang telah diisi saat mengisi *form* pendaftaran. | Lihat profil | Peserta PSB |
|  | PSB\_F\_19\_01 | Sistem menyediakan informasi status dari data pendaftaran. |
|  | PSB\_F\_19\_02 | Sistem menyediakan fasilitas ntuk dapat mengubah profil diri. |
|  | PSB\_F\_20\_00 | Sistem menyediakan halaman untuk dapat mengubah data profil diri. | Ubah profil | Peserta PSB dan santri aktif |
|  | PSB\_F\_20\_01 | NISN atau Nomor Induk Sekolah (NIS) tidak dapat diubah. |
|  | PSB\_F\_21\_00 | Sistem menyediakan fasilitas untuk mencetak kartu peserta. | Cetak kartu peserta | Peserta PSB |
|  | PSB\_F\_21\_01 | Cetak kartu peserta dapat dilakukan jika semua berkas dan data diri telah diverifikasi oleh panitia PSB. |
|  | PSB\_F\_22\_00 | Sistem menyediakan halaman untuk dapat mengerjakan tes seleksi akademik. | Lihat tes seleksi | Peserta PSB |
|  | PSB\_F\_22\_01 | Sistem menyediakan fasilitas untuk dapat mulai mengerjakan soal tes. |
|  | PSB\_F\_23\_00 | Sistem menyediakan halaman untuk mengerjakan soal tes seleksi. | Mulai tes | Peserta PSB |
|  | PSB\_F\_23\_01 | Soal tes yang diberikan merupakan tipe soal essay. |
|  | PSB\_F\_23\_02 | Sistem menyediakan fasilitas untuk mengetikkan jawaban atau meng*upload* jawaban dalam bentuk *file* pdf. |
|  | PSB\_F\_23\_03 | Pengerjaan soal dapat dilakukan pada jadwal yang telah ditentukan dan semua berkas telah diverifikasi panitia PSB. |
|  | PSB\_F\_24\_00 | Sistem menyediakan halaman untuk dapat memilih 2 program studi yang diinginkan. | Pilih program studi | Peserta PSB |
|  | PSB\_F\_24\_01 | Pemilihan program studi dapat dilakukan apabila telah mengerjakan soal tes akademik. |
|  | PSB\_F\_25\_00 | Sistem menyediakan halaman untuk dapat melihat daftar prasyarat yang harus di persiapkan. | Lihat daftar prasyarat | Peserta PSB |
|  | PSB\_F\_25\_01 | Sistem menyediakan *form* untuk dapat unggah berkas prasyarat. |
|  | PSB\_F\_25\_02 | Sistem menyediakan fasilitas unduh untuk semua berkas yang telah diunggah. |
|  | PSB\_F\_25\_03 | Sistem menyediakan fasilitas simpan pada berkas yang diunggah. |
|  | PSB\_F\_26\_00 | Sistem menyediakan fasilitas untuk dapat unggah berkas prasyarat. | Unggah berkas prasyarat | Peserta PSB |
|  | PSB\_F\_26\_01 | Sistem dapat menampilkan status dari berkas yang telah di unggah. |
|  | PSB\_F\_27\_00 | Sistem menyediakan fasilitas untuk dapat mengubah berkas yang telah diunggah. | Ubah berkas prasyarat |  |
|  | PSB\_F\_28\_00 | Sistem menyediakan halaman untuk dapat melihat daftar calon santri yang mendaftar. | Lihat daftar peserta PSB | Panitia PSB |
|  | PSB\_F\_28\_01 | Data calon santri ditampilkan dalam bentuk tabel yang berisi nomor, tanggal pendaftaran, nama NISN, asal sekolah, tanggal lahir dan aksi. |
|  | PSB\_F\_28\_02 | Sistem menyediakan fungsi pencarian. |
|  | PSB\_F\_28\_03 | Sistem dapat menampilkan jumlah data sesuai yang diinginkan. |
|  | PSB\_F\_28\_04 | Sistem menyediakan fasilitas lihat detail pada setiap data peserta PSB. |
|  | PSB\_F\_29\_00 | Sistem menyediakan fasilitas untuk mencetak daftar peserta PSB. | Cetak daftar peserta PSB | Panitia PSB |
|  | PSB\_F\_30\_00 | Sistem menyediakan halaman untuk dapat melihat detail tiap data peserta PSB. | Lihat detail data peserta PSB | Panitia PSB |
|  | PSB\_F\_30\_01 | Sistem menampilkan data dalam bentuk menu tabel. |
|  | PSB\_F\_31\_00 | Sistem menyediakan fasilitas untuk dapat melakukan verifikasi. | Verifikasi pendaftaran | Panitia PSB |
|  | PSB\_F\_31\_01 | Verifikasi dapat dilakukan pada data pribadi, berkas prasyarat dan bukti pembayaran yang telah di unggah peserta. |
|  | PSB\_F\_32\_00 | Sistem menyediakan halaman untuk dapat melihat daftar data santri yang diarsipkan. | Lihat daftar arsip data santri | Panitia PSB |
|  | PSB\_F\_32\_01 | Sistem menyediakan fungsi pencarian data arsip. |
|  | PSB\_F\_32\_02 | Data arsip ditampilkan dalam bentuk tabel yang berisi nomor, tanggal pendaftaran, nama NISN, asal sekolah, tanggal lahir dan aksi. |
|  | PSB\_F\_32\_03 | Sistem menyediakan fasilitas melihat detail arsip data santri. |
|  | PSB\_F\_33\_00 | Sistem menyediakan halaman untuk melihat detail data santri yang diarsipkan. | Lihat detail arsip data santri | Panitia PSB |
|  | PSB\_F\_33\_01 | Data arsip ditampilkan dalam bentuk tabel. |
|  | PSB\_F\_34\_00 | Sistem menyediakan halaman untuk dapat melihat daftar soal tes akademik. | Lihat daftar soal tes | Panitia PSB |
|  | PSB\_F\_34\_01 | Daftar soal akan ditampilkan dalam bentuk tabel yang berisi nomor soal, soal, bidang dan aksi. |
|  | PSB\_F\_34\_02 | Sistem menyediakan fasilitas tambah soal tes. |
|  | PSB\_F\_35\_00 | Sistem menyediakan *form* untuk menambah soal yang berisi bidang soal dan pertanyaan. | Tambah soal tes | Panitia PSB |
|  | PSB\_F\_36\_00 | Sistem menyediakan fasilitas untuk mengubah soal tes. | Ubah soal tes | Panitia PSB |
|  | PSB\_F\_36\_01 | Bidang soal dan pertanyaan dapat diubah. |
|  | PSB\_F\_37\_00 | Sistem menyediakan fasilitas untuk mengirim soal yang telah dibuat untuk dapat dikerjakan. | Kirim soal tes | Panitia PSB |
|  | PSB\_F\_38\_00 | Sistem menyediakan fasilitas untuk menghapus soal tes akademik yang telah dibuat. | Hapus soal tes | Panitia PSB |
|  | PSB\_F\_39\_00 | Sistem menyediakan halaman untuk melihat daftar jawaban peserta PSB. | Lihat daftar jawaban | Panitia PSB |
|  | PSB\_F\_39\_01 | Sistem menampilkan daftar jawaban dalam bentuk tabel yang berisi data waktu selesai, nama, NISN dan jawaban sekilas. |
|  | PSB\_F\_39\_02 | Sistem menyediakan fungsi pencarian data. |
|  | PSB\_F\_39\_03 | Sistem menyediakan fasilitas untuk melihat detail jawaban. |
|  | PSB\_F\_40\_00 | Sistem menyediakan halaman untuk melihat detail jawaban peserta PSB. | Lihat detail jawaban | Panitia PSB |
|  | PSB\_F\_40\_01 | Halaman detail jawaban menampilkan foto, nama, NISN, 2 program studi yang dipilih dan jawaban tes akademik. |
|  | PSB\_F\_41\_00 | Sistem menyediakan *form* untuk dapat menilai jawaban peserta PSB. | Nilai jawaban | Panitia PSB |
|  | PSB\_F\_41\_01 | From terdiri dari data program studi, nilai dari jawaban dan tombol kirim. |
|  | PSB\_F\_41\_02 | Sistem menyediakan fasilitas untuk mengirim penilaian jawaban. |
|  | PSB\_F\_42\_00 | Sistem menyediakan halaman untuk dapat menginput nilai tes akademik. | Input nilai tes akademik | Panitia PSB |
|  | PSB\_F\_42\_01 | Sistem menampilkan daftar nama peserta PSB dalam bentuk tabel. |
|  | PSB\_F\_42\_02 | Sistem menyediakan fungsi pencarian dan menampilkan jumlah data sesuai kebutuhan. |
|  | PSB\_F\_42\_03 | Sistem menyediakan fasilitas simpan untuk menyimpan data nila dan ubah untuk mengubah nilai. |
|  | PSB\_F\_43\_00 | Sistem menyediakan halaman untuk dapat menginput nilai tes wawancara. | Input nilai tes wawancara | Panitia PSB |
|  | PSB\_F\_43\_01 | Sistem menampilkan daftar nama peserta PSB dalam bentuk tabel. |
|  | PSB\_F\_43\_02 | Sistem menyediakan fungsi pencarian dan menampilkan jumlah data sesuai kebutuhan. |
|  | PSB\_F\_43\_03 | Sistem menyediakan fasilitas simpan dan ubah nilai tes wawancara. |
|  | PSB\_F\_44\_00 | Sistem menyediakan halaman untuk dapat melihat hasil tes seleksi peserta PSB. | Lihat akumulasi tes | Panita PSB |
|  | PSB\_F\_44\_01 | Sistem menampilkan daftar peserta dalam bentuk tabel yang berisi nomor, nama, NISN, program studi, nila tes akademik, nilai tes wawancara dan status lulus/tidak lulus. |
|  | PSB\_F\_44\_02 | Sistem menyediakan fungsi menampilkan jumlah data sesuai kebutuhan. |
|  | PSB\_F\_44\_03 | Sistem menyediakan fasilitas untuk publish hasil tes. |
|  | PSB\_F\_45\_00 | Sistem menyediakan fasilitas untuk mencetak hasil akumulasi tes seleksi. | Cetak hasil akumulasi tes | Panitia PSB |
|  | PSB\_F\_46\_00 | Sistem menyediakan halaman untuk dapat melihat daftar santri aktif. | Lihat daftar santri aktif | Panitia PSB |
|  | PSB\_F\_46\_01 | Sistem menyediakan fungsi pencarian dan tampilkan jumlah data sesuai kebutuhan. |
|  | PSB\_F\_46\_02 | Data santri aktif ditampilkan dalam bentuk tabel dengan data yang ditampilkan yaitu nama, NIS, Program studi, kelas dan status SPP . |
|  | PSB\_F\_46\_03 | Sistem menyediakan informasi total santri aktif. |
|  | PSB\_F\_46\_04 | Sistem menyediakan fasilitas untuk melihat detail data santri aktif dan mengarsipkan data. |
|  | PSB\_F\_47\_00 | Sistem menyediakan fasilitas untuk mencetak data santi. | Cetak data santri | Panitia PSB |
|  | PSB\_F\_48\_00 | Sistem menyediakan halaman untuk dapat melihat detail data diri santri aktif. | Lihat detail data santri aktif | Panitia PSB |
|  | PSB\_F\_48\_01 | Detail data yang ditampilkan adalah data pribadi dan SPP. |
|  | PSB\_F\_49\_00 | Sistem menyediakan halam untuk dapat melihat riwayat pembayaran SPP siswa aktif. | Lihat riwayat SPP | Panitia PSB |
|  | PSB\_F\_49\_01 | Data SPP ditampilkan dalam bentuk tabel, dengan data yang ditampilkan yaitu nama pembayaran, nominal bayar, tanggal dan *file*. |
|  | PSB\_F\_49\_02 | Sistem menyediakan fungsi mengubah status SPP. |
|  | PSB\_F\_50\_00 | Sistem menyediakan fasilitas untuk dapat melakukan verifikasi pembayaran SPP siswa aktif. | Verifikasi SPP | Panitia PSB |
|  | PSB\_F\_51\_00 | Sistem menyediakan halaman untuk dapat melihat data diri santri. | Lihat data diri | Santri Aktif |
|  | PSB\_F\_51\_01 | Sistem menyediakan fasilitas untuk dapat mengubah data diri. |
|  | PSB\_F\_52\_00 | Sistem menyediakan halam untuk dapat melihat total tagihan SPP. | Lihat tagihan SPP | Santri Aktif |
|  | PSB\_F\_52\_01 | Sistem menyediakan informasi status pembayaran. |
|  | PSB\_F\_52\_02 | Sistem mampu menampilkan informasi data nama, NIS, normor rekening dan total tagihan SPP. |
|  | PSB\_F\_53\_00 | Sistem menyediakan *form* untuk melakukan konfirmasi pembayaran SPP meliputi data taggal pembayaran dan nominal pembayaran. | Konfirmasi pembayaran SPP | Santri aktif |
|  | PSB\_F\_53\_01 | Sistem menyediakan fasilitas untuk mengunggah bukti pembayaran SPP. |
|  | PSB\_F\_53\_02 | Sistem menyediakan informasi status dari konfirmasi pembayaran SPP. |
|  | PSB\_F\_54\_00 | Sistem menyediakan halaman untuk dapat melihat riwayat pembayaran SPP. | Lihat riwayat pembayaran SPP | Santri Aktif |

### 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**

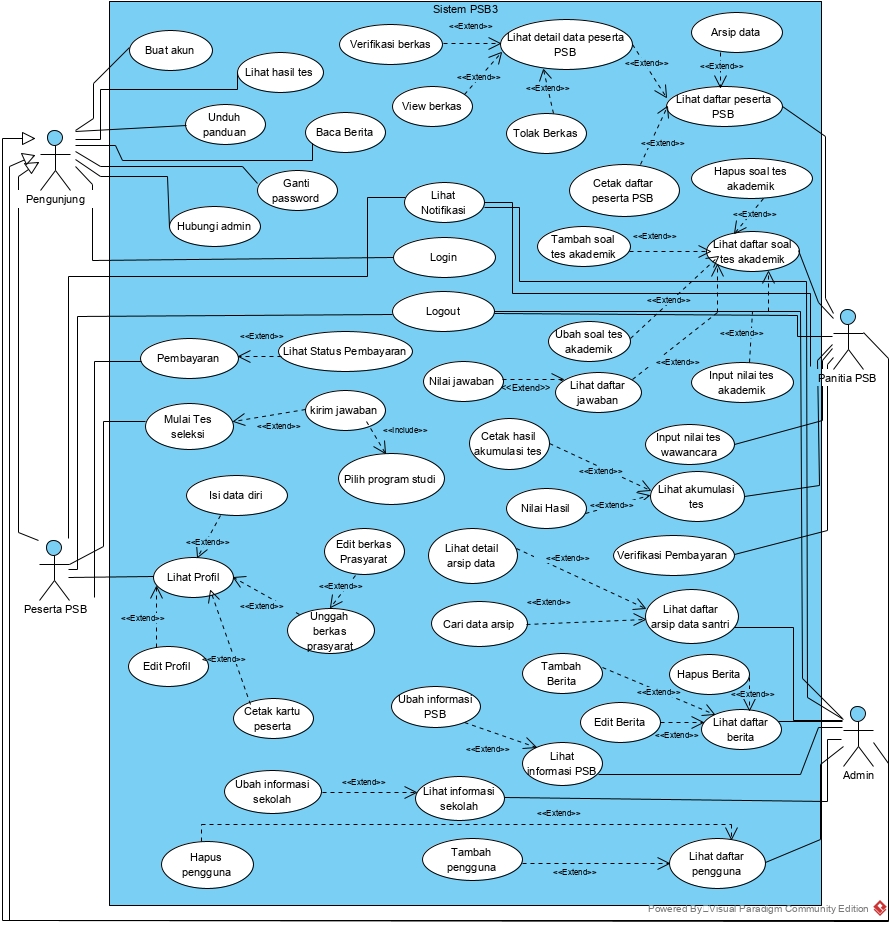
|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **Kode** | **Parameter** | **Deskripsi** |
| 1 | PSB\_NF\_01 | *Usability* | Sistem yang dikembangkan dapat digunakan dengan mudah dioperasikan oleh aktor. |

## **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 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) | |
| ***Actor*** | Pengunjung |
| ***Objective*** | Dapat masuk kedalam sistem |
| ***Preconditions*** | Halaman *login* sudah ditampilkan |
| ***Main Flows*** | 1. Sistem menampilkan halaman *login* dengan data masukan NIP/NIS/NISN dan *password*.  2. Aktor mengisi data dengan lengkap.  3. Aktor menekan tombol masuk.  4. Sistem menampilkan halaman dashboard. |
| ***Alternative Flows*** | 1. Sistem menampilkan peringatan “Harap lengkapi data terlebih dahulu” jika data NIP/NIS/NISN atau *password* kosong.  2. Sistem menampilkan peringatan “Maaf, akun tidak ditemukan” jika NIP/NIS/NISN tidak ditemukan dalam daftar pengguna.  3. Sistem menampilkan peringatan “Maaf, *password* yang dimasukkan salah” jika *password* tidak sesuai dengan *password* yang telah didaftarkan. |
| ***Post Conditions*** | Aktor berhasil login dan halaman dashboard ditampilkan oleh sistem. |

**Tabel 4.5 Use Case Scenario Buat Akun**

|  |  |
| --- | --- |
| Buat akun (PSB\_F\_02\_00) | |
| ***Actor*** | Pengunjung |
| ***Objective*** | Dapat terdaftar sebagai pengguna dalam sistem |
| ***Preconditions*** | Halaman buat akun telah ditampilkan |
| ***Main Flows*** | 1. Sistem menampilkan halaman buat akun dengan data masukan nama lengkap, NISN, *email* dan *password*.  2. Aktor mengisi data dengan lengkap.  3. Aktor menekan tombol buat akun.  4. Sistem menampilkan halaman *login*. |
| ***Alternative Flows*** | 1. Sistem menampilkan peringatan “NISN harus berjumlah 10 digit” jika jumlah digit NISN yang dimasukkan tidak sesuai.  2. Sistem menampilkan peringatan “Maaf, NISN telah terdaftar” jika NISN yang dimasukkan sama dengan NISN pengguna lain yang telah terdaftar. |
| ***Post Conditions*** | Aktor berhasil terdaftar sebagai pengguna dan sistem menampilkan halaman *login*. |

**Tabel 4.6 Use Case Scenario Ganti Password**

|  |  |
| --- | --- |
| Ganti *password* (PSB\_F\_03\_00) | |
| ***Actor*** | Pengunjung |
| ***Objective*** | Dapat mengganti *password* akun |
| ***Preconditions*** | Halaman ganti *password* telah ditampilkan |
| ***Main Flows*** | 1. Sistem menampilkan halaman ganti *password* dengan data NISN, *password* baru dan ulang *password*.  2. Aktor mengisi data dengan lengkap.  3. Aktor menekan tombol simpan.  4. Sistem menampilkan pemberitahuan “*password* berhasil diganti”.  5. Sistem menampilkan halaman *login*. |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “Maaf akun tidak ditemukan, harap periksa NISN yang dimasukkan telah benar” jika aktor memencet tombol simpan dan NISN yang dimasukkan tidak ditemukan dalam daftar pengguna. |
| ***Post Conditions*** | Pemberitahuan “*Password* berhasil diganti” dan halaman *login* berhasil ditampilkan. |

**Tabel 4.7 Use Case Scenario Logout**

|  |  |
| --- | --- |
| *Logout* (PSB\_F\_04\_00) | |
| ***Actor*** | Admin, santri aktif, peserta dan panitia PSB |
| ***Objective*** | Dapat menghilangkan hak akses pengguna |
| ***Preconditions*** | Aktor telah *login* terlebih dahulu |
| ***Main Flows*** | 1. Aktor memencet tombol *logout*.  2. Sistem menghapus data akun dari session  3. Sistem menampilkan halaman utama |
| ***Alternative Flows*** | - |
| ***Post Conditions*** | Hak akses aktor berhasil dihilangkan. |

**Tabel 4.8 Use Case Scenario Unduh Panduan**

|  |  |
| --- | --- |
| Unduh panduan (PSB\_F\_05\_00) | |
| ***Actor*** | Pengunjung |
| ***Objective*** | Dapat mengunduh panduan pendaftaran |
| ***Preconditions*** | Halaman informasi PSB sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol download panduan.  2. Sistem menampilkan pemberitahuan “Lanjut untuk mengunduh panduan pendaftaran?”.  3. Aktor memencet tombol “Yes”.  4. Sistem memproses unduhan. |
| ***Alternative Flows*** | Sistem akan menampilkan pemberitahuan “Unduhan dibatalkan” jika aktor memencet tombol “No”. |
| ***Post Conditions*** | Panduan pendaftaran berhasil di unduh. |

**Tabel 4.9** **Use Case Scenario Lihat Hasil Tes**

|  |  |
| --- | --- |
| Lihat hasil tes (PSB\_F\_06\_00) | |
| ***Actor*** | Pengunjung |
| ***Objective*** | Dapat melihat pengumuman atau hasil tes seleksi |
| ***Preconditions*** | Halaman utama sudah ditampilkan |
| ***Main Flows*** | 1. Aktor menekan tombol hasil tes.  2. Sistem menampilkan *form* cek hasil tes dengan data masukan yaitu NISN.  3. Aktor mengisi NISN dengan benar dan memencet tombol cek hasil.  4. Sistem menampilkan pesan “Selamat anda LULUS” apabila lulus dan sebaliknya akan menampilkan “Mohon maaf anda TIDAK LULUS”. |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “NISN tidak ditemukan, harap periksa kembali”, apabila NISN yang dimasukkan tidak terdaftar pada sistem. |
| ***Post Conditions*** | Hasil tes seleksi berhasil ditampilkan. |

**Tabel 4.10 Use Case Scenario Hubungi Admin**

|  |  |
| --- | --- |
| Hubungi admin (PSB\_F\_07\_00) | |
| ***Actor*** | Pengunjung |
| ***Objective*** | Dapat mengirim pesan kepada admin |
| ***Preconditions*** | Halaman hubungi admin sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet tombol hubungi kami.  2. Sistem menampilkan form hubungi admin dengan data meliputi nama, *email*, tentang dan isi pesan.  3. Aktor mengisi data dengan lengkap dan memencet tombol kirim.  4. Sistem menampilkan pemberitahuan ”Pesan terkirim”. |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “Harap mengisi semua kolom”, jika terdapat kolom data yang kosong. |
| ***Post Conditions*** | Pesan berhasil dikirim kepada admin. |

**Tabel 4.11 Use Case Scenario Lihat Informasi Sekolah**

|  |  |
| --- | --- |
| Lihat informasi sekolah (PSB\_F\_08\_00) | |
| ***Actor*** | Admin |
| ***Objective*** | Dapat menampilkan informasi sekolah |
| ***Preconditions*** | Halaman informasi sekolah sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet tombol profil sekolah.  2. Sistem menampilkan informasi nama sekolah, akreditasi, alamat, kodepos, nomor telepon, email, status sekolah, struktur organisasi sekolah, visi, misi, ekstrakurikuler sekolah dan sejarah singkat sekolah. |
| ***Alternative Flows*** | - |
| ***Post Conditions*** | Halaman informasi sekolah berhasil ditampilkan. |

**Tabel 4.12 Use Case Scenario Ubah Informasi Sekolah**

|  |  |
| --- | --- |
| Ubah informasi sekolah (PSB\_F\_09\_00) | |
| ***Actor*** | Admin |
| ***Objective*** | Dapat mengubah informasi sekolah yang akan ditampilkan |
| ***Preconditions*** | Halaman informasi sekolah sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet tombol ubah profil.  2. Sistem menampilkan halaman *edit* profil sekolah.  3. Aktor mengisi data yang ingin di ganti kemudian memencet tombol simpan.  4. Sistem menampilkan pemberitahuan “Anda yakin ingin melakukan perubahan?”.  5. Aktor memencet tombol “Yes”.  6. Sistem menampilkan pemberitahuan “Perubahan berhasil dilakukan”.  7. Sistem menampilkan halaman informasi sekolah. |
| ***Alternative Flows*** | 1. Sistem menampilkan pemberitahuan “Harap lengkapi semua kolom data”, jika terdapat kolom data yang kosong.  2. Sistem menampilkan pemberitahuan “Perubahan dibatalkan”, jika aktor memencet tombol “No”. |
| ***Post Conditions*** | Halaman ubah profil sekolah berhasil ditampilkan. |

**Tabel 4.13 Use Case Scenario Lihat Informasi PSB**

|  |  |
| --- | --- |
| Lihat informasi PSB (PSB\_F\_10\_00) | |
| ***Actor*** | Admin |
| ***Objective*** | Dapat menampilkan informasi seputar PSB |
| ***Preconditions*** | Halaman informasi PSB sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet tombol Informasi PSB.  2. Sistem menampilkan halaman informasi PSB .  3. Sistem menampilkan informasi PSB meliputi jadwal PSB, biaya, syarat pendaftaran yang berada pada sub menu. |
| ***Alternative Flows*** | - |
| ***Post Conditions*** | Halaman informasi PSB berhasil ditampilkan. |

**Tabel 4.14 Use Case Scenario Ubah Informasi PSB**

|  |  |
| --- | --- |
| Ubah informasi PSB (PSB\_F\_11\_00) | |
| ***Actor*** | Admin |
| ***Objective*** | Dapat mengubah informasi PSB yang akan ditampilkan |
| ***Preconditions*** | Halaman informasi PSB sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet tombol ubah informasi PSB.  2. Sistem menampilkan halaman *edit* informasi PSB.  3. Aktor mengisi data yang ingin di ganti kemudian memencet tombol simpan.  4. Sistem menampilkan pemberitahuan “Anda yakin ingin melakukan perubahan?”.  5. Aktor memencet tombol “Yes”.  6. Sistem menampilkan pemberitahuan “Perubahan berhasil dilakukan”.  7. Sistem menampilkan halaman informasi PSB. |
| ***Alternative Flows*** | 1. Sistem menampilkan pemberitahuan “Harap lengkapi semua kolom data”, jika terdapat kolom data yang kosong.  2. Sistem menampilkan pemberitahuan “Perubahan dibatalkan”, jika aktor memencet tombol “No”. |
| ***Post Conditions*** | Informasi PSB berhasil diperbaharui. |

**Tabel 4.15 Use Case Scenario Lihat Daftar Pesan**

|  |  |
| --- | --- |
| Lihat daftar pesan (PSB\_F\_12\_00) | |
| ***Actor*** | Admin |
| ***Objective*** | Dapat melihat daftar pesan dari pengunjung |
| ***Preconditions*** | Halaman daftar pesan sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet tombol pesan.  2. Sistem menampilkan daftar pesan dari pengunjung dengan tampilan data meliputi tanggal, nama, *email*, tentang pesan dan tombol hapus dan balas pada data aksi. |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “Tidak terdapat pesan”, jika belum terdapat pengunjung yang mengirim pesan. |
| ***Post Conditions*** | Daftar pesan dari pengunjung berhasil ditampilkan. |

**Tabel 4.16 Use Case Scenario Hapus Pesan**

|  |  |
| --- | --- |
| Hapus pesan (PSB\_F\_13\_00) | |
| ***Actor*** | Admin |
| ***Objective*** | Dapat menghilangkan pesan dari daftar pesan |
| ***Preconditions*** | Halaman daftar pesan sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet tombol hapus pada data aksi.  2. Sistem menampilkan pemberitahuan “Yakin ingin menghapus pesan?”.  3. Aktor memencet tombol “Yes”.  4. Sistem menampilkan pemberitahuan “Pesan berhasil dihapus”. |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “Proses dibatalkan”, jika aktor memencet tombol “No”. |
| ***Post Conditions*** | Pesan berhasil dihilangkan dari daftar pesan. |

**Tabel 4.17 Use Case Scenario Balas Pesan**

|  |  |
| --- | --- |
| Balas pesan (PSB\_F\_14\_00) | |
| ***Actor*** | Admin |
| ***Objective*** | Dapat mengirim pesan balasan |
| ***Preconditions*** | Halaman balas pesan sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet tombol balas pada data aksi.  2. Sistem menampilkan *form* balas pesan dengan data meliputi *email*, pesan yang akan dibalas dan isi balasan.  3. Aktor memencet tombol “Kirim”.  4. Sistem menampilkan pemberitahuan “Pesan berhasil dikirim”.  5. Sistem menampilkan halaman daftar pesan. |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “Harap semua kolom diisi terlebih dahulu”, jika terdapat kolom data yang kosong. |
| ***Post Conditions*** | Pemberitahuan “Pesan berhasil dikirim” berhasil ditampilkan. |

**Tabel 4.18 Use Case Scenario Lihat Daftar Pengguna**

|  |  |
| --- | --- |
| Lihat daftar pengguna (PSB\_F\_15\_00) | |
| ***Actor*** | Admin |
| ***Objective*** | Dapat melihat daftar pengguna dalam sistem |
| ***Preconditions*** | Halaman daftar pengguna sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet tombol pengguna.  2. Sistem menampilkan halaman daftar pengguna dengan tampilan data meliputi nama, NISN, *email*, *password* dan tombol hapus pada data aksi. |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “Belum terdapat pengguna, silahkan melakukan penambahan pengguna jika dibutuhkan”, jika belum terdapat pengguna yang terdaftar dalam sistem. |
| ***Post Conditions*** | Daftar pengguna berhasil ditampilkan. |

**Tabel 4.19 Use Case Scenario Tambah Pengguna**

|  |  |
| --- | --- |
| Tambah pengguna (PSB\_F\_16\_00) | |
| ***Actor*** | Admin |
| ***Objective*** | Dapat menambah pengguna sistem |
| ***Preconditions*** | Halaman daftar pengguna sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet tombol tambah.  2. Sistem menampilkan *form* tambah pengguna dengan data meliputi NIP, *email* dan *password* untuk panitia data NISN, NIS, *email* dan *password* untuk santri aktif.  3. Aktor memencet tombol “Tambah”.  4. Sistem menampilkan pemberitahuan “Yakin ingin menambah pengguna?”.  5. Aktor memencet tombol “Yes”.  6. Sistem menampilkan pemberitahuan “Pengguna berhasil ditambah”. |
| ***Alternative Flows*** | 1. Sistem menampilkan pemberitahuan “Harap masukkan NIP dengan benar”, jika NIP tidak berjumlah 18 digit.  2. Sistem menampilkan pemberitahuan “Harap masukkan NISN dengan benar”, jika NISN tidak berjumlah 10 digit.  3. Sistem menampilkan pemberitahuan “Penambahan dibatalkan”, jika aktor memencet tombol “No”. |
| ***Post Conditions*** | Pemberitahuan “Pengguna berhasil ditambah” berhasil ditampilkan. |

**Tabel 4.20 Use Case Scenario Hapus Pengguna**

|  |  |
| --- | --- |
| Hapus pengguna (PSB\_F\_17\_00) | |
| ***Actor*** | Admin |
| ***Objective*** | Dapat menghapus pengguna dari daftar pengguna |
| ***Preconditions*** | Halaman daftar pengguna sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet tombol hapus pada data aksi.  2. Sistem menampilkan pemberitahuan “Yakin ingin menghapus pengguna?”.  3. Aktor memencet tombol “Yes”.  4. Sistem menampilkan pemberitahuan “Pengguna berhasil dihapus”. |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “Penambahan dibatalkan”, jika aktor memencet tombol “No”. |
| ***Post Conditions*** | Pengguna berhasil dihapus dari daftar pengguna. |

**Tabel 4.21 Use Case Scenario Pembayaran**

|  |  |
| --- | --- |
| Pembayaran (PSB\_F\_18\_00) | |
| ***Actor*** | Peserta PSB |
| ***Objective*** | Dapat melakukan pembayaran sekolah |
| ***Preconditions*** | Halaman pembayaran sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memilih/mencentang tagihan biaya sekolah sesuai gender.  2. Sistem menampilkan total tagihan.  3. Aktor memencet tombol bayar.  4. Sistem menampilkan form untuk memilih metode pembayaran.  5. Aktor memilih salah satu metode pembayaran.  6. Sistem menampilkan form tagihan yang berisi total yang harus dibayar, nomor rekening tujuan dan tata cara pembayaran.  7. Aktor memencet tombol “OK”  8. Sistem menampilkan status pembayaran. |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “Harap centang tagihan yang sesuai”, jika aktor memencet tombol “Bayar” namun belum memilih tagihan yang sesuai. |
| ***Post Conditions*** | Pengguna berhasil melakukan proses pembayaran. |

**Tabel 4.22 Use Case Scenario Lihat Status Pembayaran**

|  |  |
| --- | --- |
| Lihat Status Pembayaran (PSB\_F\_18\_00) | |
| ***Actor*** | Peserta PSB |
| ***Objective*** | Dapat melihat status pembayaran sekolah |
| ***Preconditions*** | Aktor telah melakukan proses pembayaran |
| ***Main Flows*** | 1. Sistem menampilkan tabel dengan data berupa nama, tanggal transaksi, metode pembayaran, jenis bank, total yang harus dibayar, rekening tujuan, status pembayaran dan action.  2. Sistem menampilkan total tagihan.  3. Aktor memencet tombol bayar.  4. Sistem menampilkan form untuk memilih metode pembayaran.  5. Aktor memilih salah satu metode pembayaran.  6. Sistem menampilkan form tagihan yang berisi total yang harus dibayar, nomor rekening tujuan dan tata cara pembayaran.  7. Aktor memencet tombol “OK”  8. sistem menampilkan halaman pembayaran dan status pembayaran . |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “Harap centang tagihan yang sesuai”, jika aktor memencet tombol “Bayar” namun belum memilih tagihan yang sesuai. |
| ***Post Conditions*** | Pengguna berhasil melakukan proses pembayaran. |

**Tabel 4.23 Use Case Scenario Isi Data Diri**

|  |  |
| --- | --- |
| Isi *form* pendaftaran (PSB\_F\_19\_00) | |
| ***Actor*** | Peserta PSB |
| ***Objective*** | Dapat mengisi *form* data pendaftaran |
| ***Preconditions*** | Halaman dashboard sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet tombol isi *form* pendaftaran.  2. Sistem menampilkan halaman *form* pendaftaran dengan isian data diri meliputi nama santri, NISN, jenis kelamin, tempat tanggal lahir, jumlah saudara, alamat, nomor handphone, asal sekolah, alamat sekolah, tinggal dengan siapa, bidang yang digemari, cita-cita, alat transportasi ke sekolah, daftar prestasi dan organisasi. Data orang tua meliputi nama ayah dan ibu, alamat orang tua, nomor handphone orang tua, pekerjaan ayah dan ibu, pendidikan ayah dan ibu dan oenghasilan orang tua. Data wali meliputi nama, alamat, nomor handphone , pekerjaan, dan penghasilan wali.  3. Aktor mengisi data dengan lengkap dan benar.  4. Aktor memencet tombol “Simpan”.  5. Sistem menampilkan pemberitahuan “Data berhasil disimpan”. |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “Proses tidak dapat dilakukan, harap lengkapi data terlebih dahulu”, jika terdapat kolom data yang kosong. |
| ***Post Conditions*** | *Form* pendaftaran berhasil diisi. |

**Tabel 4.24 Use Case Scenario Lihat Profil**

|  |  |
| --- | --- |
| Lihat profil (PSB\_F\_20\_00) | |
| ***Actor*** | Peserta PSB |
| ***Objective*** | Dapat melihat informasi data diri |
| ***Preconditions*** | Halaman profil sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet link profil.  2. Sistem menampilkan halaman profil berisi informasi data diri, orang tua dan wali. |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “Belum terdapat informasi data diri”, jika aktor memencet link profil namun belum mengisi *form* pendaftaran. |
| ***Post Conditions*** | Informasi data diri aktor berhasil di tampilkan. |

**Tabel 4.25 Use Case Scenario Ubah Profil**

|  |  |
| --- | --- |
| Ubah profil (PSB\_F\_21\_00) | |
| ***Actor*** | Peserta PSB |
| ***Objective*** | Dapat mengubah data diri |
| ***Preconditions*** | Halaman profil sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet tombol ubah profil.  2. Sistem menampilkan informasi data meliputi data diri, orang tua dan wali.  3. Aktor mengisi data yang ingin diubah dan memencet tombol simpan.  4. Sistem menampilkan pemberitahuan “Yakin ingin mengubah data?”.  5. Aktor memencet tombol “Yes”.  6. Sistem menampilan pemberitahuan “Data berhasil diubah”. |
| ***Alternative Flows*** | 1. Sistem menampilkan pemberitahuan “Harap lengkapi semua data”, jika aktor memencet tombol simpan namun terdapat kolom data yang kosong.  2. Sistem menampilkan pemberitahuan “Proses dibatalkan”, jika aktor memencet tombol “No”. |
| ***Post Conditions*** | Pemberitahuan “Data berhasil diubah” berhasil ditampilkan. |

**Tabel 4.26 Use Case Scenario Cetak Kartu Peserta**

|  |  |
| --- | --- |
| Cetak kartu peserta (PSB\_F\_22\_00) | |
| ***Actor*** | Peserta PSB |
| ***Objective*** | Dapat mencetak kartu peserta PSB |
| ***Preconditions*** | Halaman profil sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet tombol cetak kartu peserta.  2. Sistem menampilkan pemberitahuan “Lanjut untuk mengunduh kartu peserta?”  3. Aktor memencet tombol “Yes”.  4. Sistem menampilan pemberitahuan “Kartu peserta berhasil di unduh”. |
| ***Alternative Flows*** | 1. Sistem menampilkan pemberitahuan “Unduhan dibatalkan”, jika aktor memencet tombol “No”.  2. Sistem menampilkan pemberitahuan “Proses tidak dapat dilakukan, data diri belum diverifikasi”, jika status dara diri belum di verifikasi panitia PSB. |
| ***Post Conditions*** | Pemberitahuan “Kartu peserta berhasil diunduh” berhasil ditampilkan. |

**Tabel 4.27 Use Case Scenario Lihat Tes Seleksi**

|  |  |
| --- | --- |
| Lihat tes seleksi (PSB\_F\_23\_00) | |
| ***Actor*** | Peserta PSB |
| ***Objective*** | Dapat melihat halaman untuk memulai tes seleksi |
| ***Preconditions*** | Halaman tes seleksi sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet link tes seleksi.  2. Sistem menampilkan halaman tes seleksi untuk dapat memulai tes seleksi.  3. Sistem menampilkan tombol mulai tes seleksi. |
| ***Alternative Flows*** | 1. Sistem menampilkan pemberitahuan “Tes seleksi belum dibuka”, jika aktor memencet tombol tes seleksi pada tanggal selain tanggal yang telah ditetapkan. |
| ***Post Conditions*** | Halaman untuk memulai tes berhasil ditampilkan. |

**Tabel 4.28 Use Case Scenario Tes Seleksi**

|  |  |
| --- | --- |
| Mulai tes (PSB\_F\_24\_00) | |
| ***Actor*** | Peserta PSB |
| ***Objective*** | Dapat memulai tes seleksi |
| ***Preconditions*** | Halaman tes seleksi sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet tombol mulai mengerjakan.  2. Sistem menampilkan pemberitahuan “Tes seleksi dikerjakan selama 25 menit dalam bentuk essay, mulai tes tes seleksi?.  3. Aktor memencet tombol “Ok”  4. Sistem menampilkan halaman soal tes seleksi.  5. Aktor mengetikkan jawaban atau melampirkan *file* jawaban pada kolom jawaban dan memencet tombol submit.  6. Sistem menampilkan pemberitahuan “Jawaban berhasil dikirim” |
| ***Alternative Flows*** | 1. Sistem menampilkan pemberitahuan “Tes seleksi tidak dapat dimulai, harap periksa berkas dan data diri apakah sudah di verifikasi”, jika status berkas prasayarat dan data diri belum diverifikasi.  2. Sistem menampilkan pemberitahuan “Silahkan lampirkan file dengan format pdf”, jika aktor melampikan file selain format pdf. |
| ***Post Conditions*** | Tes seleksi berhasil dikerjakan. |

**Tabel 4.29 Use Case Scenario Pilih Program Studi**

|  |  |
| --- | --- |
| Pilih program studi (PSB\_F\_25\_00) | |
| ***Actor*** | Peserta PSB |
| ***Objective*** | Dapat memilih program studi yang diinginkan |
| ***Preconditions*** | Aktor telah selesai mengerjakan tes seleksi |
| ***Main Flows*** | 1. Sistem menampilkan *form* memilih program studi, meliputi pilihan program studi pertama dan program studi kedua.  3. Aktor memencet tombol selesai.  4. Sistem menampilkan pemberitahuan “Yakin dengan pilihan program studi?”.  5. Aktor memencet tombol “Yes”.  6. Sistem menampilkan pemberitahuan “Program studi berhasil dipilih”. |
| ***Alternative Flows*** | 1. Sistem menampilkan pemberitahuan “Harap isi semua kolom data”, jiika terdapat kolom data yang kosong.  2. Sistem menampilkan pemberitahuan “Proses dibatalkan”, jika aktor memencet tombol “No”. |
| ***Post Conditions*** | Aktro berhasil memilih program studi. |

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

|  |  |
| --- | --- |
| Lihat daftar prasyarat (PSB\_F\_26\_00) | |
| ***Actor*** | Peserta PSB |
| ***Objective*** | Dapat melihat daftar prasyarat pendaftaran |
| ***Preconditions*** | Halaman *upload* berkas sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet link *upload* berkas.  2. Sistem menampilkan halaman daftar prasayarat.  3. Sistem menampilkan *form* untuk dapat melakukan *upload* berkas prasyarat meliputi ijazah dan SHUN (sudah dilegalisir), kartu peserta UN, kartu NISN, kartu keluarga, kartu tanda penduduk (KTP) orang tua, kartu perlindungan sosial (opsional), kartu program keluarga harapan (PKH)/kartu Indonesia pintar (KIP), surat keterangan tidak mampu terbaru dan akta kelahiran |
| ***Alternative Flows*** | - |
| ***Post Conditions*** | Daftar prasyarat berhasil ditampilkan. |

**Tabel 4.31 Use Case Scenario Unggah Berkas Prasyarat**

|  |  |
| --- | --- |
| Unggah berkas prasyarat (PSB\_F\_27\_00) | |
| ***Actor*** | Peserta PSB |
| ***Objective*** | Dapat mengunggah berkas prasyarat pendaftaran |
| ***Preconditions*** | Halaman *upload* berkas sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet memencet tombol browse pada file yang ingin diunggah.  2. Aktor memilih berkas yang ingin di unggah pada komputer.  3. Sistem menerima *file* unggahan.  4. Aktor memencet tombol “simpan”.  5. Sistem menampilkan pemberitahuan “Yakin ingin menyimpan berkas?”.  6. Aktor memencet tombol “Yes”.  5. Sistem menampilkan pemberitahuan “Berkas berhasil disimpan”. |
| ***Alternative Flows*** | 1. Sistem menampilkan pemberitahuan “Harap periksa kembali berkas, pastikan *form*at *file* dalam bentuk pdf”, jika aktor mengunggah berkas dengan *form*at selain pdf.  2. Sistem menampilkan pemberitahuan “Proses dibatilkan”, jika aktor memencet tombol “No”. |
| ***Post Conditions*** | Aktor berhasil mengunggah berkas prasyarat. |

**Tabel 4.32 Use Case Scenario Ubah Berkas Prasyarat**

|  |  |
| --- | --- |
| Ubah berkas prasyarat (PSB\_F\_28\_00) | |
| ***Actor*** | Peserta PSB |
| ***Objective*** | Dapat mengunggah ulang berkas prasyarat pendaftaran |
| ***Preconditions*** | Halaman *upload* berkas sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet memencet tombol *edit* berkas.  2. Sistem menampilkan halaman untuk mengunggah ulang berkas.  3. Aktor memencet tombol silang pada samping *file*.  4. Aktor memencet tombol browse  5. Sistem menerima *file* unggahan.  6. Aktor memencet tombol simpan berkas.  7. Sistem menampilkan pemberitahuan “Yakin ingin menyimpan berkas?”.  8. Aktor memencet tombol “Yes”.  9. Sistem menampilkan pemberitahuan “Berkas berhasil diperbaharui”. |
| ***Alternative Flows*** | 1. Sistem menampilkan pemberitahuan “Harap periksa kembali berkas, pastikan *form*at *file* dalam bentuk pdf”, jika aktor mengunggah berkas dengan *form*at selain pdf.  2. Sistem menampilkan pemberitahuan “Proses dibatilkan”, jika aktor memencet tombol “No”. |
| ***Post Conditions*** | Aktor berhasil mengunggah berkas prasyarat yang baru. |

**Tabel 4.33 Use Case Scenario Lihat Daftar Peserta PSB**

|  |  |
| --- | --- |
| Lihat daftar peserta PSB (PSB\_F\_29\_00) | |
| ***Actor*** | Panitia PSB |
| ***Objective*** | Dapat melihat seluruh daftar peserta PSB yang telah terdaftar pada sistem |
| ***Preconditions*** | 1. Aktor telah terautentikasi sebagai panitia PSB  2. Terdapat daftar santri yang telah terdaftar pada sistem. |
| ***Main Flows*** | 1. Aktor memencet memencet link calon santri.  2. Sistem menampilkan halaman daftar peserta dengan data meliputi tanggal pendaftaran, nama, *NISN*, asal sekolah tanggal lahir, status dan tombol detail pada data aksi.  3. Sistem menampilkan tombol cetak daftar peserta.  4. Sistem menampilkan tombol lihat detail data peserta PSB.  5. Sistem menampilkan kolom pencarian data. |
| ***Alternative Flows*** | 1. Sistem menampilkan pemberitahuan “Belum terdapat calon santri”, jika aktor memencet link calon santri namun belum terdapat peserta PSB. |
| ***Post Conditions*** | Daftar peserta PSB atau calon santri berhasil ditampilkan. |

**Tabel 4.34 Use Case Scenario Cetak Daftar Peserta**

|  |  |
| --- | --- |
| Cetak daftar peserta PSB (PSB\_F\_30\_00) | |
| ***Actor*** | Panitia PSB |
| ***Objective*** | Dapat mencetak daftar peserta PSB yang terdaftar |
| ***Preconditions*** | Halaman daftar peserta sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet tombol cetak daftar peserta.  2. Sistem menampilkan pemberitahuan “Lanjut untuk mencetak?”.  3. Aktor memencet tombol “Yes”.  4. Sistem memproses pencetakan daftar peserta PSB. |
| ***Alternative Flows*** | 1. Sistem menampilkan pemberitahuan “Belum terdapat calon santri”, jika aktor memencet tombol cetak daftar peserta dan belum terdapat daftar peserta PSB.  2. Sistem menampilkan pemberitahuan “Proses dibatalkan”, jika aktor memencet tombol “No”. |
| ***Post Conditions*** | Daftar peserta PSB berhasil dicetak. |

**Tabel 4.35 Use Case Scenario Lihat Detail Peserta PSB**

|  |  |
| --- | --- |
| Lihat detail data peserta PSB (PSB\_F\_31\_00) | |
| ***Actor*** | Panitia PSB |
| ***Objective*** | Dapat melihat lebih detail data peserta PSB |
| ***Preconditions*** | Halaman detail data peserta PSB sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet tombol detail pada data aksi.  2. Sistem menampilkan halaman detai data peserta PSB.  3. Sistem menampilkan data dalam 3 sub menu diantaranya data diri, berkas prasyarat dan bukti pembayaran.  4. Sistem menampilkan tombol verifikasi disetiap sub menu. |
| ***Alternative Flows*** | - |
| ***Post Conditions*** | Detail data peserta berhasil ditampilkan. |

**Tabel 4.36 Use Case Scenario Verifikasi Pendaftaran**

|  |  |
| --- | --- |
| Verifikasi pendaftaran (PSB\_F\_32\_00) | |
| ***Actor*** | Panitia PSB |
| ***Objective*** | Dapat melakukan verifikasi pada data pendaftaran peserta PSB |
| ***Preconditions*** | Peserta PSB telah meng-upload berkas prasyarat |
| ***Main Flows*** | 1. Aktor memencet tombol “Verifikasi”.  2. Sistem menampilkan pemberitahuan “Yakin untuk melakukan verifikas?”.  3. Sistem memencet tombol “Yes”.  4. Sistem menampilkan pemberitahuan “Berhasil diverifikasi” |
| ***Alternative Flows*** | 1. Sistem menampilkan pemberitahuan “Proses dibatalkan”, jika aktor memencet tombol “No” |
| ***Post Conditions*** | Pemberitahuan “Berhasil diverifikasi” berhasil ditampilkan. |

**Tabel 4.37 Use Case Scenario Lihat Daftar Arsip Data Santri**

|  |  |
| --- | --- |
| Lihat daftar arsip data santri (PSB\_F\_33\_00) | |
| ***Actor*** | Panitia PSB |
| ***Objective*** | Dapat melihat daftar data santri dalam data arsip |
| ***Preconditions*** | Halaman arsip sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet link arsip.  2. Sistem menampilkan halaman arsip data dan menampilkan daftar data santri dengan tampilan data meliputi nama, NISN, asal sekolah dan tanggal lahir.  3. Sistem menampilkan tombol detail data santri dan kolom pencarian data. |
| ***Alternative Flows*** | 1. Sistem menampilkan pemberitahuan “Tidak terdapat data arsip”, jika aktor memencet link arsip dan belum terdapat daftar data arsip. |
| ***Post Conditions*** | Daftar arsip data santri berhasil ditampilkan. |

**Tabel 4.38 Use Case Scenario Lihat Detail Arsip Data Santri**

|  |  |
| --- | --- |
| Lihat detail arsip data santri (PSB\_F\_34\_00) | |
| ***Actor*** | Panitia PSB |
| ***Objective*** | Dapat melihat lebih detail data diri santri yang di arsip |
| ***Preconditions*** | Halaman arsip sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet tombol detail.  2. Sistem menampilkan halaman detai data santri.  3. Sistem menampilkan data dalam 3 sub menu diantaranya data diri, berkas prasyarat dan bukti pembayaran. |
| ***Alternative Flows*** | - |
| ***Post Conditions*** | Detail data santri pada arsip data berhasil ditampilkan. |

**Tabel 4.39 Use Case Scenario Lihat Daftar Soal Tes**

|  |  |
| --- | --- |
| Lihat daftar soal tes (PSB\_F\_35\_00) | |
| ***Actor*** | Panitia PSB |
| ***Objective*** | Dapat melihat daftar soal tes seleksi |
| ***Preconditions*** | Halaman tes akademik sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet link tes akademik.  2. Sistem menampilkan *form* tambah soal dan daftar soal tes seleksi yang telah dibuat.  3. Sistem menampilkan daftar soal dengan data meliputi soal, bidang dan tombol *edit*, hapus dan tombol kirim pada data aksi. |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “Belum terdapat soal, silahkan melakukan penambahan soal jika dibutuhkan”, jika belum terdapat soal tes seleksi. |
| ***Post Conditions*** | Daftar soal tes berhasil ditampilkan. |

**Tabel 4.40 Use Case Scenario Tambah Soal Tes**

|  |  |
| --- | --- |
| Tambah soal tes (PSB\_F\_36\_00) | |
| ***Actor*** | Panitia PSB |
| ***Objective*** | Dapat menambah soal tes seleksi |
| ***Preconditions*** | Halaman tes akademik sudah ditampilkan |
| ***Main Flows*** | 1. Aktor mengisi kolom bidang dan pertanyaan dengan lengkap pada form tambah soal.  2. Aktor memencet tombol tambah soal.  3. Sistem menampilkan pemberitahuan “Soal berhasil ditambah”. |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “Harap isi kolom data dengan lengkap”, jika terdapat kolom data yang kosong. |
| ***Post Conditions*** | Berhasil menambah soal tes seleksi. |

**Tabel 4.41 Use Case Scenario Ubah Soal Tes**

|  |  |
| --- | --- |
| Ubah soal tes (PSB\_F\_37\_00) | |
| ***Actor*** | Panitia PSB |
| ***Objective*** | Dapat mengubah soal tes yang sudah ditambahkan |
| ***Preconditions*** | Halaman tes akademik sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet tombol *edit* pada data aksi.  2. Sistem menampilkan *form* untuk mengubah soal.  3. Aktor mengubah data sesuai dengan yang diinginkan.  4. Aktor memencet tombol simpan.  5. Sistem menampilkan pemberitahuan “Yakin ingin mengubah?”.  6. Aktor memencet tombol “Yes”. |
| ***Alternative Flows*** | 1. Sistem menampilkan pemberitahuan “Proses dibatalkan”, jika aktor memencet tombol “No”  2. sistem menampilkan pemberitahuan “Isi kolom data dengan lengkap”, jika terdapat kolom data yang kosong. |
| ***Post Conditions*** | Soal tes seleksi berhasil diubah. |

**Tabel 4.42 Use Case Scenario Kirim Soal Tes**

|  |  |
| --- | --- |
| Kirim soal tes (PSB\_F\_38\_00) | |
| ***Actor*** | Panitia PSB |
| ***Objective*** | Dapat mengirim soal tes yang telah dibuat ke peserta PSB |
| ***Preconditions*** | Halaman tes akademik sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memilih soal yang ingin dikirim.  2. Aktro memencet tombol kirim pada data aksi.  3. Sistem menampilkan pemberitahuan “Lanjutu untuk mengirim soal?”  4. Aktor memencet tombol “Yes”.  5. Sistem menampilkan pemberitahuan “Berhasil dikirim”. |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “Proses dibatalkan”, jika aktor memencet tombol “No”. |
| ***Post Conditions*** | Soal tes berhasil dikirim ke peserta PSB. |

**Tabel 4.43 Use Case Scenario Hapus Soal Tes**

|  |  |
| --- | --- |
| Hapus soal tes (PSB\_F\_39\_00) | |
| ***Actor*** | Panitia PSB |
| ***Objective*** | Dapat menghapus soal dari daftar soal tes |
| ***Preconditions*** | Halaman tes akademik sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memilih soal yang ingin dihapus.  2. Aktor memencet tombol hapus pada data aksi.  3. Sistem menampilkan pemberitahuan “Yakin ingin menghapus soal?”  4. Aktor memencet tombol “Yes”.  5. Sistem menampilkan pemberitahuan “Berhasil dihapus”. |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “Proses dibatalkan”, jika aktor memencet tombol “No”. |
| ***Post Conditions*** | Soal tes berhasil dihapus dari daftar soal. |

**Tabel 4.44 Use Case Scenario Lihat Daftar Jawaban**

|  |  |
| --- | --- |
| Lihat daftar jawaban (PSB\_F\_40\_00) | |
| ***Actor*** | 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 4.45 Use Case Scenario Lihat Detail Jawaban**

|  |  |
| --- | --- |
| Lihat detail jawaban (PSB\_F\_41\_00) | |
| ***Actor*** | Panitia PSB |
| ***Objective*** | Dapat melihat detail jawaban dari peserta PSB |
| ***Preconditions*** | Halaman detail jawaban sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet tombol lihat detail pada data aksi.  2. Sistem menampilkan halaman detail jawaban dengan data meliputi, nama, NISN, program studi yang dipilih dan jawaban dari peserta PSB .  3. Aktor membuka *file* jawaban dari peserta. |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “Tidak terdapat jawaban”, jika aktor memencet tombol lihat detail dan peserta tidak melakukan tes seleksi. |
| ***Post Conditions*** | Detail jawaban peserta PSB berhasil ditampilkan. |

**Tabel 4.46 Use Case Scenario Nilai Jawaban**

|  |  |
| --- | --- |
| Nilai jawaban (PSB\_F\_42\_00) | |
| ***Actor*** | Panitia PSB |
| ***Objective*** | Dapat memberi penilaian terhadap jawaban peserta PSB |
| ***Preconditions*** | Halaman detail jawaban sudah ditampilkan |
| ***Main Flows*** | 1. Sistem menampilkan *form* penialian pada halaman detail jawaban.  2. Aktor memasukkan penilaian pada *form* penilaian dengan benar, dengan data meliputi program studi yang terpilih dan nilai atas jawaban tes.  3. Aktor memencet tombol kirim.  4. Sistem menampilkan pemberitahuan “Yakin dengan penilaian tersebut?”.  5. Aktor memencet tombol “Yes”.  6. Sistem menampilkan pemberitahuan “Penilaian berhasil dikirim”. |
| ***Alternative Flows*** | 1. Sistem menampilkan pemberitahuan “Harap lengkapi kolom data”, jika aktor memencet tombol kirim dan terdapat kolom data yang kosong.  2. Sistem menampilkan pemberitahuan “Proses dibatalkan”, jika aktro memencet tombol “No”.  3. Sistem menampilkan pemberitahuan “Masukkan nilai dengan angka 1-100”, jika aktor memasukkan angka melebihi 100 dan kurang dari 1.  4. Sistem menampilkan pemberitahuan “Pastikan pilihan program studi sudah benar”, jika aktor memasukkan program studi selain IPA, IPS dan keagamaan. |
| ***Post Conditions*** | Pemberitahuan “Penilaian berhasil dikirim” berhasil ditampilkan. |

**Tabel 4.47 Use Case Scenario Input Nilai Tes Akademik**

|  |  |
| --- | --- |
| Input nilai tes akademik (PSB\_F\_43\_00) | |
| ***Actor*** | Panitia PSB |
| ***Objective*** | Dapat memasukkan nilai dari jawaban peserta PSB |
| ***Preconditions*** | 1. Tes akademik sudah dilakukan  2. Penilaian tes akademik sudah diakukan |
| ***Main Flows*** | 1. Aktor memencet link tab menu input nilai.  2. Sistem menampilkan halaman input nila tes akademik.  3. Sistem menampilkan daftar peserta PSB dalam bentuk tabel dengan data meliputi tanggal pendaftaran, nama, NISN, program studi dan kolom penilaian pada data nilai.  4. Aktor memasukkan nilai pada kolom nilai pada data nilai.  5. Aktor memencet tombol “Simpan”.  6. Sistem menampilkan pemberitahuan “Nilai berhasil disimpan”. |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “Masukkan nilai dengan angka 1-100”, jika aktor memasukkan angka melebihi 100, kurang dari 1 dan memasukkan karakter selain angka. |
| ***Post Conditions*** | Pemberitahuan “Nilai berhasil disimpan” berhasil ditampilkan. |

**Tabel 4.48 Use Case Scenario Input Nilai Tes Wawancara**

|  |  |
| --- | --- |
| Input nilai tes wawancara (PSB\_F\_44\_00) | |
| ***Actor*** | Panitia PSB |
| ***Objective*** | Dapat memasukkan nilai dari tes wawancara |
| ***Preconditions*** | Halaman input nilai tes wawancara sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet link tes wawancara.  2. Sistem menampilkan halaman input nila tes wawancara.  3. Sistem menampilkan daftar peserta PSB dalam bentuk tabel dengan data meliputi tanggal pendaftaran, nama, NISN, asal sekolah tanggal lahir dan kolom penilaian pada data nilai.  4. Aktor memasukkan nilai pada kolom nilai pada data nilai.  5. Aktor memencet tombol “Simpan”.  6. Sistem menampilkan pemberitahuan “Nilai berhasil disimpan”. |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “Masukkan nilai dengan angka 1-100”, jika aktor memasukkan angka melebihi 100, kurang dari 1 dan memasukkan karakter selain angka. |
| ***Post Conditions*** | Pemberitahuan “Nilai berhasil disimpan” berhasil ditampilkan. |

**Tabel 4.49 Use Case Scenario Lihat Akumulasi Tes**

|  |  |
| --- | --- |
| Lihat akumulasi tes (PSB\_F\_45\_00) | |
| ***Actor*** | Panitia PSB |
| ***Objective*** | Dapat melihat hasil akumulasi penilaian tes wawancara dan akademik |
| ***Preconditions*** | Halaman akumulasi tes sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet link akumulasi tes.  2. Sistem menampilkan halaman akumulasi tes.  3. Sistem menampilkan akumulasi tes dalam bentuk tabel dengan data meliputi nama, NISN, program studi, tes akademik, tes wawancara dan status lulus/tidak lulus.  4. Aktor melihat status dari hasil akumulasi tes lulus/tidak lulus. |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “Akumulasi tes belum bisa dilakukan”, jika aktor memencet link akumulasi tes namun niali tes belum dimasukkan oleh panitia PSB. |
| ***Post Conditions*** | Halaman akumlasi tes berhasil ditampilkan. |

**Tabel 4.50 Use Case Scenario Cetak Hasil Akumulasi**

|  |  |
| --- | --- |
| Cetak hasil akumulasi (PSB\_F\_46\_00) | |
| ***Actor*** | Panitia PSB |
| ***Objective*** | Dapat mencetak hasil akumulasi tes seleksi akademik dan wawancara |
| ***Preconditions*** | Halaman akumulasi tes sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet tombol cetak hasil.  2. Sistem menampilkan pemberitahuan “Lanjut untuk mencetak hasil penerimaan santri baru?”.  3. Aktor memencet “Yes”.  4. Sistem memproses data untuk dicetak. |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “Proses dibatalkan”, jika aktor memencet tombol “No”. |
| ***Post Conditions*** | Hasil akumulasi tes berhasil dicetak. |

**Tabel 4.51 Use Case Scenario Lihat Daftar Santri Aktif**

|  |  |
| --- | --- |
| Lihat daftar santri aktif (PSB\_F\_47\_00) | |
| ***Actor*** | Panitia PSB |
| ***Objective*** | Dapat melihat daftar santri yang berstatus aktif |
| ***Preconditions*** | Halaman daftar santri aktif sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet link santri aktif.  2. Sistem menampilkan halaman daftar dengan data dalam betuk tabel, meliputi data nama, NIS, program studi, kelas, status SPP dan tombol lihat detail dan arsip pada data aksi. |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “Tidak terdapat santri aktif”, jika belum terdapat data pada tabel santri aktif. |
| ***Post Conditions*** | Daftar santri aktif berhasil ditampilkan. |

**Tabel 4.52 Use Case Scenario Cetak Data Santri**

|  |  |
| --- | --- |
| Cetak data santri (PSB\_F\_48\_00) | |
| ***Actor*** | Panitia PSB |
| ***Objective*** | Dapat mencetak seluruh data santri yang berstatus aktif |
| ***Preconditions*** | Halaman daftar santri aktif sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet tombol cetak data.  2. Sistem menampilkan pemberitahuan “lanjut untuk mencetak data santri aktif?”.  3. Aktor memencet “Yes”.  4. Sistem memproses data untuk dicetak. |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “Proses dibatalkan”, jika aktor memencet “No”. |
| ***Post Conditions*** | Daftar santri berstatus aktifberhasil dicetak. |

**Tabel 4.53 Use Case Scenario Lihat Detail Data Santri**

|  |  |
| --- | --- |
| Lihat detail data santri (PSB\_F\_49\_00) | |
| ***Actor*** | Panitia PSB |
| ***Objective*** | Dapat melihat detail data diri santri yang berstatus aktif |
| ***Preconditions*** | Halaman detail data santri sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet tombol lihat detail.  2. Sistem menampilkan halaman detail data diri santri.  3. Sistem menampilkan informasi data diri dan SPP.  4. Aktor memilih informasi yang ingin diketahui. |
| ***Alternative Flows*** | - |
| ***Post Conditions*** | Detail data diri santri berstatus aktif berhasil ditampilkan. |

**Tabel 4.54 Use Case Scenario Lihat Riwayat SPP**

|  |  |
| --- | --- |
| Lihat riwayat SPP (PSB\_F\_50\_00) | |
| ***Actor*** | Panitia PSB |
| ***Objective*** | Dapat melihat riwayat transaksi pembayaran SPP santri |
| ***Preconditions*** | 1. Halaman detail data santri sudah ditampilkan  2. Terdapat daftar transaksi pembayaran SPP |
| ***Main Flows*** | 1. Aktor memencet sub menu SPP pada halaman detail data santri.  2. Sistem menampilkan daftar transaksi pembayaran SPP dengan data meliputi nama pembayaran, tanggal pembayaran, nominal bayar dan bukti pembayaran.  3. Sistem menampilkan tombol verifikasi pada setiap transaksi pembayaran. |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “Belum terdapat transaksi pembayaran SPP”, jika aktor mengakses sub menu SPP dan belum terdapat riwayat pembayaran. |
| ***Post Conditions*** | Riwayat pembayaran SPP santri berhasil ditampilkan. |

**Tabel 4.55 Use Case Scenario Verifikasi SPP**

|  |  |
| --- | --- |
| Verifikasi SPP (PSB\_F\_51\_00) | |
| ***Actor*** | Panitia PSB |
| ***Objective*** | Dapat melakukan verifikasi pembayaran SPP |
| ***Preconditions*** | Halaman riwayat SPP sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet tombol ceklis pada data aksi di setiap transaksi pembayaran.  2. Sistem menampilkan pemberitahuan “Bukti pembayaran sudah di cek, lanjut untuk proses verifikasi?”.  3. Aktor memencet tombol “Yes”.  4. Sistem menampilkan pemberitahuan “Berhasil, silahkan mengubah status pembayaran menjadi lunas”.  5. Aktor mengubah status pembayaran menjadi lunas. |
| ***Alternative Flows*** | 1. Sistem menampilkan pemberitahuan “Proses dibatalkan”, jika aktor memencet tombol “No”.  2. Sistem menampilkan *form* alasan penolakan pembayaran SPP, jika aktor memencet tombol silang pada data aksi. |
| ***Post Conditions*** | Verifikasi pembayaran SPP berhasil dilakukan. |

**Tabel 4.56 Use Case Scenario Lihat Data Diri**

|  |  |
| --- | --- |
| Lihat data diri (PSB\_F\_52\_00) | |
| ***Actor*** | Santri aktif |
| ***Objective*** | Dapat melihat detail informasi data diri |
| ***Preconditions*** | Halaman my profil sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet link my profil.  2. Sistem menampilkan detail informasi data diri  3. Sistem menampilkan pemberitahuan “Silahkan perbaharui data diri jika terdapat perbedaan”  4. Aktor memencet tombol “Ok”.  5. Sistem menampikan halaman data diri. |
| ***Alternative Flows*** | - |
| ***Post Conditions*** | Data diri santri berhasil ditampilkan. |

**Tabel 4.57 Use Case Scenario Lihat Tagihan SPP**

|  |  |
| --- | --- |
| Lihat tagihan SPP (PSB\_F\_53\_00) | |
| ***Actor*** | Santri aktif |
| ***Objective*** | Dapat melihat total tagihan SPP |
| ***Preconditions*** | Halaman tagihan SPP sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet sub menu SPP.  2. Sistem menampilkan detail tagihan SPP beserta nomor rekening sekolah.  3. Sistem menampilkan *form* untuk melakukan konfirmasi pembayaran. |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “Belum ada tagihan SPP”, jika aktor mengakses SPP dan status pembayaran sudah lunas. |
| ***Post Conditions*** | Detail tagihan SPP berhasil ditampilkan. |

**Tabel 4.58 Use Case Scenario Konfirmasi Pembayaran SPP**

|  |  |
| --- | --- |
| Konfirmasi pembayaran SPP (PSB\_F\_54\_00) | |
| ***Actor*** | Santri aktif |
| ***Objective*** | Dapat melakukan konfirmasi pembayaran SPP |
| ***Preconditions*** | Halaman tagihan SPP sudah ditampilkan |
| ***Main Flows*** | 1. Aktor mengisi *form* konfirmasi pembayaran SPP dengan benar dan lengkap, meliputi tanggal pembayaran dan nominal pembayaran.  2. Aktor memencet tombol browse untuk mengunggah bukti pembayaran.  3. Aktor memencet tombol konfirmasi.  4. Sistem menampilkan pemberitahuan “Lanjutkan konfirmasi pembayaran SPP?”.  5. Aktor memencet tombol “Yes”.  6. Sistem menampilkan pemberitahuan “Berhasil” |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “harap isi data dengan lengkap”, jika terdapat kolom data yang kosong.  2. Sistem menampilkan pemberitahuan “Harap lampirkan foto bukti pembayaran”, jika aktor tidak mengunggah bukti pembayaran.  3. Sistem menampilkan pemberitahuan “Harap lampirkan dalam *form*at JPEG/PDF”, jika aktor melampirkan *file* selain JPEG/PDF. |
| ***Post Conditions*** | Konfirmasi pembayaran SPP berhasil dilakukan. |

**Tabel 4.59 Use Case Scenario Lihat Riwayat Pembayaran SPP**

|  |  |
| --- | --- |
| Lihat riwayat pembayaran SPP (PSB\_F\_55\_00) | |
| ***Actor*** | Santri aktif |
| ***Objective*** | Dapat melihat riwayat konfirmasi pembayaran SPP |
| ***Preconditions*** | Halaman riwayat pembayaran sudah ditampilkan |
| ***Main Flows*** | 1. Aktor memencet sub menu riwayat pembayaran.  2. Sistem menampilkan daftar riwayat pembayaran SPP.  3. Sistem menampilkan data yaitu nama pembayaran, tanggal pembayaran, nominal pembayaran, *file* bukti pembayaran dan status dari pembayaran. |
| ***Alternative Flows*** | Sistem menampilkan pemberitahuan “Riwayat pembayaran SPP belum ditemukan”, jika aktor belum pernah melakukan proses konfirmasi pembayaran SPP. |
| ***Post Conditions*** | Riwayat pembayaran SPP 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 defenisikan pada proses sebelumnya. Adapun rangkaian tahap yang akan dilakukan yaitu: perancangan arsitektur, perancangan database, perancangan komponen dan perancangan antarmuka pengguna.

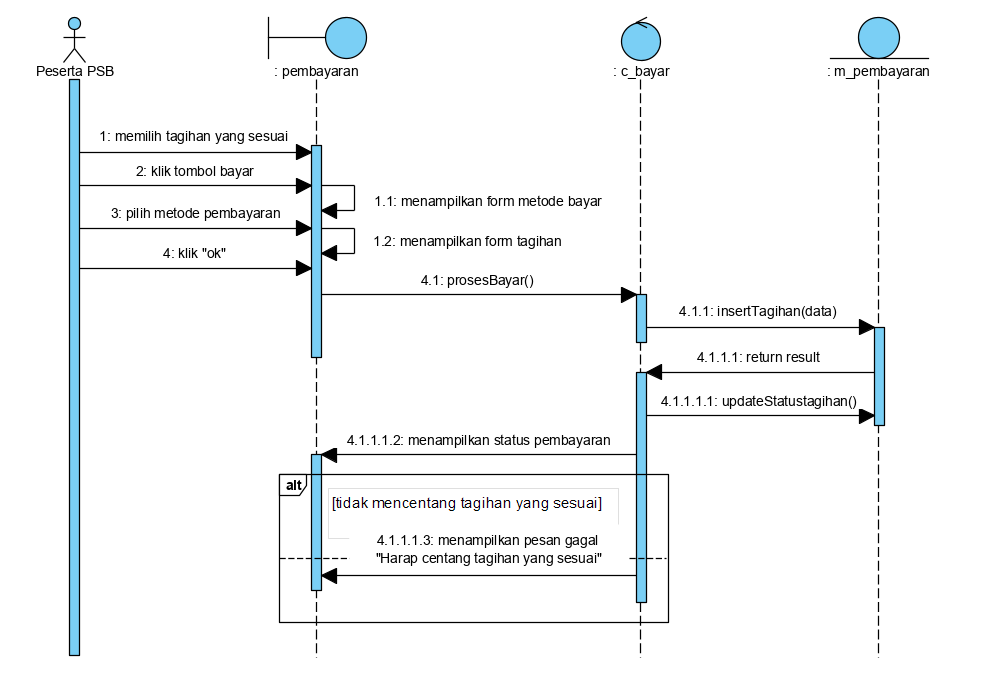
### 5.1.1 Perancangan Arsitektur

Perancangan arsitektur merupakan proses untuk mengambarkan hubungan antar komponen dalam sistem yang akan dibangun, perancangan arsitektur digmodelkan dalam pemodelan *sequence diagram* dan pemodelan *class diagram.*

#### 5.1.1.1 Perancangan Sequence Diagram

Pemodelan *sequence diagram* menggambarkan hubungan antar objek dalam sistem, pemodelan ini dirancangn 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 defenisikan, tetapi dalam dokumentasi ini hanya akan menampilkan 3 *sequence diagram* yaitu proses pembayaran, verifikasi berkas dan proses tes seleksi sebagai sampel.

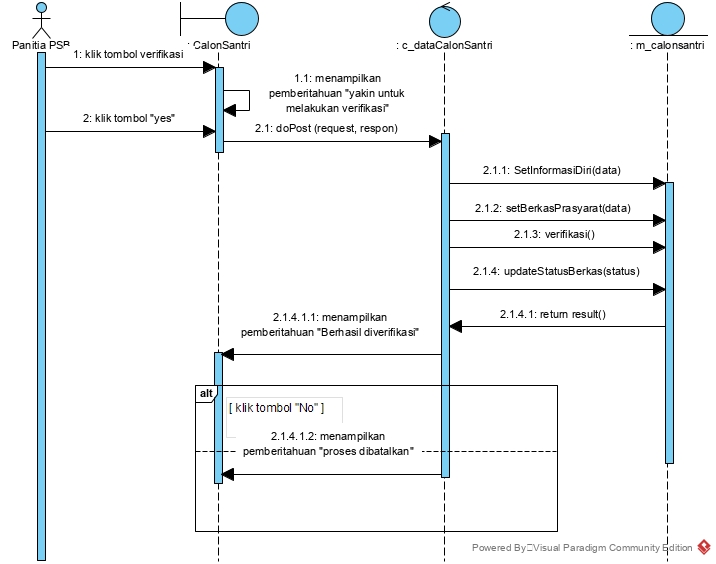
1. *Sequence Diagram* Pembayaran



**Gambar 5.1 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.

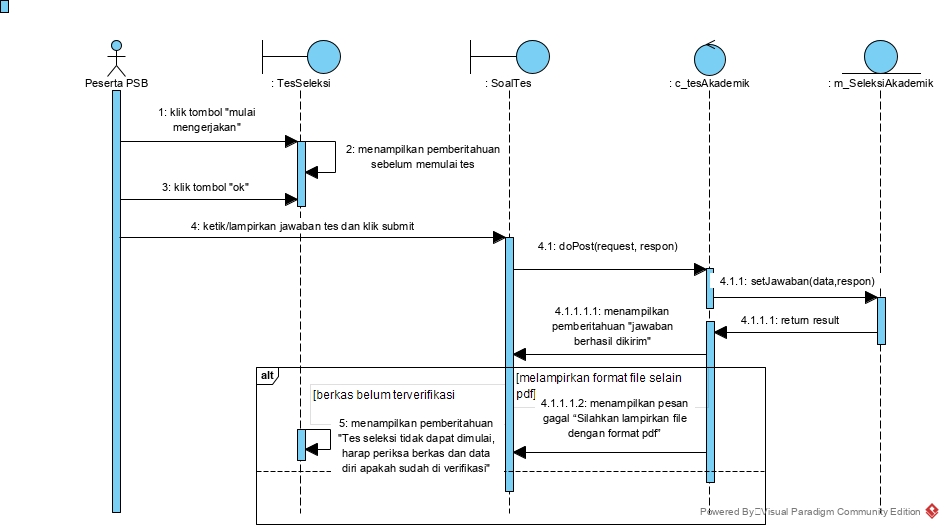
2. *Sequence Diagram* Verifikasi Berkas



**Gambar 5.2 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”.

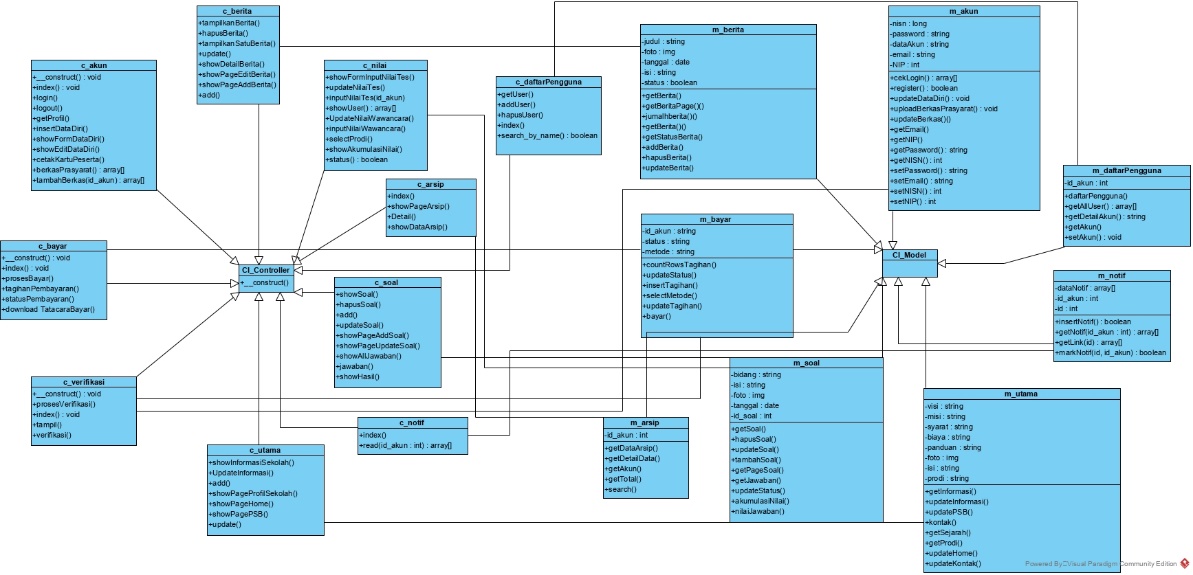
3. *Sequence Diagram* Tes seleksi



**Gambar 5.3 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”.

#### 5.1.1.2 Perancangan Class Diagram



**Gambar 5.4 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 manganut konsep MVC (Model, View Controller), dimana 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, setalah didapatkan data tersebut akan dibawa ke controller untuk ditampilkan pada *view*. Hasil dari perancangan *Class Diagram* dapat dilihat pada Gambar 5.4.

### 5.1.2 Perancangan *Database*

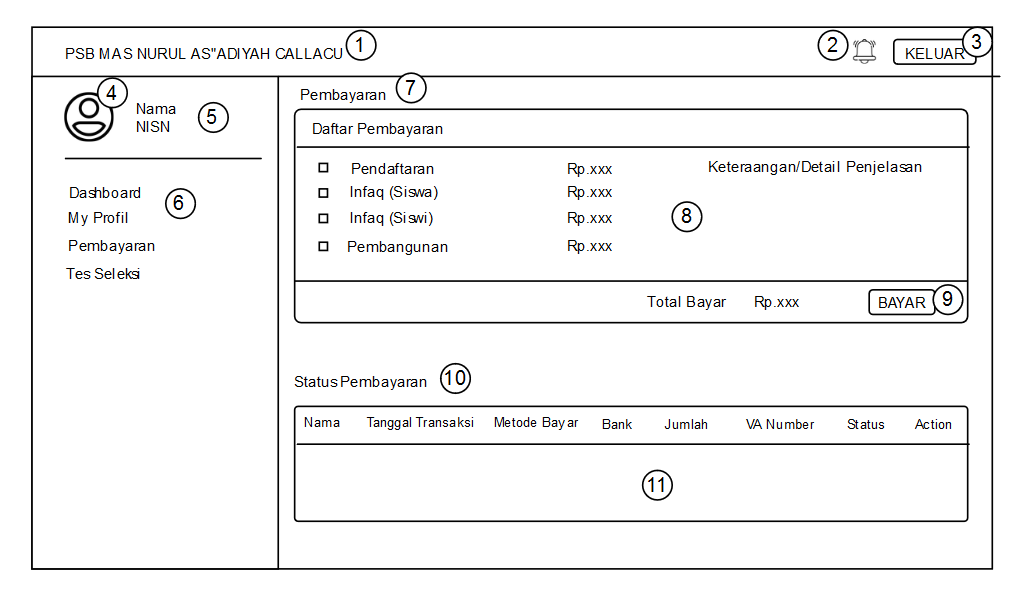
### 5.1.3 Perancangan Komponen

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 *pseudocode.* Terdapat 3 method yang akan dirancang diantaranyan nilaiJawaban(), bayar() dan tesSeleksi()

### 5.1.4 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.

#### 5.1.4.1 Perancangan Antarmuka Pembayaran

****

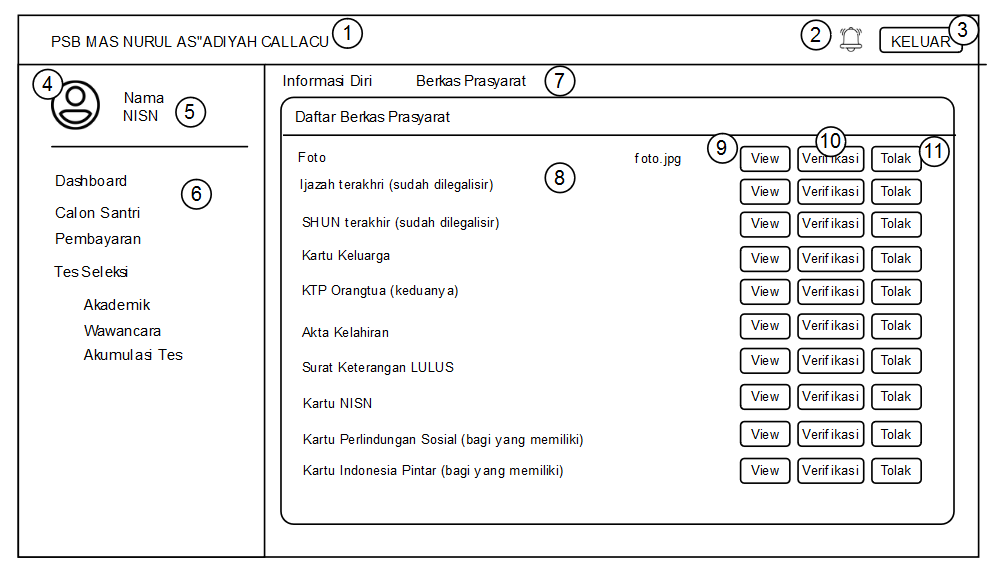
**Gambar 5.5 Perancangan Antarmuka Pembayaran**

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

**Tabel 5.1 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 |

#### 5.1.4.2 Perancangan Antarmuka Verifikasi Berkas



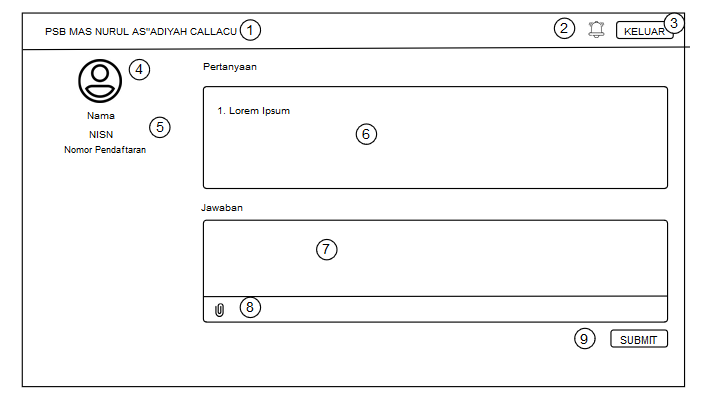
**Gambar 5.6 Perancangan Antar Muka Verifikasi Berkas**

Pada Gambar 5.5 adalah hasil rancangan antarmuka proses verifikasi berkas, detail penjelasan terkait komponen penyusun yang terdapat didalamnya dijelaskan pada Tabel 5.2.

**Tabel 5.2 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 |

#### 5.1.4.3 Perancangan Antarmuka Tes Seleksi



**Gambar 5.7 Perancangan Antarmuka Tes Seleksi**

Pada Gambar 5.6 adalah hasil rancangan antarmuka proses verifikasi berkas, detail penjelasan terkait komponen penyusun yang terdapat didalamnya dijelaskan pada Tabel 5.3.

**Tabel 5.3 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 |

**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. Retrieved from https://www.academia.edu/10793943/A\_Comparison\_Between\_Three\_SDLC\_Models\_Waterfall\_Model\_Spiral\_Model\_and\_Incremental\_Iterative\_Model

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.

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. Retrieved from https://ilmukomputer.org/2013/09/24/kupas-tuntas-css/

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*.

No, V., & Junaidi, A. (2016). Studi Perbandingan Per*form*ansi Antara MongoDB dan MySQL Dalam Lingkungan Big Data. *Prosiding Annual Research Seminar 2016*, *2*(1), 460–465.

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

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

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

Silalahi, M. (2018). Perbandingan Per*form*ansi Database Mongodb Dan Mysql Dalam Aplikasi File Multimedia Berbasis Web. *Computer Based Information System 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

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

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

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

Sutabri, T. (2012b). *Konsep Sistem* Informasi (I. Nastiti, ed.). Yogjakarta: 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.