**BAB I**

**PENDAHULUAN**

* 1. **Latar Belakang**

Perkembangan ilmu pengetahuan dan teknologi khususnya teknologi informasi yang semakin pesat di segala bidang tidak terlepas dari perkembangan kebutuhan manusia. Teknologi informasi merupakan alat untuk mempermudah, mempercepat dan merapikan pekerjaan. Selain dari itu, teknologi informasi juga memungkinkan sebuah informasi dapat diakses dalam waktu nyata tanpa dibatasi dengan ruang dan waktu. Dunia pendidikan tidak luput dari kebutuhan akan teknologi informasi. Teknologi informasi tidak hanya merupakan kebutuhan yang sangat penting bagi sebuah lembaga pendidikan, akan tetapi juga memberikan nilai lebih tinggi bagi lembaga tersebut. Hal-hal tersebut menjadikan motivasi untuk membangun sebuah sistem yang sangat dibutuhkan oleh lembaga pendidikan dalam membantu proses pengolahan data nilai siswa.

SD Inpres Bello merupakan suatu Instansi Negeri yang berdiri pada tahun 1983 dan bergerak di bidang pendidikan dasar yang beralamat di jalan. H.R. Koroh, kelurahan Bello kecamatan Maulafa, Kota Kupang. Pada tahun 2022, jumlah guru sebanyak 21 orang terdiri dari guru laki-laki berjumlah 7 orang dan guru perempuan berjumlah 14 orang. Jumlah siswa dari kelas satu sampai kelas enam sebanyak 344 orang yang terdiri dari siswa laki-laki berjumlah 176 orang, dan siswa perempuan berjumlah 168 orang.

Pemanfaatan komputer untuk pengolahan data akademik belum begitu optimal dan belum akurat informasinya, sehingga kemudahan cara kerja dan pengolahan data dalam jumlah yang besar dengan ketelitian yang tinggi belum didapati pada SD Inpres Bello. Pada saat ini kendala yang dihadapi merupakan media penyimpanan berupa buku yang diperlukan tidak hanya satu melainkan lebih dari satu. Hal yang lainnya apabila terjadi perubahan data, misalnya nilai siswa ditulis tidak benar maka harus mencoret data yang lama dengan data yang baru, pencoretan itu membuat media penyimpanan berupa buku menjadi kurang rapi dan tidak bersih. Meskipun sebagian sudah mengunakan media penyimpanan menjadi *file* tapi masih juga mengalami kendala jika adanya pencarian data tentang sebuah data seperti mencari data nilai satu siswa. Sering kali proses pencatatan, pencarian data, memerlukan waktu yang tak sedikit apalagi jumlah data-data di lembaga pendidikan itu bisa dibilang tidak sedikit. Untuk mengatasi permasalahan yang dihadapi oleh SD Inpres Bello dalam hal mengelola data-data akademik yang diperlukan sistem yang komputerisasi yaitu sistem yang berbasis *web* untuk mempermudah dalam pengolahan data nilai, data mata pelajaran, data jadwal mata pelajaran, data siswa, data guru, data kelas di SD Inpres Bello, sehingga proses pencarian data, pengarsipan dan pembuatan laporan akan lebih cepat, singkat, tepat dan akurat. Proses pencarian data dan nilai siswa membutuhkan waktu yang lama karena harus mencari dan membuka berkas-berkas terlebih dahulu. Terdapat beberapa permasalahan mengenai pengolahan data akademik pada SD Inpres Bello data yang dihasilkan kurang akurat karena masih terdapat data yang berulang atau tidak tercatat, dikarenakan masih menggunakan media kertas yang kurang menunjang untuk jangka waktu yang panjang karena jumlah data guru dan siswa yang banyak maka data yang ditampung akan semakin besar, sehingga akan memperlambat kinerja sistem untuk menyajikan informasi secara cepat dan tepat. Apabila diterapkan sistem informasi akademik berbasis *web*, maka semua data dapat tersimpan dengan rapi, *integrity* terjamin, pengolahan data dan informasi dapat di lakukan secara cepat, tepat dan akurat.

1

Berdasarkan analisis tersebut maka akan dibangun sebuah “Sistem Informasi Akademik Berbasis *Web* pada SD Inpres Bello” yang diharapkan dapat mempermudah penyampaian informasi khususnya data mata pelajaran, data guru, data siswa, data nilai, absensi, jadwal dan informasi – informasi mengenai gambaran umum SD Inpres Bello yang dapat membantu orang tua siswa agar mengetahui perkembangan prestasi anaknya secara bertahap sehingga jika prestasi belajar anak menurun maka orang tua dapat segera memotivasi dan membimbing anaknya untuk meningkatkan prestasi belajar siswa.

**1.2. Rumusan Masalah**

Berdasarkan uraian diatas, maka perumusan masalah dalam penelitian ini adalah bagaimana membangun Sistem Informasi Akademik Berbasis *Web* pada SD Inpres Bello.

* 1. **Tujuan Penelitian**

Adapun tujuan dalam penelitian ini adalah membangun Sistem Informasi Akademik Berbasis *Web* Pada SD Inpres Bello, sehingga dapat memperluas jangkauan siswa untuk mengakses semua informasi yang berhubungan dengan proses belajar mengajar.

**1.4. Manfaat Penelitian**

Adapun manfaat yang akan diperoleh dari penelitian yang dilakukan pada SD Inpres Bello diantaranya adalah sebagai berikut:

1. Membantu pihak sekolah dalam mengelolah data akademik siswa.
2. Mempermudah siswa dan orangtua untuk memperoleh informasi secara cepat mengenai data kegiatan proses belajar mengajar.

**1.5**. **Ruang Lingkup Penelitian**

Agar pembahasan dalam penelitan ini tidak melebar maka dibatasi ruang lingkup penelitian yaitu:

1. Sistem informasi ini hanya akan diimplementasikan di SD Inpres Bello.
2. Sistem informasi berbasis *web* ini hanya akan memberikan informasi yang berhubungan dengan proses belajar mengajar seperti jadwal pelajaran, data nilai, data siswa, data mata pelajaran, data guru, data kelas.

**BAB II**

**TINJAUAN PUSTAKA**

**2.1. Tinjauan Umum SD Inpres Bello**

SD Inpres Bello adalah salah satu bentuk pendidikan formal yang ada di Kota Kupang berdiri sejak tahun 1983 di jalan. H.R. Koroh, kelurahan Bello kecamatan Maulafa, Kota Kupang. Dengan luas tanah mencapai 5.170 m², memiliki 344 siswa yang terdiri dari 176 siswa laki-laki dan 168 siswi perempuan.

* + 1. **Visi dan Misi**

Untuk meningkatkan mutu pendidikan SD Inpres Bello maka disusunlah visi dan misi sebagai berikut:

1. **Visi**

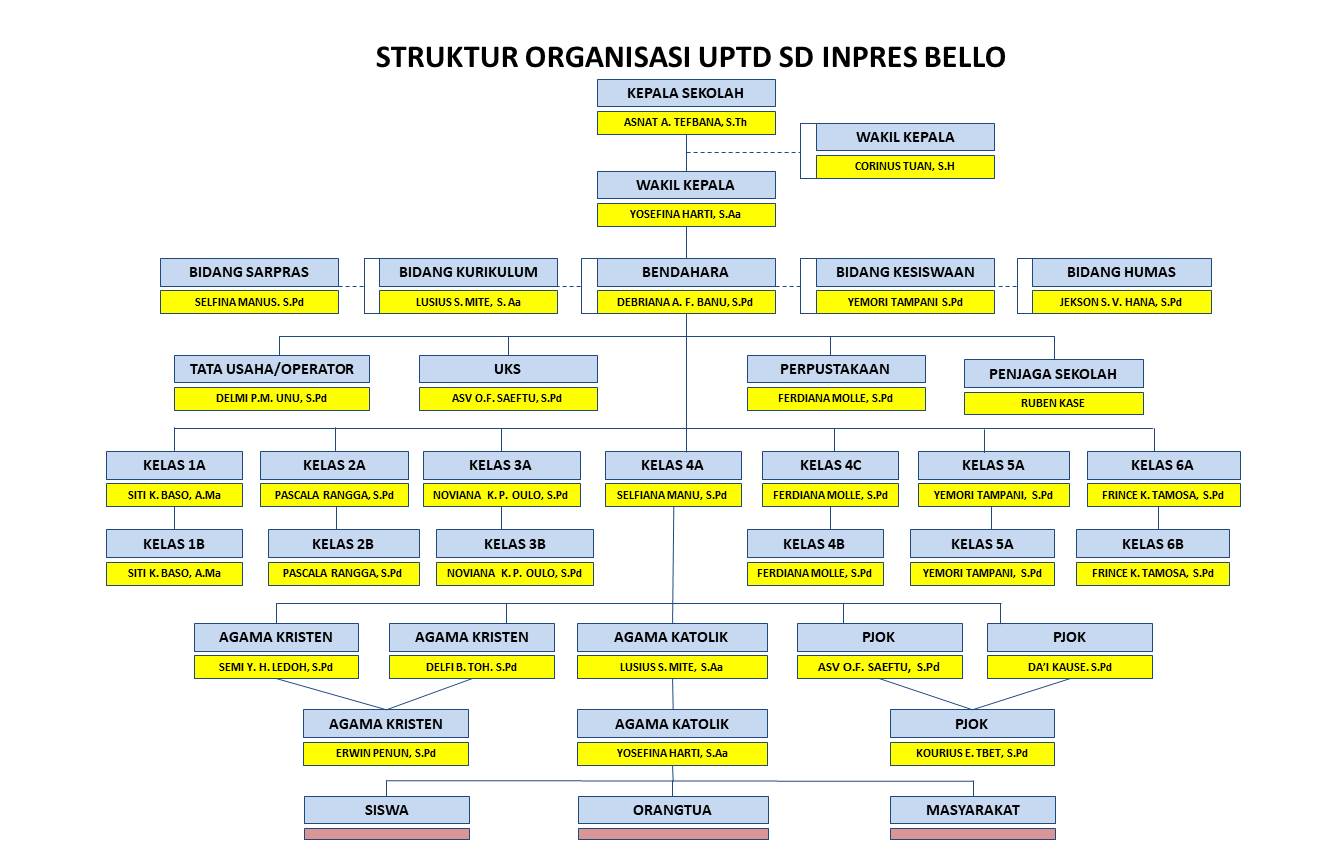
Luhur dalam pekerti, prima, dalam prestasi, santun dalam berperilaku.

1. **Misi**
2. Memberikan dasar-dasar keimanan dan ketakwaan terhadap Tuhan Yang Maha Esa.
3. Meningkatkan disiplin belajar mengajar.
4. Menciptakan suasana belajar yang aktif, kreatif, efektif, dan menyenangkan.
5. Meningkatkan kegiatan ekstrakurikuler.
6. Pembinaan prestasi siswa dalam ilmu pengetahuan, keagamaan, seni dan olahraga.
7. Memupuk atau menumbuhkan kembangkan rasa cinta terhadap sesama manusia dan lingkungannya.
8. Membiasakan siswa hidup bersih.
9. Mengembangkan nilai-nilai budi pekerti luhur.
10. Meningkatkan profesionalisme guru atau personil.
11. Membina hubungan sekolah dengan komite sekolah dan masyarakat sekitar sekolah.

* + 1. **Struktur Organisasi**

Struktur organisasi adalah suatu kerangka yang mengenai tugas, pekerjaan, hubungan, wewenang dan tanggung jawab dari orang-orang atau yang bekerja sama dalam organisasi tersebut.

3

****

Gambar 1. Struktur organisasi sekolah

* + 1. ***Job* Deskripsi**

Berikut ini merupakan *Job* deskripsi dari SD Inpres Bello sebagai berikut:

1. **Kepala Sekolah**
2. Kepala sekolah berfungsi sebagai pendidik, mengelola administrator (pendorong), pengayom, dan pembimbing.
3. Kepala sekolah mempunyai tugas menyusun rencana dan program sekolah yaitu membina kesiswaan, pembelajaran, dan ketenagaan administrsi sekolah serta membina dan melaksanakan kerja sama atau hubungan dengan masyarakat
4. **Wakil kepala sekolah**

Tugas dan peranan wakil kepala sekolah yaitu membantu kepala sekolah dalam menjalankan roda kepemimpinan melalui kegiatan-kegitan sebagai berikut:

1. Menyusun perencanaan, membuat program kegiatan dan program pelaksanaan.
2. Pengorganisasian
3. Pengarahan, ketenagaan, pengawasan, pelaksanaan, identifikasi, pengumpulan data dan penyusunan laporan.
4. **Komite sekolah**
5. Mengkordinasikan rapat komite sekolah dan rapat semester.
6. Mengarsip dan mendokumentasikan.
7. Mengkordinasikan kegiatan lokal.
8. **Perpustakaan**

Tugas dan kewajiban perpustakaan yaitu mengelolah perpustakaan seperti melakukanperencanaan, pengadaan buku, inventarisasi buku, membuat katalog penyusunan buku, mengatur peminjaman buku dan pemeliharaan buku.

1. **Tata Usaha (TU)**

Adapun tugas dari Tata Usaha adalah:

1. Penyusunan program kerja tata usaha.
2. Pengurutan administrasi ketenagaan dan siswa.
3. Penyusunan dan penyajian dan statistik sekolah.
4. Pengolahan keuangan sekolah.
5. **Wali Kelas**

Tugas dan tanggung jawab wali kelas adalah membantu kepala sekolah dalam kegiatan-kegiatan berikut:

1. Mengelola kelas.
2. Mengetahui identitas dan kepribadian anak didik.
3. Merekapitulasi kehadiran siswa.
4. Pengisian pendaftaran nilai siswa.

e. Membuat catatan khusus tentang siswa.

1. **Guru**

Guru bertugas membuat perangkat program pengajaran, melaksanakan kegiatan pembelajaran, melaksanakan kegiatan penilaian proses belajar seperti ulangan harian, ulangan mingguan, ulangan bulanan, ulangan semester, dan ujian akhir.

1. **Siswa**

Tugas dan tanggung jawab siswa yaitu mengikuti proses belajar, mengerjakan tugas, menjaga kebersihan sekolah, mentaati guru dan tertib sekolah.

1. **Masyarakat**

Tugas dan tanggung jawab masyarakat dalam dunia pendidikan yaitu menjadi fasilitator dalam menunjang pelaksanaan pendidikan nasioanal, ikut serta dalam menyelenggarakan pendidikan swasta, membantu pengadaan tenaga, sarana dan prasarana serta membantu mengembangkan profesi baik secara langsung maupun tidak langsung.

**2.2 Tinjauan Umum Sistem Informasi**

**2.2.1. Sistem**

Sistem merupakan sebuah kumpulan objek-objek yang saling berelasi dan berinteraksi, serta berhubungan antara objek bisa dilihat satu kesatuan yang dirancang untuk mencapai satu tujuan yang ditetapkan (Tohari, 2014).

Istilah sistem secara umum dapat didefinisikan sebagai kumpulan hal atau elemen yang saling bekerja sama atau dihubungkan dengan cara- cara tertentu sehingga membentuk satu kesatuan untuk melaksana suatu fungsi guna mencapai suatu tujuan. Elemen-elemen sistem secara umum, suatu sistem mempunyai tiga elemen atau komponen dasar yaitu:

1. Blok masukan (*input block*), Input memiliki data yang masuk ke dalam sistem informasi, juga metode-metode untuk menangkap data yang dimasukan.
2. Blok model (*output block)*, blok ini terdiri dari kombinasi prosedur logika dan model matematika yang akan memanipulasi data inputan dan data yang tersimpan di basis data.
3. Blok keluaran (*output block*), produk dari sistem informasi adalah keluaran yang merupakan informasi berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.
4. Blok teknologi (*technology block*), blok teknologi digunakan untuk menerima input, menyimpan, mengakses data, menghasilkan dan mengirimkan keluaran dari sistem secara keseluruhan. Teknologi terdiri dari tiga bagian utama, yaitu teknisi (*brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*).
5. Basis data (*database block)*, basis data merupakan kumpulan data yang saling berhubungan satu dengan lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak (*software*) untuk memanipulasinya.

**2.2.2 Informasi**

Menurut Kristanto (2003), informasi merupakan kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima.

Menurut Jogiyanto (2004), informasi adalah hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penggunan yang mengambarkan suatu kejadian-kejadian (*event*) yang nyata (*fact*) yang digunakan untuk pengambilan keputusan kualitas informasi tergantung dari hal yaitu:

1. Akuratn

Informasi harus bebas dari kesalahan-kesalahan, tidak menyesatkan dan harus jelas mencerminkan maksud dan makna yang terkandung dari data pendukungnya.

1. Tepat waktu

Informasi yang dihasilkan dari suatu proses pengolahan data, datangnya tidak boleh terlambat (usang). Informasi yang terlambat tidak akan mempunyai nilai yang baik, apabila digunakan sebagai dasar dalam pengambilan keputusan dapat menimbulkan kesalahan dalam tindakan yang diambil.

1. Relevan

Informasi harus mempunyai manfaat bagi pihak yang menerimanya.

**2.2.3 Sistem Informasi**

Menurut Hutahaean (2014), sistem informasi merupakan suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi bersifat material dan kegiatan strategi dari suatu organisasi serta menyediakan pihak luar terbentuk dengan laporan-laporan yang diperlukan. Menurut Mulyanto (2009), sistem informasi sebagai suatu komponen yang terdiri dari manusia, teknologi informasi, dan prosedur kerja yang memproses, menyimpan menganalisis dan menyebarkan informasi untuk mencapai suatu tujuan.

1. Untuk meningkatkan aksebilitas data yang ada secara efektif dan efisien kepada pengguna, tanpa dengan perantara sistem informasi.
2. Memperbaiki produktifitas aplikasi pengembangan dan pemeliharaan sistem.
3. Menjamin tersedianya kualitas dan ketrampilan dalam memanfaatkan sistem informasi secara kritis.
4. Mengidentifikasi kebutuhan mengenai keterampilan pendukung sistem informasi.
5. Mengantisipasi dan memahami akan konsekuensi ekonomi.

**2.2.4 Akademik**

Menurut Pacitan (2012), sistem informasi akademik adalah suatu sistem yang dirancang untuk keperluan pengolahan data akademik dengan penerapan teknologi komputer baik *hardware* maupun *software* yang bertujuan memberikan informasi terhadap para siswa, orang tua siswa dan masyarakat tentang sekolah, fasilitas sekolah, data siswa, data nilai siswa. Sistem Informasi Akademik Sekolah merupakan sebuah Sistem Informasi Akademik berisi kumpulan dari interaksi sistem informasi yang menyediakan informasi kebutuhan akademik sekolah.

**2.3*****Web***

Menurut Yuhefizar (2012), *web* adalah salah satu layanan yang didapat oleh pemakai komputer yang terhubumg ke internet. Web ini menyediakan informasi bagi pemakai komputer yang terhubung ke internet dari sekedar informasi “sampah” atau informasi yang tidak berguna sama sekali sampai informasi yang serius; dari informasi yang gratisan sampai informasi yang komersial. *Website* atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya itu baik yang bersifat statis maupun dinamis yang dihubungkan dengan jaringan-jaringan secara umum situs web mempunyai fungsi sebagai:

1. Fungsi komunikasi

Situs *web* yang mempunyai fungsi komunikasi pada umumnya adalah situs *web* dinamis. Karena dibuat menggunakan pemrograman *web* (*server side*) maka dilengkapi fasilitas yang memberikan fungsi-fungsi komunikasi, seperti *web* *mail,form contact*, *chatting form*, dan yang lainnya.

1. Fungsi informasi

Situs *web* yang memiliki fungsi informasi pada umumnya lebih menekankan pada kualitas bagian kontennya, karena tujuan situs tersebut adalah menyampaikan isisnya. Situs ini sebaiknya berisi teks dan grafik yang dapat di-*download* dengan cepat.

1. Fungsi *entertaiment*

Situs *web* juga memiliki fungsi *entertainment* atau hiburan. Bila situs *web* berfungsi sebagai sarana hiburan maka pengunaan animasi gambar dan elemen bergerak dapat meningkatkan mutu presentasi desainnya, meski tetap harus mempertimbangkan kecepatan *downloadnya.*

1. Fungsi transaksi

Situs *web* dapat dijadikan sarana transaksi bisnis, baik barang, jasa, atau lainnya. Situs *web* ini menghubungkan perusahaan, konsumen, dan komunitas tertentu melalui transaksi elektronik. Pembayarannya bisa menggunakan kartu kredit, transfer, atau dengan membayar secara langsung.

**2.4. PHP (*Personal Home Page*)**

Menurut Arief (2011), PHP adalah bahasa *server-side-scripting* yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis , karena PHP merupakan *server side-scripting* maka sintaks dan perinta-perintah PHP akan dieksekusi di*server* kemudian ke *browser* dengan format HTML dengan demikian kode program yang ditulis dalam PHP tidak akan terlihat oleh *user* sehingga keamanan halaman *web* lebih terjamin. PHP dirancang untuk membuat halaman *web* yang dinamis , yaitu halaman *web* yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman *web.*

**2.5. MySQL**

Pengertian MySQL MySQL adalah sebuah sistem database relasional, sehingga dapat mengkelompokkan informasi kedalam tabel-tabel, atau grup-grup informasi yang berkaitan. Setiap tabel memuat bidang-bidang yang terpisah, yang mempresentasikan setiap bit informasi. Seperti membuat berbagai tipe data seperti teks, angka, tanggal, dan lainya (Sianipar, 2015). MySQL banyak dipakai untuk kepentingan penanganan *database* karena selain handal juga bersifat open-source. Konsekuensi dari *open-source*, perangkat lunak ini dapat dipakai oleh siapa saja tanpa membayar dan source codenya bisa diunduh oleh siapa saja. Sejumlah aktifitas yang terkait dengan data yang didukung oleh perangkat lunak tersebut:

1. Menyimpan data ke dalam tabel.

2. Menghapus data dalam tabel.

3. Mengubah data dalam tabel.

4. Mengambil data yang tersimpan dalam tabel.

5. Memungkinkan untuk memilih data tertentu yang diambil.

6. Memungkinkan untuk melakukan pengaturan hak akses terhadap data.

**2.6 Xampp**

Menurut Madcoms (2016), Xampp adalah sebuah paket kumpulan *software* yang terdiri dari Apache*,* MySQL, PhpMyAdmin*,* PHP*,* Perl, Fezilladan lain-lain. *Xampp* berfungsi untuk memudahkan instalasi lingkungan PHP, dimana biasanya lingkungan pengembangan *web* memerlukan PHP*, Apache,* SQL, dan PhpMyAdmin serta *software-software* yang terkait dengan pengembangan *web*. Xampp merupakan *web server apache* yang di dalamnya ternama *serverMySQL* 50 yang didukung dengan bahasa pemrograman PHPuntuk membuat *website* yang dinamis. Xampp mendukung dua sistem operasi yaitu *windows* dan L*inux.* Untuk L*inux* proses penginstalanya menggunakan *command line* sedangkan untuk *windows* dalam proses penginstalanya menggunakan *interface* grafis sehingga lebih mudah dalam pengunaan xampp di windows dibanding dengan L*inux*.

* 1. **Penelitian Terdahulu**

Dengen dan Marisa (2016) melakukan penelitian dengan judul Sistem Informasi Akademik Berbasis *Web* SMP Negeri 4 Samarinda. Tujuan penelitian ini adalah membangun sistem informasi akademik berbasis *web* yang dapat digunakan sebagai salah satu fasilitas di SMP Negeri 4 Samarinda untuk penyajian informasi kepada siswa mengenai data nilai dan data absensi serta menyajikan informasi kepada *user* mengenai daftar pegawai, daftar kelas, serta daftar jadwal mata pelajaran. Metode yang digunakan adalah studi lapangan, studi kepustakaan, internet, analisis data, desain. Penelitian ini menghasilkan informasi mengenai daftar data pegawai, data pengajar, data kelas, data sarana dan prasarana yang dimiliki serta data jadwal mata pelajaran pada tiap-tiap kelas, kekurangan aplikasi ini hanya kalangan tertentu yang memiliki hak akses terhadap sistem ini yaitu siswa, pengajar dan administrator.

Erik (2014) melakukan penelitian dengan judul Perancangan Sistem Informasi Akademik pada Sekolah Dasar Negeri Sukajadi 9 Bandung. Tujuan penelitian adalah perancangan sistem akademik yang sudah terkomputerisasi, proses pencarian data siswa, data guru, data kelas, data nilai lebih efektif. Metode yang digunakan dengan menggunakan metode deskriptis da metode aktif dimana gambaran tentang penelitian selanjutnya akan di ambil tindakan untuk membuat suatu sistem. Penelitian ini menghasilkan proses pencarian data siswa.

Rizki (2016) melakukan penelitian dengan judul Sistem Informasi Akademik Berbasis *Web* di SMP Negeri 2 Kecamatan Gaung Anak Serka. Tujuan penelitian ini membangun sistem informasi akademik berbasi *web* yang dapat untuk memperbaiki kinerja sistem pengolah data akademik sehingga menjadi kemudahan untuk sarana pengolahan data akademik dapat diperoleh dengan efektif dan efisien. Metode pengambilan data yang digunakan penulis yaitu melakukan observasi, wawancara, dan studi pustaka. Penelitian ini menghasilkan laporan siswa secara *online* yang berupa laporan nilai siswa yang bersangkutan, jadwal pelajaran, dan data pengajar, sehingga membantu kecepatan dan kualitas dalam menyampaikan informasi.

**BAB III**

**METODE PENELITIAN**

**3.1. Lokasi Waktu Rencana Penelitian**

Waktu penelitian dimulai pada bulan Mei 2021, sedangkan untuk lokasi tempat penelitian di SD inpres Bello, yang beralamat di jalan H.R Koroh Km. 9, RT. 02/RW. 01, Kelurahan. Bello, Kecamatan. Maulafa, Kota Kupang.

**3.2 Bahan Penelitian dan Alat Penelitian**

**3.2.1 Bahan Penelitian**

Teknik pengumpulan data merupakan salah satu faktor keberhasilan dalam suatu penelitian/penulisan. Dalam menyelesaikan penulisan tugas akhir ini, penulis mengambil data dengan cara memakai metode pengumpulan data. Jenis data yang diperoleh biasanya bisa berupa data primer maupun data sekunder. Dimana data primer merupakan data yang diambil atau diperoleh dari responden melalui kuesioner atau data hasil wawancara atau secara langsung, sedangkan data sekunder merupakan data yang sudah ada dari sumbernya. Data primer diambil dari pembagian kuesioner pada SD inpres Bello.

Proses pengumpulan data dalam membantu menyelesaikan penelitian/penulisan ini diambil dari data yang sesuai, dan penulis melakukan penulisan berdasarkan penelitian tentunya dengan langkah-langkah sebagai berikut:

* 1. Observasi

Observasi merupakan teknik pengumpulan data atau pengamatan yang dilakukan secara langsung demi untuk mendapatkan informasi yang lengkap. Data yang dihasilkan dari pengamatan atau penelitian ini berdasarkan fakta yang terjadi di SD Inpres.

* 1. Wawancara

Wawancara atau *interview* merupakan kegiatan yang dilakukan dengan proses tanya jawab melalui percakapan secara langsung maupun tidak langsung. Wawancara di lakukan secara langsung dengan guru SD Inpres Bello.

c. Studi pustaka

Penelitian dengan memanfaatkan obyek penelitian serta informasi yang diperoleh dari sumber-sumber bacaan yang ada diperpustakaan kampus, karya ilmiah, internet, serta sumber-sumber lain yang dapat menunjang penelitian/penulisan ini.

**3.2.2. Alat Penelitian**

Pada penelitian ini, penulis mengunakan peralatan penelitian berupa *hardware* dan *software* dalam pengerjaannya. Peralatan tersebut antara lain:

1. *Hardware* (perangkat keras)

Perangkat keras yang digunakan untuk membangun sebuah sistem informasi pengolahan data nilai siswa sebagai berikut:

1. Laptop Acer *Aspire 3* (A314-35) dengan processor Intel® Celeron® N5100 @ 1.10GHz (4CPlus), ~1.1GHz

10

1. RAM 4,00 GB.
2. Koneksi internet minimal 64 kbps.
3. *Software* (perangkat lunak)

Perangkat lunak yang digunakan adalah:

1. Sistem operasi *Windows* 10 64 Bit *operation system.*
2. *Chrome* untuk menjalankan *web.*
3. *Boostrap* mendesain tampilan *web.*
4. *Sublime Text.*
5. Xampp V.3.2.4.

**3.3 Prosedur Penelitian**

Dalam tahapan penelitian ini terdapat beberapa tahap yang dilakukan peneliti dalam melakukan pengumpulan data hingga selesai. Penelitian ini digunakan sebagai pedoman dalam melaksanakan penelitian ini agar hasil yang dicapai tujuan yang telah ditentukan sebelumnya. Berikut ini *flowchart* metode penelitian:



Gambar 2. *Flowchart* Penelitian

Berdasarkan alur *flowchart* pada Gambar 2 maka dapat dijelaskan sebagai berikut:

1. Studi literatur

Dilakukan dengan mencari literatur pendukung penelitian yang mampu memberikan informasi memadai untuk menyelesaikan penelitian ini serta dapat membantu mempertegas teori-teori yang ada untuk memadukan informasi yang relevan dengan topik atau masalah yang sedang atau akan diteliti.

1. Pengumpulan data

Pada tahap ini dengan melakukan pengumpulan data informasi yang berkaitan dengan penelitian yaitu pengumpulan data mata pelajaran, data jadwal mata pelajaran, data guru, data siswa, dan data kelas.

1. Analisis Data

Pada tahap ini melakukan analisis kebutuhan menjadi sebuah informasi terkait pengolahan data nilai siswa sehingga karakteristik data tersebut bisa dipahami dan bermanfaat untuk solusi permasalahan terutama masalah yang berkaitan dengan penelitian ini.

1. Perancangan program

Pada tahap ini dibuat berdasarkan hasil analisa kebutuhan yang akan dilakukan dan digunakan untuk membangun aplikasi pengolahan data akademik berbasis *website* pada SD Inpres Bello.

1. *Coding*

*Coding* dimulai ketika beberapa bentuk desain perancangan yang telah dilakukan dan spesifikasi dari modul untuk dikembangkan tersedia. dengan desain, modul ditugaskan untuk *coding.*

1. Implementasi program

Pada tahap ini perancangan yang telah dibuat diimplementasikan ke dalam program sehingga sesuai dengan perancangan.

1. Pengujian program

Pada tahap ini dilakukan pengujian kembali pada aplikasi yang telah dijalankan. Jika pengujian ini hasilnya benar maka akan lanjut, jika hasilnya salah maka akan kembali ke analisa.

1. Pemeliharaan Sistem

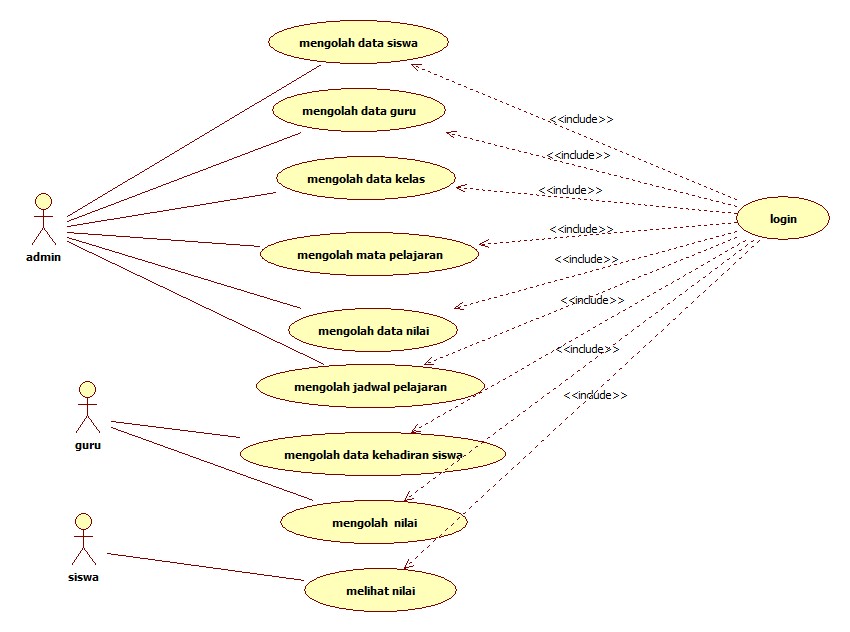
Pemeliharaan sistem adalah suatu kombinasi dari berbagai tindakan yang dilakukan untuk menjaga suatu sistem dalam atau memperbaikinya sampai suatu kondisi yang bisa diterima. Adapun beberapa hal yang harus dipelihara oleh sistem:

1. Sistem mengalami perubahan-perubahan karena permintaan baru dari pemakai sistem.
2. Sistem mengalami perubahan karena perubahan pada pengolahan data nilai siswa dan jadwal mata pelajaran

**3.4. Analisis dan Perancangan Sistem**

1. *Use case* *diagram*

*Use case diagram* merupakan model diagram yang digunakan untuk menggambarkan *requirement* fungsional yang diharapkan dari sebuah sistem. *Use case* *diagram* menekankan pada “siapa” melakukan “apa” dalam lingkungan sistem perangkat lunak yang akan dibangun. *Use case diagram* sebenarnya terdiri dari dua bagian besar yaitu *Use case* *diagram* dan *Use case description*. Untuk lebih jelas dapat dilihat pada Gambar 3 berikut:

`

Gambar 3. *Use case* *diagram* Aplikasi Pengolahan data nilai siswa

SD Inpres Bello

1. Spesifikasi *Use Case*

Spesifikasi *Use case* ini merupakan spesifikasi yang menjelaskan bagian-bagian *brief description* dari setiap *Use case* dan *basic flow* dari *actor* atau pengguna untuk lebih jelasnya dapat dilihat pada tabel-tabel berikut.

Tabel 1. Spesifikasi *Use case* : *login*

|  |  |
| --- | --- |
| *Use case name* | *Login.* |
| *Brief description* | *Use case* ini digunakan *actor* yang ingin mengakses halaman *user* |
| *Actor* | Admin, guru, siswa |
| *Basic flow* | 1. *Use case* ini dimulai ketika *actor* memilih untuk melakukan *login.* 2. Sistem meminta *actor* untuk mengisi *username* dan *password* 3. *Actor* mengisi *username* dan *password*. 4. *Actor* memberitahu kepada sistem bahwa *username* dan *password* telah diisi. 5. Sistem memeriksa *username* dan *password* yang telah diisi. E-1:*username* dan  *password* salah. E-2: *username* dan *password* tidak lengkap. 6. Sistem memberikan akses ke *actor.* 7. *Use case* selesai. |
| *Alternative flow* | *None* |
| *Error flow* | E-1: *username* dan *password* salah   1. Sistem memberitahu *actor* bahwa *usernam*e dan *password* salah. 2. Kembali ke b*asic flow* langkah ke-tiga.   E-2: *username* dan *password* tidak lengkap   1. Sistem memberitahu *actor* bahwa *usernam*e dan *password* tidak lengkap. 2. Kembali ke b*asic flow* langkah ke-tiga. |
| *Pre conditions* | *None* |
| *Post condition* | *Actor* memasuki halaman *user* dan dapat menggunakan fungsi-fungsi pada halaman user. |

Tabel 2. Spesifikasi *Use case* : admin mengolah data siswa

|  |  |
| --- | --- |
| *Use case name* | Admin mengolah data siswa |
| *Brief description* | *Use case* ini digunakan *actor* untuk melakukan pengolahan data siswa. u*ser* dapat menambahkan, mengubah, mencari dan menghapus data. |
| *Actor* | Admin |
| *Basic flow* | 1. *Use case* ini dimulai ketika *actor* memilih untuk mengolah data siswa*.* 2. Sistem memberikan pilihan kepada *actor* untuk menambah, mengubah, menghapus atau mencari data siswa. 3. *Actor* memilih menambah data siswa.   A-1 : *actor* memilih mengubah data siswa.  A-2 : *actor* memilih menghapus data siswa.  A-3 : *actor* memilih mencari data siswa.   1. Sistem menampilkan *form* data siswa. 2. *Actor* mengisi data siswa. 3. *Actor*  memberitahukan kepada sistem bahwa data sudah diisi. 4. Sistem memeriksa data siswa.E-1: Data siswa tidak lengkap. 5. Sistem menyimpan informasi ke basis data. 6. *Use case* selesai. |
| *Alternative flow* | A-1 : *actor* memilih mengubah data siswa.   1. Sistem menampilkan data siswa yang sudah terdaftar. 2. *Actor* memilih data siswa yang ingin diubah. 3. Sistem menampilkan data siswa yang sudah dipilih. 4. *Actor* memasukan data baru. 5. Kembali ke *basic flow* langkah ke-enam.   A-2 : *actor* memilih menghapus data siswa.   1. Sistem menampilkan data siswa yang sudah terdaftar. 2. *Actor* memilih data siswa yang ingin dihapus. 3. Sistem memberikan pesan bahwa data berhasil dihapus. 4. Kembali ke *basic flow* langkah ke-sembilan   A-3 : *actor* memilih mencari data siswa.   1. Sistem meminta *actor* untuk menginputkan kata kunci. 2. *Actor* menginput kata kunci pencarian. 3. Sistem melakukan pencarian.   E-2: Data yang dicari tidak ditemukan.   1. Sistem menampilkan data siswa yang dicari di view siswa. 2. Kembali ke *basic flow* langkah ke-sembilan |
| *Error flow* | E-1: Data siswa tidak lengkap.   1. Sistem memberitahu *actor* bahwa data siswa tidak lengkap. 2. Kembali ke *basic flow* langkah ke-lima.   E-2: Data yang dicari tidak ditemukan.   1. Sistem mengosongkan *view* siswa. 2. Kembali ke *alternative flow* (A-3) langkah ke-dua. |
| *Pre conditions* | 1. *Use case login* sudah dilakukan 2. *Actor* sudah memasuki sistem |
| *Post condition* | Data siswa ter-*update* |

Tabel 2. Spesifikasi *Use case* : admin mengolah data siswa (lanjutan)

Tabel 3. Spesifikasi *Use case* : admin mengolah data guru

|  |  |
| --- | --- |
| *Use case name* | Admin mengolah data guru |
| *Brief description* | *Use case* ini digunakan *actor* untuk melakukan pengolahan data guru. *User* dapat menambahkan, mengubah, mencari dan menghapus data. |
| *Actor* | Admin |
| *Basic flow* | 1. *Use case* ini dimulai ketika *actor* memilih untuk mengolah data guru*.* 2. Sistem memberikan pilihan kepada *actor* untuk menambah,mengubah,menghapus atau mencari data guru. 3. *Actor* memilih menambah data guru.   A-1 : *actor* memilih mengubah data guru.  A-2 : *actor* memilih menghapus data guru.  A-3 : *actor* memilih mencari data guru.   1. Sistem menampilkan *form* data guru. 2. *Actor* mengisi data guru. 3. *Actor*  memberitahukan kepada sistem bahwa data sudah diisi. 4. Sistem memeriksa data guru.E-1: Data guru tidak lengkap. 5. Sistem menyimpan informasi ke basis data. 6. *Use Case* selesai. |
| *Alternative flow* | A-1 : *actor* memilih mengubah data guru.   1. Sistem menampilkan data guru yang sudah terdaftar. 2. *Actor* memilih data guru yang ingin diubah. 3. Sistem menampilkan data guru yang sudah dipilih. 4. *Actor* memasukan data baru. 5. Kembali ke *basic flow* langkah ke-enam.   A-2 : *actor* memilih menghapus data guru.   1. Sistem menampilkan data guru yang sudah terdaftar. 2. *Actor* memilih data guru yang ingin dihapus. 3. Sistem memberikan pesan bahwa data berhasil dihapus. 4. Kembali ke *basic flow* langkah ke-sembilan   A-3 : *actor* memilih mencari data guru.   1. Sistem meminta *actor* untuk menginputkan kata kunci. 2. *Actor* menginput kata kunci pencarian. 3. Sistem melakukan pencarian.   E-2: Data yang dicari tidak ditemukan.   1. Sistem menampilkan data guru yang dicari di *view* guru. 2. Kembali ke *basic flow* langkah ke-sembilan. |
| *Error flow* | E-1: Data guru tidak lengkap.   1. Sistem memberitahu *actor* bahwa data guru tidak lengkap. 2. Kembali ke *basic flow* langkah ke-lima.   E-2: Data yang dicari tidak ditemukan.   1. Sistem mengosongkan *view* guru. 2. Kembali ke *alternative flow* (A-3) langkah ke-dua. |
| *Pre conditions* | 1. *Use case login* sudah dilakukan 2. *Actor* sudah memasuki sistem |
| *Post condition* | Data guru ter-*update* |

Tabel 3. Spesifikasi *Use case* : admin mengolah data guru (lanjutan)

Tabel 4. Spesifikasi *Use case* : admin mengolah data kelas

|  |  |
| --- | --- |
| *Use case name* | Admin mengolah data kelas |
| *Brief description* | *Use case* ini digunakan *actor* untuk melakukan pengolahan data kelas. *User* dapat menambahkan, mengubah, mencari dan menghapus data. |
| *Actor* | Admin |
| *Basic flow* | 1. *Use case* ini dimulai ketika *actor* memilih untuk mengolah data kelas*.* 2. Sistem memberikan pilihan kepada *actor* untuk menambah, mengubah, menghapus atau mencari data kelas. 3. *Actor* memilih menambah data kelas.   A-1 : *actor* memilih mengubah data kelas.  A-2 : *actor* memilih menghapus data kelas.  A-3 : *actor* memilih mencari data kelas.   1. Sistem menampilkan *form* data kelas. 2. *Actor* mengisi data kelas. 3. *Actor*  memberitahukan kepada sistem bahwa data sudah diisi. 4. Sistem memeriksa data kelas.E-1: Data kelas tidak lengkap. 5. Sistem menyimpan informasi ke basis data. 6. *Use case* selesai. |
| *Alternative flow* | A-1 : *actor* memilih mengubah data kelas.   1. Sistem menampilkan data kelas yang sudah terdaftar. 2. *Actor* memilih data kelas yang ingin diubah. 3. Sistem menampilkan data kelas yang sudah dipilih. 4. *Actor* memasukan data baru. 5. Kembali ke *basic flow* langkah ke-enam.   A-2 : *actor* memilih menghapus data kelas.   1. Sistem menampilkan data kelas yang sudah terdaftar. 2. *Actor* memilih data kelas yang ingin dihapus. 3. Sistem memberikan pesan bahwa data berhasil dihapus. 4. Kembali ke *basic flow* langkah ke-sembilan   A-3 : *actor* memilih mencari data kelas.   1. Sistem meminta *actor* untuk menginputkan kata kunci. 2. *Actor* menginput kata kunci pencarian. 3. Sistem melakukan pencarian.   E-2: Data yang dicari tidak ditemukan.   1. Sistem menampilkan data kelas yang dicari di *view* kelas. 2. Kembali ke *basic flow* langkah ke-sembilan. |
| *Error flow* | E-1: Data kelas tidak lengkap.   1. Sistem memberitahu *actor* bahwa data kelas tidak lengkap. 2. Kembali ke *basic flow* langkah ke-lima.   E-2: Data yang dicari tidak ditemukan.   1. Sistem mengosongkan *view* kelas. 2. Kembali ke *alternative flow* (A-3) langkah ke-dua. |
| *Pre conditions* | 1. *Use case login* sudah dilakukan 2. *Actor* sudah memasuki sistem |
| *Post condition* | Data kelas ter-*update* |

Tabel 4. Spesifikasi *Use case* : admin mengolah data kelas (lanjutan)

Tabel 5. Spesifikasi *Use case* : Admin mengolah data mata pelajaran

|  |  |
| --- | --- |
| *Use case name* | Admin mengolah data mata pelajaran |
| *Brief description* | *Use case* ini digunakan *actor* untuk melakukan pengolahan data mata pelajaran. *User* dapat menambahkan, mengubah, mencari dan menghapus data. |
| *Actor* | Admin |
| *Basic flow* | 1. *Use case* ini dimulai ketika *actor* memilih untuk mengolah data mata pelajaran*.* 2. Sistem memberikan pilihan kepada *actor* untuk menambah,mengubah,menghapus atau mencari data mata pelajaran. 3. *Actor* memilih menambah data mata pelajaran.   A-1 : *actor* memilih mengubah data mata pelajaran.  A-2 : *actor* memilih menghapus data mata pelajaran.  A-3 : *actor* memilih mencari data mata pelajaran.   1. Sistem menampilkan *form* data mata pelajaran. 2. *Actor* mengisi data mata pelajaran. 3. *Actor*  memberitahukan kepada sistem bahwa data sudah diisi. 4. Sistem memeriksa data mata pelajaran.E-1: Data mata pelajaran tidak lengkap. 5. Sistem menyimpan informasi ke basis data. 6. *Use case* selesai. |
| *Alternative flow* | A-1 : *actor* memilih mengubah data mata pelajaran.   1. Sistem menampilkan data mata pelajaran yang sudah terdaftar. 2. *Actor* memilih data mata pelajaran yang ingin diubah. 3. Sistem menampilkan data mata pelajaran yang sudah dipilih. 4. *Actor* memasukan data baru. 5. Kembali ke *basic flow* langkah ke-enam.   A-2 : *actor* memilih menghapus data mata pelajaran.   1. Sistem menampilkan data mata pelajaran yang sudah terdaftar. 2. *Actor* memilih data mata pelajaran yang ingin dihapus. 3. Sistem memberikan pesan bahwa data berhasil dihapus. 4. Kembali ke *basic flow* langkah ke-sembilan   A-3 : *actor* memilih mencari data mata pelajaran.   1. Sistem meminta *actor* untuk menginputkan kata kunci. 2. *Actor* menginput kata kunci pencarian. 3. Sistem melakukan pencarian.   E-2: Data yang dicari tidak ditemukan.   1. Sistem menampilkan data mata pelajaran yang dicari di *view* mata pelajaran. 2. Kembali ke *basic flow* langkah ke-sembilan. |
| *Error flow* | E-1: Data mata pelajaran tidak lengkap.   1. Sistem memberitahu *actor* bahwa data mata pelajaran tidak lengkap. 2. Kembali ke *basic flow* langkah ke-lima.   E-2: Data yang dicari tidak ditemukan.   1. Sistem mengosongkan view mata pelajaran. 2. Kembali ke *alternative flow* (A-3) langkah ke-dua. |
| *Pre conditions* | 1. *Usecase login* sudah dilakukan 2. *Actor* sudah memasuki sistem |
| *Post condition* | Data mata pelajaran ter-*update* |

Tabel 5. Spesifikasi *Use case* : Admin mengolah data mata pelajaran (lanjutan)

Tabel 6. Spesifikasi *Use case* : Admin mengolah data mata pelajaran (Lanjutan)

Tabel 6. Spesifikasi *Use case* : Admin mengolah data nilai siswa

|  |  |
| --- | --- |
| *Use case name* | Admin mengolah data nilai siswa terbaik |
| *Brief description* | *Use case* ini digunakan *actor* untuk melakukan pengolahan data nilai siswa terbaik. *actor* dapat menambahkan, mengubah, mencari dan menghapus data. |
| *Actor* | Admin |
| *Basic flow* | 1. *Use case* ini dimulai ketika *actor* memilih untuk mengolah data nilai siswa terbaik*.* 2. Sistem memberikan pilihan kepada *actor* untuk menambah,mengubah,menghapus atau mencari data nilai siswa terbaik. 3. *Actor* memilih menambah data nilai siswa terbaik.   A-1 : *actor* memilih mengubah data nilai siswa terbaik.  A-2 : *actor* memilih menghapus data nilai siswa terbaik.  A-3 : *actor* memilih mencari data nilai siswa terbaik.   1. Sistem menampilkan tugas siswa yang sudah dikerjakan. 2. *Actor* mengisi data nilai siswa terbaik tugas siswa yang sudah dikerjakan. 3. *Actor*  memberitahukan kepada sistem bahwa data sudah diisi. 4. Sistem memeriksa data nilai siswa terbaik.E-1: Data nilai siswa terbaik tidak lengkap. 5. Sistem menyimpan informasi ke basis data. 6. *Use case* selesai. |
| *Alternative flow* | A-1 : *actor* memilih mengubah data nilai siswa terbaik.   1. Sistem menampilkan data nilai siswa terbaik yang sudah terdaftar. 2. *Actor* memilih data nilai siswa terbaik yang ingin diubah. 3. Sistem menampilkan data nilai siswa terbaik yang sudah dipilih. 4. *Actor* memasukan data baru. 5. Kembali ke *basic flow* langkah ke-enam.   A-2 : *actor* memilih menghapus data nilai siswa terbaik.   1. Sistem menampilkan data nilai siswa terbaik yang sudah terdaftar. 2. *Actor* memilih data nilai siswa terbaik yang ingin dihapus. 3. Sistem memberikan pesan bahwa data berhasil dihapus. 4. Kembali ke *basic flow* langkah ke-sembilan   A-3 : *actor* memilih mencari data nilai siswa terbaik.   1. Sistem meminta *actor* untuk menginputkan kata kunci. 2. *Actor* menginput kata kunci pencarian. 3. Sistem melakukan pencarian.   E-1: Data yang dicari tidak ditemukan.   1. Sistem menampilkan data nilai siswa terbaik yang dicari di *view* nilai siswa terbaik. 2. Kembali ke *basic flow* langkah ke-sembilan. |
| *Error flow* | E-1: Data nilai siswa terbaik tidak lengkap.   1. Sistem memberitahu *actor* bahwa data nilai siswa terbaik tidak lengkap. 2. Kembali ke *basic flow* langkah ke-lima.   E-2: Data yang dicari tidak ditemukan.   1. Sistem mengosongkan view nilai siswa terbaik. 2. Kembali ke *alternative flow* (A-3) langkah ke-dua. |
| *Pre conditions* | 1. *Use case login* sudah dilakukan 2. *Actor* sudah memasuki sistem |
| *Post condition* | Data nilai siswa terbaik ter-*update* |

Tabel 6. Spesifikasi *Use case* : Admin mengolah data nilai siswa (lanjutan)

Tabel 7. Spesifikasi *Use case*: Admin mengolah data Jadwal mata pelajaran siswa.

|  |  |
| --- | --- |
| *Use case name* | Admin mengolah data Jadwal mata pelajaran siswa |
| *Brief description* | *Use case* ini digunakan *actor* untuk melakukan pengolahan data Jadwal mata pelajaran siswa. *actor* dapat menambahkan, mengubah, mencari dan menghapus data. |
| *Actor* | admin |
| *Basic flow*  Tabel 7. Spesifikasi *Use case*: Admin mengolah data Jadwal mata pelajaran siswa (lanjutan) | 1. *Use case* ini dimulai ketika *actor* memilih untuk mengolah data Jadwal mata pelajaran siswa*.* 2. Sistem memberikan pilihan kepada *actor* untuk menambah, mengubah, menghapus atau mencari data Jadwal mata pelajaran siswa. 3. *Actor* memilih menambah data Jadwal mata pelajaran siswa.   A-1 : *actor* memilih mengubah data Jadwal mata pelajaran siswa.  A-2 : *actor* memilih menghapus data Jadwal mata pelajaran siswa.  A-3 : *actor* memilih mencari data Jadwal mata pelajaran siswa.   1. Sistem menampilkan tugas siswa yang sudah dikerjakan. 2. *Actor* mengisi data Jadwal mata pelajaran siswa tugas siswa yang sudah dikerjakan. 3. *Actor*  memberitahukan kepada sistem bahwa data sudah diisi. 4. Sistem memeriksa data Jadwal mata pelajaran siswa.E-1: Data Jadwal mata pelajaran siswa tidak lengkap. 5. Sistem menyimpan informasi ke basis data. 6. *Use case* selesai. |
| *Alternative flow* | A-1 : *actor* memilih mengubah data Jadwal mata pelajaran siswa.   1. Sistem menampilkan data Jadwal mata pelajaran siswa yang sudah terdaftar. 2. *Actor* memilih data Jadwal mata pelajaran siswa yang ingin diubah. 3. Sistem menampilkan data Jadwal mata pelajaran siswa yang sudah dipilih. 4. *Actor* memasukan data baru. 5. Kembali ke *basic flow* langkah ke-enam.   A-2 : *actor* memilih menghapus data Jadwal mata pelajaran siswa.   1. Sistem menampilkan data Jadwal mata pelajaran 2. siswa yang sudah terdaftar. 3. *Actor* memilih data Jadwal mata pelajaran siswa yang ingin dihapus. 4. Sistem memberikan pesan bahwa data berhasil dihapus. 5. Kembali ke *basic flow* langkah ke-sembilan   A-3 : *actor* memilih mencari data Jadwal mata pelajaran siswa.   1. Sistem meminta *actor* untuk menginputkan kata kunci. 2. *Actor* menginput kata kunci pencarian. 3. Sistem melakukan pencarian.   E-2: Data yang dicari tidak ditemukan.   1. Sistem menampilkan data Jadwal mata pelajaran siswa yang dicari di *view* jadwal mata pelajaran siswa. 2. Kembali ke *basic flow* langkah ke-sembilan. |
| *Error flow* | E-1: Data Jadwal mata pelajaran siswa tidak lengkap.   1. Sistem memberitahu *actor* bahwa data Jadwal mata pelajaran siswa tidak lengkap. 2. Kembali ke b*asic flow* langkah ke-lima.   E-2: Data yang dicari tidak ditemukan.   1. Sistem mengosongkan *view* jadwal mata pelajaran siswa. 2. Kembali ke *alternative flow* (A-3) langkah ke-dua. |
| *Pre conditions* | 1. *Use case login* sudah dilakukan. 2. *Actor* sudah memasuki sistem. |
| *Post condition* | Data Jadwal mata pelajaran siswa ter-*update* |

Tabel 7. Spesifikasi *Use case*: Admin mengolah data Jadwal mata pelajaran siswa (lanjutan)

Tabel 8. Spesifikasi *Use case* : Guru mencetak Nilai

|  |  |
| --- | --- |
| *Use case name* | Guru mencetak raport |
| *Brief description* | *Use case* ini digunakan *actor* untuk mencetak raport. *User* dapat memilih siswa dan mencetak raport siswa tersebut. |
| *Actor* | Guru |
| *Basic flow* | 1. *Use case* ini dimulai ketika *actor* memilih untuk mencetak raport*.* 2. Sistem memberikan pilihan kepada *actor* untuk memilih siswa. 3. *Actor* memilih siswa yang inggin dicetak raportnya. 4. Sistem menampilkan data siswa yang dipilih. 5. *Actor* memilih tombol cetak untuk mencetak rapot. 6. *Use case* selesai. |
| *Alternative flow* | *None* |
| *Error flow* | *None* |
| *Pre conditions* | 1. *Use case login* sudah dilakukan 2. *Actor* sudah memasuki sistem |
| *Post condition* | *None* |

Tabel 9. Spesifikasi *Use case* : Guru mengolah data kehadiran siswa

|  |  |
| --- | --- |
| *Use case name* | Guru mengolah data Keterangan |
| *Brief description* | *Use case* ini digunakan *actor* untuk melakukan pengolahan data Keterangan. *actor* dapat menambahkan, mengubah, mencari dan menghapus data. |
| *Actor* | Guru |
| *Basic flow* | 1. *Use case* ini dimulai ketika *actor* memilih untuk mengolah data Keterangan*.* 2. Sistem memberikan pilihan kepada *actor* untuk menambah, mengubah, menghapus atau mencari data Keterangan.   Tabel 9. Spesifikasi *Use case* : Guru mengolah data kehadiran siswa (lanjutan)   1. *Actor* memilih menambah data Keterangan.   A-1 : *actor* memilih mengubah data Keterangan.  A-2 : *actor* memilih menghapus data Keterangan.  A-3 : *actor* memilih mencari data Keterangan.   1. Sistem menampilkan tugas siswa yang sudah dikerjakan. 2. *Actor* mengisi data Keterangan tugas siswa yang sudah dikerjakan. 3. *Actor*  memberitahukan kepada sistem bahwa data sudah diisi. 4. Sistem memeriksa data Keterangan.   E-1: Data Keterangan tidak lengkap.   1. Sistem menyimpan informasi ke basis data. 2. *Use case* selesai. |
| *Alternative flow* | A-1 : *Actor* memilih mengubah data Keterangan.   1. Sistem menampilkan data Keterangan yang sudah terdaftar. 2. *Actor* memilih data Keterangan yang ingin diubah. 3. Sistem menampilkan data Keterangan yang sudah dipilih. 4. *Actor* memasukan data baru. 5. Kembali ke *basic flow* langkah ke-enam.   A-2 : *actor* memilih menghapus data Keterangan.   1. Sistem menampilkan data Keterangan yang sudah terdaftar. 2. *Actor* memilih data Keterangan yang ingin dihapus. 3. Sistem memberikan pesan bahwa data berhasil dihapus. 4. Kembali ke *basic flow* langkah ke-sembilan   A-3 : *actor* memilih mencari data Keterangan.   1. Sistem meminta *actor* untuk menginputkan kata kunci. 2. *Actor* menginput kata kunci pencarian. 3. Sistem melakukan pencarian.   E-2: Data yang dicari tidak ditemukan.   1. Sistem menampilkan data Keterangan yang dicari di *view* Keterangan. 2. Kembali ke *basic flow* langkah ke-sembilan. |
| *Error flow* | E-1: Data Keterangan tidak lengkap.   1. Sistem memberitahu *actor* bahwa data Keterangan tidak lengkap. 2. Kembali ke *basic flow* langkah ke-lima.   E-2: Data yang dicari tidak ditemukan.   1. Sistem mengosongkan view Keterangan. 2. Kembali ke *alternative flow* (A-3) langkah ke-dua. |
| *Pre conditions* | 1. *Use case login* sudah dilakukan 2. *Actor* sudah memasuki sistem |
| *Post condition* | Data Keterangan ter-*update* |

Tabel 10. Spesifikasi *Use case* : Guru mengolah data nilai

|  |  |
| --- | --- |
| *Use case name* | Guru mengolah data nilai |
| *Brief description* | *Use case* ini digunakan *actor* untuk melakukan pengolahan data nilai. User dapat menambahkan, mengubah, mencari dan menghapus data. |
| *Actor* | Guru |
| *Basic flow* | 1. *Use case* ini dimulai ketika *actor* memilih untuk mengolah data nilai*.* 2. Sistem memberikan pilihan kepada *actor* untuk menambah, mengubah, menghapus atau mencari data nilai. 3. *Actor* memilih menambah data nilai.   A-1 : *actor* memilih mengubah data nilai.  A-2 : *actor* memilih menghapus data nilai.  A-3 : *actor* memilih mencari data nilai.   1. Sistem menampilkan tugas siswa yang sudah dikerjakan. 2. *Actor* mengisi data nilai tugas siswa yang sudah dikerjakan. 3. *Actor*  memberitahukan kepada sistem bahwa data sudah diisi. 4. Sistem memeriksa data nilai.E-1: Data nilai tidak lengkap. 5. Sistem menyimpan informasi ke basis data. 6. *Use Case* selesai. |
| *Alternative flow* | A-1 : A*ctor* memilih mengubah data nilai.   1. Sistem menampilkan data nilai yang sudah terdaftar. 2. *Actor* memilih data nilai yang ingin diubah. 3. Sistem menampilkan data nilai yang sudah dipilih. 4. *Actor* memasukan data baru. 5. Kembali ke *basic flow* langkah ke-enam.   A-2 : A*ctor* memilih menghapus data nilai.   1. Sistem menampilkan data nilai yang sudah terdaftar. 2. *Actor* memilih data nilai yang ingin dihapus. 3. Sistem memberikan pesan bahwa data berhasil dihapus. 4. Kembali ke *basic flow* langkah ke-sembilan   A-3 : A*ctor* memilih mencari data nilai.   1. Sistem meminta *actor* untuk menginputkan kata kunci. 2. *Actor* menginput kata kunci pencarian. 3. Sistem melakukan pencarian.   E-2: Data yang dicari tidak ditemukan.   1. Sistem menampilkan data nilai yang dicari di view nilai. 2. Kembali ke *basic flow* langkah ke-sembilan. |
| *Error flow* | E-1: Data nilai tidak lengkap.   1. Sistem memberitahu *actor* bahwa data nilai tidak lengkap. 2. Kembali ke *basic flow* langkah ke-lima.   E-2: Data yang dicari tidak ditemukan.   1. Sistem mengosongkan *view* nilai. 2. Kembali ke *alternative flow*(A-3) langkah ke-dua. |
| *Pre conditions* | 1. *Use case login* sudah dilakukan 2. *Actor* sudah memasuki sistem |
| *Post condition* | Data nilai ter-*update* |

Tabel 10. Spesifikasi *Use case* : Guru mengolah data nilai (lanjutan)

1. ***Sequence diagram***

*Sequence diagram* adalah suatu diagram yang menggambarkan interaksi antar objek di dalam sebuah sistem. Interaksi tersebut berupa pesan yang digambarkan terhadap waktu. Perancangan *sequence diagram* Aplikasi Pengolahan data nilai siswa SD Inpres Bello dapat dilihat pada Gambar 4 berikut.

1. *Sequence diagram login*

*Admin*, guru dan siswa memasukan *username* dan *password* ke dalam sistem melalui *form* yang telah disediakan. Sistem secara otomatis akan memvalidasi username dan *password* yang telah dimasukan. Apabila benar, maka admin, guru atau siswa bisa masuk ke dalam sistem. Kemudian apabila salah akan dikembalikan ke halaman *login*.

Gambar 4*. Sequence diagram login*

1. *Sequence diagram* mengolah data siswa
2. *Insert* data siswa

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data siswa untuk menginputkan data siswa. Jika data valid maka akan disimpan oleh sistem ke *database*.

Gambar 5*. Sequence diagram insert* data siswa

1. *Update* data siswa

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data siswa untuk menginputkan data siswa. Jika data *valid* maka akan disimpan oleh sistem ke *database*.

Gambar 6*. Sequence diagram update* data siswa

1. *Delete* data siswa

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data siswa untuk menghapus data siswa. Jika data *valid* maka akan disimpan oleh sistem ke *database.*

Gambar 7*. Sequence diagram delete* data siswa

1. Cari data siswa

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data siswa untuk mencari data siswa. Jika data *valid* maka akan disimpan oleh sistem ke *database.*

Gambar 8*. Sequence diagram* cari data siswa

1. *Sequence diagram* mengelola data guru
2. *Insert* data guru

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data guru untuk menginputkan data guru. Jika data valid maka akan disimpan oleh sistem ke *database*.

Gambar 9*. Sequence diagram insert* data guru

1. *Update* data guru

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data guru untuk mengubah data guru. Jika data valid maka akan disimpan oleh sistem ke *database*.

Gambar 10. *Sequence diagram update* data guru

1. *Delete* data guru

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data guru untuk menghapus data guru. Jika data valid maka akan disimpan oleh sistem ke *database*.

Gambar 11*. Sequence diagram delete* data guru

1. Cari data guru

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data guru untuk mencari data guru. Jika data valid maka akan disimpan oleh sistem ke *database*.

Gambar 12*. Sequence diagram* cari data guru

1. *Sequence diagram* mengelola data kelas
2. *Insert* data kelas

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data kelas untuk menginputkan data kelas. Jika data valid maka akan disimpan oleh sistem ke *database*.

Gambar 13. *Sequence diagram insert* data kelas

1. *Update* data kelas

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data kelas untuk mengubah data kelas. Jika data valid maka akan disimpan oleh sistem ke *database*.

Gambar 14*. Sequence diagram update* data kelas

1. *Delete* data kelas

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data kelas untuk menghapus data kelas. Jika data valid maka akan disimpan oleh sistem ke *database*.

Gambar 15*. Sequence diagram delete* data kelas

1. Cari data kelas

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data kelas untuk mencari data kelas. Jika data valid maka akan disimpan oleh sistem ke *database*.

Gambar 16*. Sequence diagram* caridata kelas

1. *Sequence diagram* mengelola data mata pelajaran
2. *Insert* data mata pelajaran

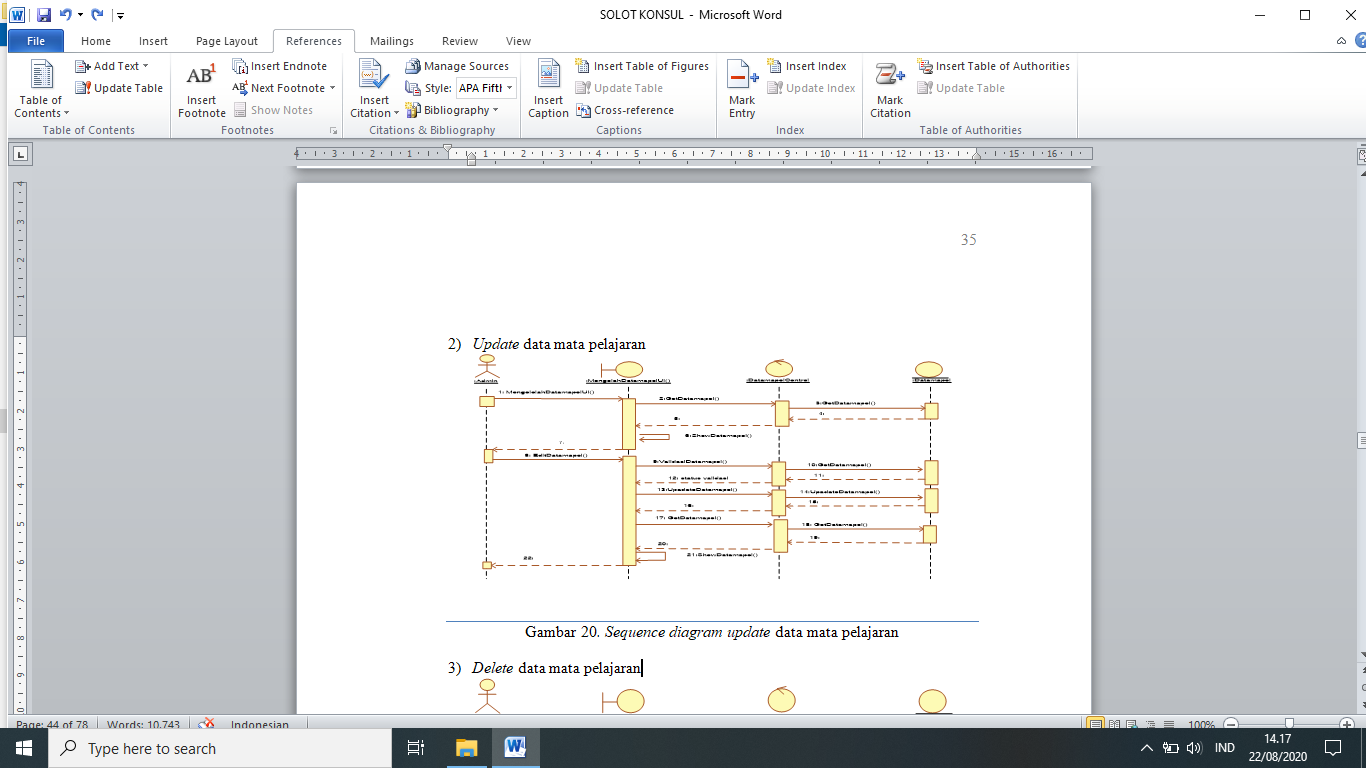
*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data mata pelajaran untuk menginputkan data mata pelajaran. Jika data valid maka akan disimpan oleh sistem ke *database*.



Gambar 17*. Sequence diagram insert* data mata pelajaran

1. *Update* data mata pelajaran

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data mata pelajaran untuk mengubah data mata pelajaran. Jika data *valid* maka akan disimpan oleh sistem ke *database*.



Gambar 18*. Sequence diagram update* data mata pelajaran

1. *Delete* data mata pelajaran

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data mata pelajaran untuk menghapus data mata pelajaran. Jika data *valid* maka akan disimpan oleh sistem ke *database*.

Gambar 19*. Sequence diagram delete* data mata pelajaran

1. Cari data mata pelajaran

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data mata pelajaran untuk mencari data mata pelajaran. Jika data valid maka akan disimpan oleh sistem ke *database*.

Gambar 20*. Sequence diagram* cari data mata pelajaran

1. *Sequence diagram* mengelola data nilai siswa
2. *Insert* data nilai siswa

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data nilai siswa untuk menginputkan data nilai siswa. Jika data valid maka akan disimpan oleh sistem ke *database*.



Gambar 21*. Sequence diagram insert* data nilai siswa

1. *Update* data nilai siswa

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data nilai siswa untuk mengubah data nilai siswa. Jika data *valid* maka akan disimpan oleh sistem ke *database*.



Gambar 22*. Sequence diagram update* data nilai siswa terbaik

1. *Delete* data nilai siswa

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data nilai siswa untuk menghapus data nilai siswa. Jika data *valid* maka akan disimpan oleh sistem ke *database*.



Gambar 23*. Sequence diagram delete* data nilai

1. cari data nilai siswa

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data nilai siswa untuk mencari data nilai siswa. Jika data *valid* maka akan disimpan oleh sistem ke *database*.

Gambar 24*. Sequence diagram* cari data nilai

1. *Sequence diagram* cetak data raport

*Admin*, guru wali kelas memasukan *username* dan *password* kedalam sistem melalui *form* yang telah disediakan. Sistem secara otomatis akan memvalidasi username dan *password* yang telah dimasukan. Setelah itu akan masuk ke halaman data cetak raport kemudian admin, guru akan memilih data cetak raport dan akan menampilkan hasil.

Gambar 25*. Sequence diagram* Cetak data raport

1. *Sequence diagram* mengelola data jadwal mata pelajaran
2. *insert* data jadwal mata pelajaran

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data jadwal mata pelajaran untuk menginputkan data jadwal mata pelajaran. Jika data valid maka akan disimpan oleh sistem ke *database*.

Gambar 26*. Sequence diagram insert* data jadwal mata pelajaran

1. *Update* data jadwal mata pelajaran

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data jadwal mata pelajaran untuk mengubah data jadwal mata pelajaran. Jika data *valid* maka akan disimpan oleh sistem ke *database*.



Gambar 27*. Sequence diagram update* data jadwal mata pelajaran

1. *Delete* data jadwal mata pelajaran

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data jadwal mata pelajaran untuk menghapus data jadwal mata pelajaran . Jika data valid maka akan disimpan oleh sistem ke *database*.



Gambar 28*. Sequence diagram delete* data jadwal mata pelajaran

1. Cari jadwal mata pelajaran

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data jadwal mata pelajaran untuk mencari data jadwal mata pelajaran. Jika data valid maka akan disimpan oleh sistem ke *database*.



Gambar 29*. Sequence diagram* cari data jadwal mata pelajaran

1. *Sequence diagram* mengelola data kehadiran siswa
2. *Insert* data keterangan kehadiran siswa

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data keterangan kehadiran siswa untuk menginputkan data keterangan kehadiran siswa. Jika data *valid* maka akan disimpan oleh sistem ke *database*.



Gambar 30*. Sequence diagram insert* data kehadiran siswa

1. *Update* data kehadiran siswa

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data keterangan kehadiran siswa untuk mengubah data keterangan kehadiran siswa. Jika data valid maka akan disimpan oleh sistem ke *database*.

Gambar 31*. Sequence diagram update* data kehadiran siswa

1. *Delete* data kehadiran siswa

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data keterangan kehadiran siswa untuk menghapus data keterangan kehadiran siswa. Jika data valid maka akan disimpan oleh sistem ke *database*.

Gambar 32*. Sequence diagram delete* data kehadiran siswa

1. Cari data kehadiran siswa

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data keterangan kehadiran siswa untuk mencari data keterangan kehadiran siswa. Jika data valid maka akan disimpan oleh sistem ke *database*.

Gambar 33*. Sequence diagram* cari data kehadiran siswa

1. *Sequence diagram* mengelola data nilai
2. *Insert* data nilai

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data nilai untuk menginputkan data nilai. Jika data valid maka akan disimpan oleh sistem ke *database*.

Gambar 34*. Sequence diagram insert* data nilai

1. *Update* data nilai

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data nilai untuk mengubah data nilai. Jika data valid maka akan disimpan oleh sistem ke *database*.

Gambar 35*. Sequence diagram update* data nilai

1. *Delete* data nilai

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data nilai untuk menghapus data nilai. Jika data valid maka akan disimpan oleh sistem ke *database*.

Gambar 36*. Sequence diagram delete* data nilai

1. Cari data nilai

*Admin* masuk ke halaman utama administrator, setelah itu masuk ke halaman data nilai untuk mencari data nilai. Jika data valid maka akan disimpan oleh sistem ke *database*.

Gambar 37*. Sequence diagram* cari data nilai

1. ***Class Diagram***

*Class diagram* merupakan suatu diagram yang memperlihatkan atau menampilkan struktur dari sebuah sistem, sistem tersebut akan menampilkan sistem kelas, atribut, dan hubungan antara kelas ketika suatu sistem telah selesai membuat diagram. Gambar *class diagram* dapat dilihat pada Gambar 31 berikut.



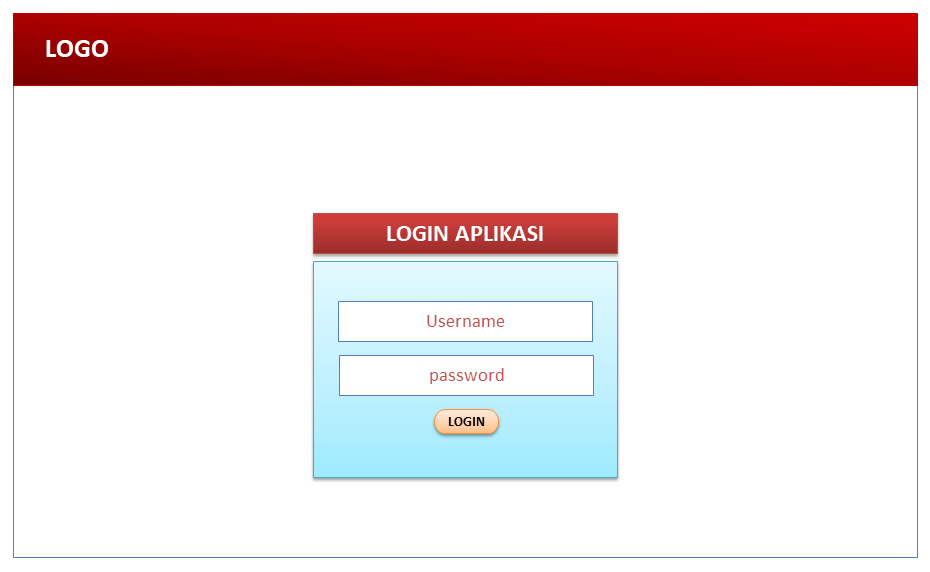
Gambar 38*. Class diagram* Aplikasi Sistem Informasi Akademik Berbasis *Web* SD Inpres Bello

1. **Perancangan Antarmuka**
2. Perancangan antarmuka halaman user

Perancangan antarmuka merupakan rancangan bangun dari percakapan antara pemakai dengan sistem komputer. Percakapan yang terdiri dari proses memasukan data, menampilkan keluaran informasi, maupun keduanya. Berikut tampilan antarmuka dapat dilihat pada gambar berikut.

1. Perancangan antarmuka menu *login user*

Perancangan antarmuka *login* merupakan rancangan bangun dari percakapan pemakai sistem dengan *website*. Dalam *login* terdiri dari *username* dan *password.*



Gambar 39. Perancangan antarmuka *login user*

1. Perancangan antarmuka menu mengelola data siswa

Perancangan antarmuka menu data siswa berfungsi untuk mengelola data siswa.



Gambar 40. Perancangan antarmuka mengelola data siswa

1. Perancangan antarmuka menu mengelola data guru

Perancangan antarmuka menu data guru berfungsi mengelola data guru.

Gambar 41. Perancangan antarmuka mengelola data guru

1. Perancangan antarmuka menu mengelolah data kelas

Perancangan antarmuka menu data kelas berfungsi mengelola data kelas.

Gambar 42. Perancangan antarmuka mengelola data kelas

1. Perancangan antarmuka menu mengelola data mata pelajaran

Perancangan antarmuka menu data mata pelajaran berfungsi mengelola data mata pelajaran.

Gambar 43. Perancangan antarmuka mengelolah data mata pelajaran

1. Perancangan antarmuka menu mengelola data Nilai

Perancangan antarmuka menu data nilai siswa terbaik berfungsi mengelola data nilai siswa.

Gambar 44. Perancangan antarmuka mengelola data nilai

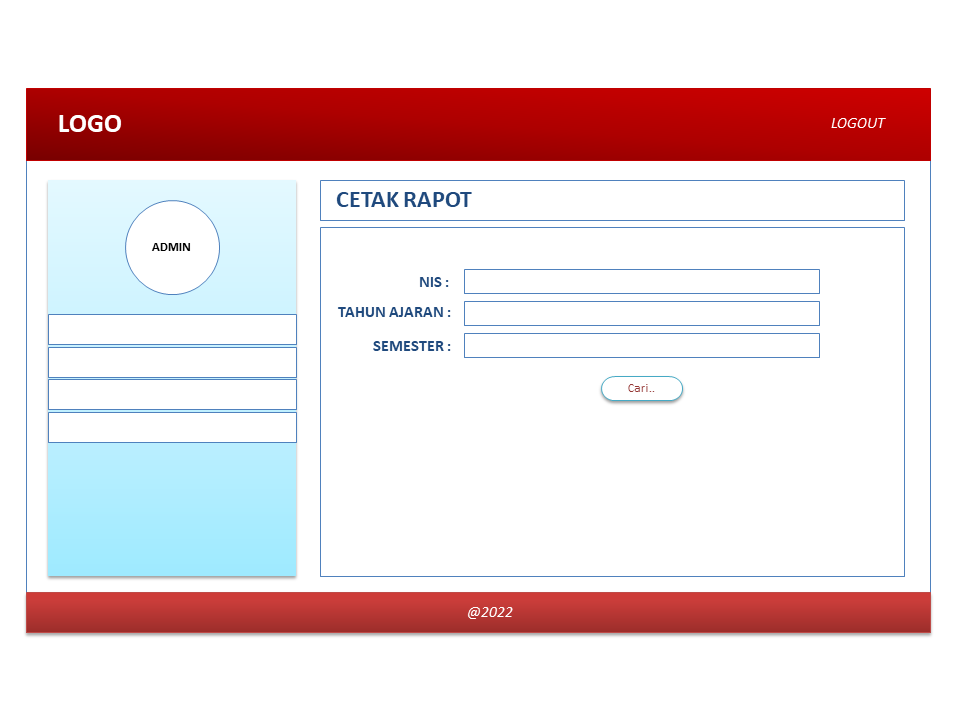
1. Perancangan antarmuka menu mengelola data jadwal mata pelajaran

Perancangan antarmuka menu data jadwal mata pelajaran berfungsi mengelola data jadwal mata pelajaran.

Gambar 45. Perancangan antarmuka mengelola data jadwal mata pelajaran

1. Perancangan antarmuka mencetak Nilai

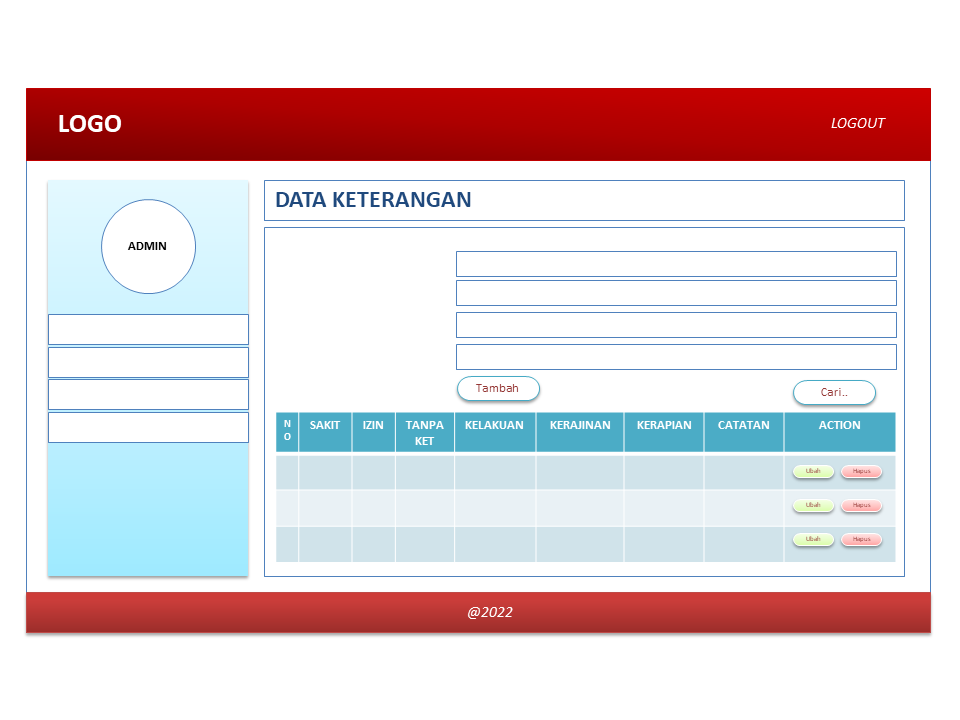
Perancangan antarmuka menu mencetak nilai berfungsi untuk mengelola data nilai.



Gambar 46. Perancangan antarmuka mencetak nilai

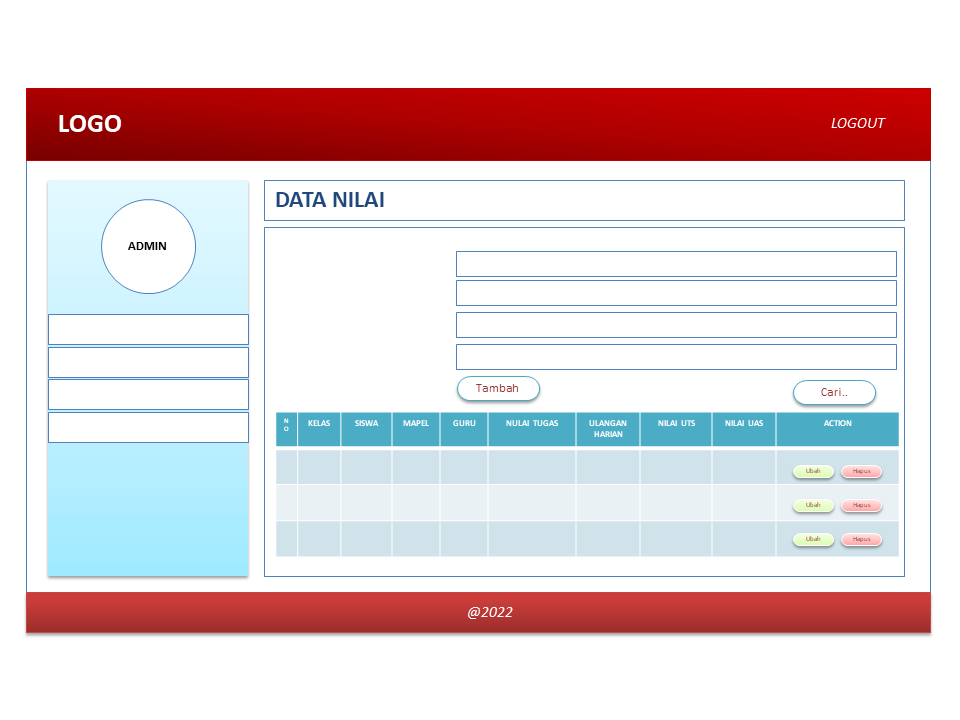
1. Perancangan antarmuka mengelola data kehadiran siswa

Perancangan antarmuka menu data keterangan berfungsi mengelola data kehadiran siswa.

Gambar 47. Perancangan antarmuka mengelola data kehadiran siswa

1. Perancangan antarmuka mengelola data nilai

Perancangan antarmuka menu data nilai berfungsi mengelola data nilai.

Gambar 48. Perancangan antarmuka mengelola data nilai