**LAPORAN KERJA PRAKTEK**

**SISTEM RAPORT ONLINE SD NEGERI KARANGAYU 02  
BERBASIS WEB**

****

**DISUSUN OLEH :**

**ARSI ANGGRAENI HAPSARI  
G.211.16.0029**

**PROGRAM STUDI S1-TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI**

**UNIVERSITAS SEMARANG**

**2021**

**DAFTAR ISI**

BAB I PENDAHULUAN 1

1.1 Latar Belakang 1

1.2 Batasan Masalah 2

1.3 Tujuan Kerja Praktek 2

1.4 Manfaat Kerja Praktek 2

1.5 Metodologi Penelitian 3

1.5.1 Obyek Kerja Praktek 3

1.5.2 Jenis dan Metode Pengumpulan Data 3

1.5.3 Metode Pengembangan Sistem 4

1.6 Sistematika Penulisan 5

BAB II TINJAUAN UMUM 7

2.1 Profil SD N Karangayu 02 7

2.2 Visi Misi SD N Karangayu 02 8

2.3 Struktur Organisasi 8

2.4 Fungsi dan Tugas Sekolah 9

2.5 Lokasi SD N Karangayu 02 11

BAB III LANDASAN TEORI 12

3.1 Sistem 12

3.2 Aplikasi 12

3.3 Raport 12

3.3.1 Fungsi Raport 12

3.4 Web 13

3.5 *Hyper Text Markup Language* (HTML) 13

3.6 *Cascading Style Sheet* (CSS) 14

3.7 Bootstrap 14

3.8 JavaScript 14

3.9 JQuery 15

3.10 PHP 15

3.10.1 PHP *Native* 15

3.11 *Structure Query Language* (SQL) 16

3.12 MySQL 17

3.13 Alat Bantu Dalam Perancangan 17

3.13.1 *Use Case Diagram* 18

3.13.2 *Class Diagram* 19

3.13.3 *Activity Diagram* 20

3.13.4 *Squence Diagram* 21

BAB IV ANALISA DAN PEMBAHASAN 24

4.1 Analisa Sistem 24

4.2 Analisa Kebutuhan 24

4.2.1 Analisa Kebutuhan *User* 24

4.2.2 Analisa Kebutuhan *Software* 24

4.2.3 Analisa Kebutuhan *Hardware* 24

4.3 Perancangan Sistem 25

4.3.1 *Use Case Diagram* 25

4.3.2 Skenario *Use Case* 25

4.3.3 Activity Diagram 30

4.3.4 Squence Diagram 34

4.3.5 Class Diagram 36

4.4 Perancangan Struktur Database 37

4.5 Perancanngan Antar Muka 42

4.5.1 Desain Tampilan Menu Utama 42

4.5.2 Desain Tampilan Menu Admin 43

4.5.3 Desain Tampilan Menu Siswa 43

4.5.4 Tampilan Nilai Siswa 44

4.5.5 Tampilan Raport Siswa 44

BAB V PENUTUP 48

5.1 Kesimpulan 48

5.2 Saran 48

Daftar Pustaka 50

**DAFTAR PUSTAKA**

Gambar 1.1 Model *Prototype* (Rosa dan Shalahuddin, 20013) 5

Gambar 2.1 SD Negeri Karangayu 02 7

Gambar 2.2 Struktur Organisasi SD Negeri Karangayu 02 9

Gambar 2.3 Lokasi SD Negeri Karangayu 02 11

Gambar 4.1 *Use Case Diagram* 25

Gambar 4.2 *Activity Diagram* Kelola Data Guru 31

Gambar 4.3 *Activity Diagram* Kelola Data Siswa 31

Gambar 4.4 *Activity Diagram* Kelola Data Akademik 32

Gambar 4.5 *Activity Diagram* Kelola Data Nilai 33

Gambar 4.6 *Activity Diagram* Cetak Raport Siswa 33

Gambar 4.7 *Sequence Diagram* Kelola Data Guru 34

Gambar 4.8 *Sequence Diagram* Kelola Data Siswa 34

Gambar 4.9 *Sequence Diagram* Kelola Data Akademik 35

Gambar 4.10 *Sequence Diagram* Kelola Data Nilai 35

Gambar 4.11 *Sequence Diagram* Cetak Raport Siswa 36

Gambar 4.12 *Class Diagram* 36

Gambar 4.13 Desain Tampil Menu Utama 42

Gambar 4.14 Desain Tampil Menu Admin 43

Gambar 4.15 Desain Tampilan Menu Guru 43

Gambar 4.16 Desain Tampilan Menu Siswa 44

Gambar 4.17 Tampilan Nilai Siswa 44

Gambar 4.18 Tampilan Raport Siswa 45

**Daftar Tabel**

Tabel 2.1 Fungsi dan Tugas Sekolah 9

Tabel 3.1 Simbol *Use Case Diagram* 18

Tabel 3.2 Simbol *Class Diagram* 19

Tabel 3.3 Simbol *Activity Diagram* 20

Tabel 3.4 Simbol *Sequence Diagram* 22

Tabel 4.1 Skenario *Use Case* Kelola Data Guru 25

Tabel 4.2 Skenario *Use Case* Kelola Data Siswa 27

Tabel 4.3 Skenario *Use Case* Kelola Data Akademik 28

Tabel 4.4 Skenario *Use Case* Kelola Data Nilai 29

Tabel 4.5 Skenario *Use Case* Cetak Raport 30

Tabel 4.6 Tabel Admin 37

Tabel 4.7 Tabel Guru 37

Tabel 4.8 Tabel Kelas 38

Tabel 4.9 Tabel Kelas Guru 38

Tabel 4.10 Tabel Mapel 39

Tabel 4.11 Tabel Nilai 39

Tabel 4.12 Tabel Pengguna 40

Tabel 4.13 Tabel Prestasi 40

Tabel 4.14 Tabel Semester 40

Tabel 4.15 Tabel Sikap 41

Tabel 4.16 Tabel Siswa 41

Tabel 4.17 Tabel Tahun Ajar 42

**BAB 1**

**PENDAHULUAN**

* 1. **Latar Belakang Masalah**

Sekolah adalah salah satu sarana organisasi dalam memberikan pelayanan kepada masyarakat dalam bidang pendidikan. Salah satu bagian terpenting dari suatu sekolah adalah siswa dan nilai siswa. Dalam satu sekolah ada ratusan siswa dan masing-masing mempunyai nilai yang berbeda-beda. Tak jarang penyimpanan nilai siswa dicatat dan disimpan secara konvensional, sehingga membutuhkan waktu yang sangat lama dalam pengerjaannya. Sedangkan hasil dari pengolahan data nilai siswa hanya bisa dilihat baik oleh siswa maupun orang tua siswa hanya pada buku raport. Hal ini juga menyebabkan para orang tua tidak bisa setiap saat mengecek nilai putra/putrinya di sekolah.

Sistem pengolahan data nilai siswa pada SD Negeri Karangayu 02 Semarang belum sepenuhnya terkomputerisasi, masih ada prosedur yang dilakukan seperti menerima data penilaian siswa dari setiap guru mata pelajaran, setelah itu dicatat dalam buku kumpulan nilai dan menyalinnya kembali dalam satu buku yang disebut dengan raport. Setelah penyalinan tersebut, raport disimpan dan dibagikan kepada setiap siswa pada waktu penyerahan raport berlangsung. Dalam jangka waktu yang ditentukan oleh sekolah, siswa diminta untuk mengembalikan raport tersebut kepada Guru Wali setiap kelas. Selama raport berada pada siswa, tidak semua siswa bisa menjaga raport tersebut dengan baik. Masih ada juga siswa yang ceroboh dan hal ini yang dikhawatirkan bisa memnyebabkan raport hilang, robek, basah terkena air dan kelalaian siswa lainnya yang mengakibatkan kerusakan pada raport. Bukan dikalangan siswa saja, Guru Wali terkadang keliru atau lupa akan tempat penyimpanan raport siswanya. Hal ini dikarenakan lemari tempat penyimpanan raport yang kurang tertata rapi dan banyaknya raport siswa dalam satu lemari.

Berdasarkan uraian diatas, penulis bermaksud mengembangkan model sistem raport online SD N Karangayu 02 Semarang sehingga dapat memudahkan proses pengolahan nilai dan diharapkan mampu memberikan informasi yang baik dapat diakses dengan mudah dan efisien.

* 1. **Batasan Masalah**

Batasan masalah pada laporan kerja praktek ini adalah sebagai berikut :

1. Sistem ini dapat diakses oleh Admin, Guru Wali dan Siswa SD N Karangayu 02
2. Sistem ini khusus menangani pengelolaan nilai SD N Karangayu 02
3. Pembuatan sistem menggunakan PHP *Native* dan *MySQL* sebagai database penyimpanan
4. Analisa sistem menggunakan *UML* (*Unified Modelling Language*)
5. Metode pengembangan menggunakan *Prototype*
   1. **Tujuan Kerja Praktek**
6. Terbentuknya sistem nilai yang dapat mengolah nilai siswa, pencarian, meng-update, menyimpan dan laporan nilai
7. Membuat proses pengolahan nilai menjadi lebih mudah dan fleksibel
   1. **Manfaat Kerja Praktek**

Manfaat dalam pembuatan laporan ini diantaranya adalah :

1. Bagi penulis
2. Dapat mengukur pemahanan dan pengetahuan penulis terhadap teori-teori yang didapat selama perkuliahan
3. Sebagai pembelajaran serta menambah wawasan tentang dunia kerja khususnya mengenai sistem pengolahan nilai
4. Bagi SD N Karangayu 02
5. Membantu kinerja Guru Wali kelas dalam pencatatan nilai raport yang cepat dan efisien
6. Menunjang kecepatan dan ketepatan dalam penyajian informasi tentang perkembangan nilai siswa
7. Bagi FTIK USM

Menambah literature pada perpustakaan fakultas dan sebagi tolak ukur sejauh mana kemampuan mahasiswa dalam menguasai materi yang diterapkan di lapangan kerja

* 1. **Metodologi Penelitian**
     1. **Obyek Kerja Praktek**

Penelitian dilakukan di SD Negeri Karangayu 02 Jl. Kenconowungu IV/16 Kota Semarang. Selama 1 bulan mulai tanggal 30 Oktober s/d 30 November 2019

* + 1. **Jenis Data dan Metode Pengumpulan Data**

Untuk menyusun laporan yang baik dan benar diperlukan data-data yang akurat. Dalam usaha memperoleh data tersebut dapat menggunakan beberapa metode. Adapun jenis data dan metode yang digunakan aladah sebagai berikut :

1. Data Primer

Data primer adalah data-data yang diperoleh secara langsung dari sumber yang menjadi objek penelitian, yaitu pada SD N Karangayu 02. Data ini diperoleh dengan menggunakan 2 metode :

1. Metode Observasi

Metode pengumpulan data dengan cara melakukan pengataman langsung di SD N Karangayu 02. Disini penulis datang langsung ke SD N Karangayu 02 disambut oleh bapak Ibnu selaku Koordinator Kurikulum dan ibu Yuni selaku Kepala SD N Karangayu 02, berikutnya penulis diajak berkeliling SD, kemudian menuju ruang TU yang merupakan tempat penyimpanan raport siswa, pak Ibnu menjelaskan gambaran besar alur pengisian raport. Catatan yang diperoleh dari pengamatan ini adalah berupa data siswa, nilai UTS, nilai MID, nilai harian, absensi dan ektrakulikuler.

1. Metode Wawancara

Melakukan tanya jawab secara langsung kepada bapak Ibnu selaku Koordinator Kurikulum yang sekaligus merangkap sebagai Guru Wali. Disini penulis melakukan tanya jawab dengan pak Ibnu. Pak Ibnu memaparkan bagaimana beliau memperoleh nilai siswa dari guru mata pelajaran yang kemudian direkap menjadi satu di buku nilai, lalu baru memasukkan data tersebut satu satu ke dalam raport siswa. Disini penulis juga melalukan tanya jawab seputar alur pengambilan raport yang ditentukan sekolah untuk para wali siswa.

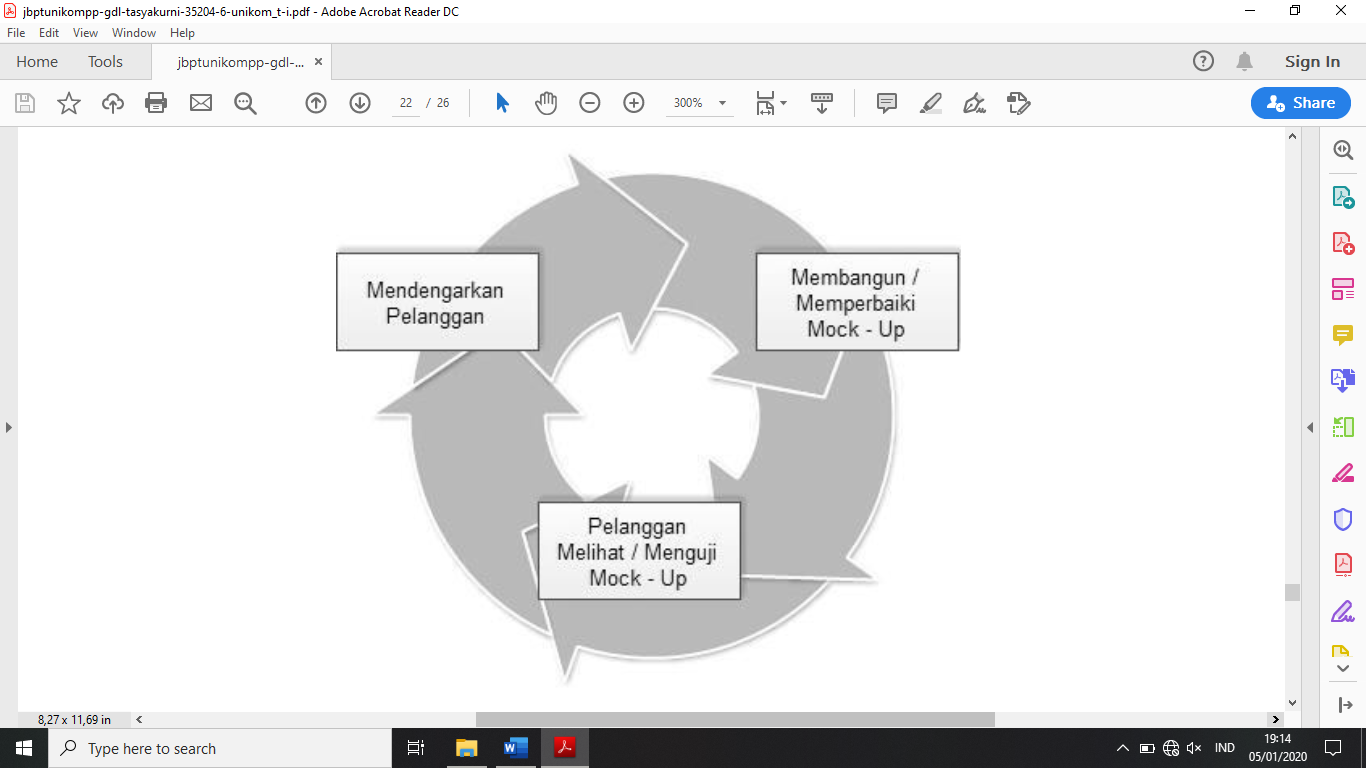
1. Data Sekunder

Data sekunder adalah data-data yang diperoleh dari berbagai sumber seperti buku, literatur dan artikel di internet yang dapat mendukung data primer. Data ini diperoleh menggunakan metode Studi Pustaka/Literatur, yang merupakan metode pengumpulan data dengan membaca dan mempelajari buku-buku yang berhubungan dengan masalah yang akan diteliti sebagai tinjauan pustaka. Disini penulis mendapat data sekunder dengan meminjam jurnal dan buku-buku yang ada di perpustakaan FTIK, serta mencari jurnal di internet sebagai referensi yang mana data-data yang penulis cari sudah tertera pada daftar pustaka pembuatan laporan.

* + 1. **Metode Pengembangan Sistem**

Metode pengembangan sistem yang digunakan dalam perancangan model Sistem Raport SD Negeri Karangayu 02 adalah metode Prototype. Prototype adalah perangkat lunak yang berupa model fisik kerja sistem dan berfungsi sebagai versi awal dari sistem.

Model Prototype sangat cocok digunakan dalam penjabaran kebutuhan customer secara lebih mendetail karena customer sering kali tidak menyampaikan kebutuhannya secara lengkap dan detail. Untuk mengantisipasi agar proyek dapat berjalan sesuai target waktu dan biaya, maka sebaiknya spesifikasi kebutuhan sistem harus sudah disepakati oleh pengembang dengan customer secara tertulis. Dokumen tersebut akan dijadikan patokan dalam pengerjaan sistem agar spesifikasi kebutuhan sistem masih berada didalam ruang lingkup proyek.



*Gambar 1.1 Model Prototyping* (A. S & Shalahuddin, Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek, 2013)*.*

* 1. **Sistematika Penulisan**

Penyusunan laporan kerja praktek ini berisi beberapa bab, yang masing-masing bab berisi uraian singkat terkait kerja praktek lapangan. Hal ini dimaksudkan agar pembahasan lebih sistematis dengan topic permasalahan yang ada. Berikut sistematika laporan ini yaitu :

**BAB I PENDAHULUAN**

Bab ini menjelaskan mengenai latar belakang masalah, batasan masalah, tujuan dan manfaat kerja praktek, metode penelitian dan sistematika penulisan.

**BAB II TINJAUAN UMUM INSTANSI**

Bab ini menjelaskan tentang sejarah singkat instansi, struktur organisasi, dan tugas serta fungsi masing-masing struktur.

**BAB III LANDASAN TEORI**

Bab ini menjelaskan dasar teori-teori pendukung yang digunakan untuk menyusun laporan kerja praktek.

**BAB IV ANALISA DAN PEMBAHASAN**

Bab ini berisi analisa dara, pembahasan perancangan dan pembuatan sistem serta hasil tampilan layout dari sistem yang telah dibuat.

**BAB V PENUTUP**

Bab ini berisi kesimpulan dari pembahasan dan saran-saran yang dapat bermanfaat dari pihak yang terkait.

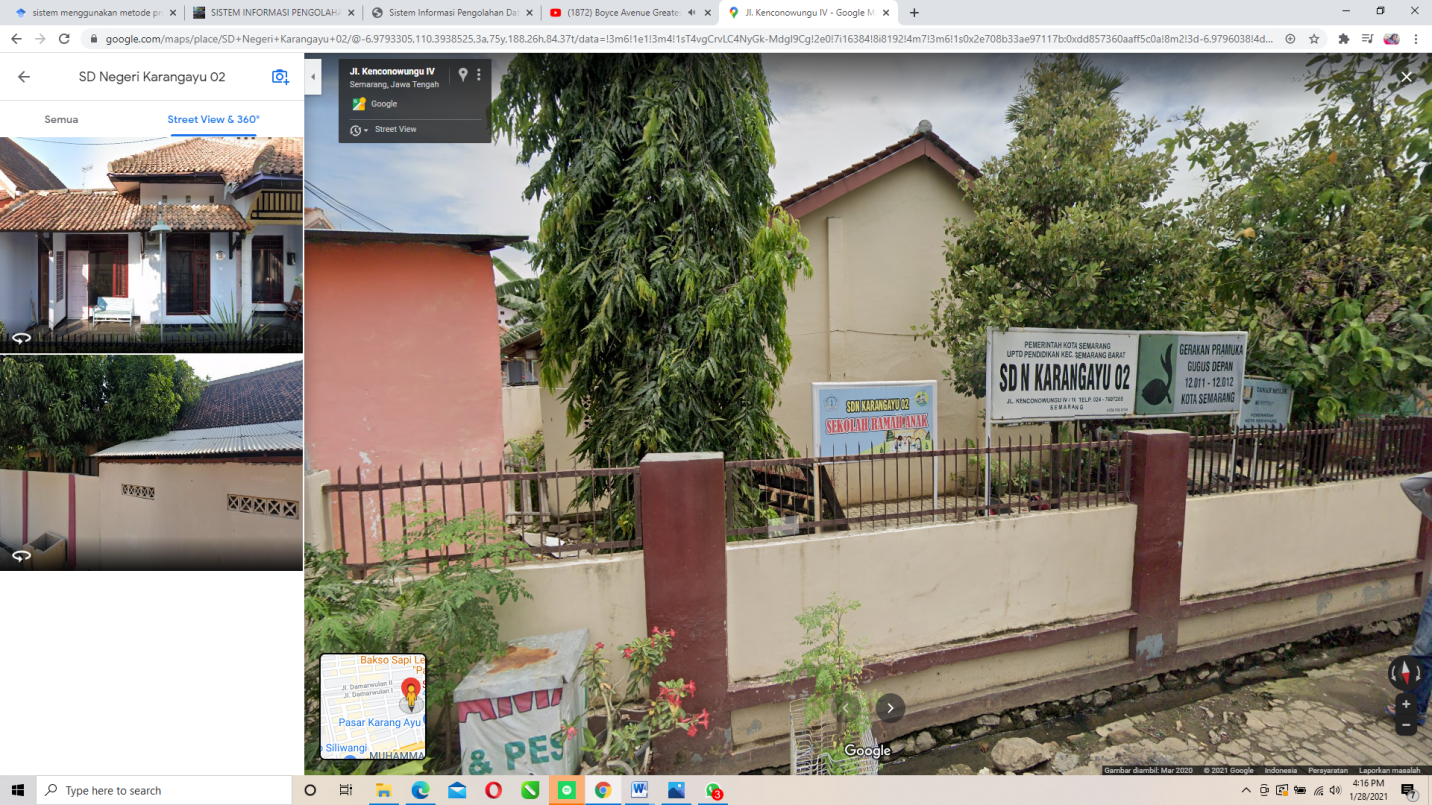
**DAFTAR PUSTAKA**

**LAMPIRAN**

**BAB II**

**TINJAUAN UMUM SD NEGERI KARANGAYU 02**

**2.1 Profil SD Negeri Karangayu 02 Semarang**

SD Negeri Karangayu 02 berdiri sejak tahun 1967. Pada mulanya SD N Karangayu 02 bernama SDUD (Sekolah Dasar Unit Desa). Kemudian nama sekolah berubah menjadi SD Inpres 02,03,04,05 pada tahun 1970. Selanjutnya nama sekolah berubah menjadi SD N Karangayu 02,03,04 pada tahun 1972. Pada tahun 1980 nama SD berubah lagi menjadi SD N Karangayu 02.

*Gambar 2.1 SD Negeri Karangayu 02 Semarang*

**2.2 Visi Misi SD N Karangayu 02 Semarang**

1. Visi

Visi merupakan cita-cita sebuah sekolahan yang ingin dicapai di masa depan untuk menjamin kesuksesan jangka panjang.

Visi SD N Karangayu 02 sebagai berikut :

a. Mewujudkan siswa yang cerdas, trampil, berdasarkan iman, taqwa, dan budi pekerti luhur.

2. Misi

Misi adalah sesuatu yang harus dilaksanakan agar tujuan sekolah dapat tercapai. Adapun misi dari SD N Karangayu 02 adalah :

a. Meningkatkan pelayanan PBM secara optimal kepada peserta didik dengan menggunakan strategi dan model PAIKEM

b. Memberikan pendidikan ketrampilan sesuai kebutuhan peserta didik.

c. Meningkatkan pendidikan agama kepada peserta didik sehingga menjadi insan yang beriman, taqwa, dan berakhlak mulia.

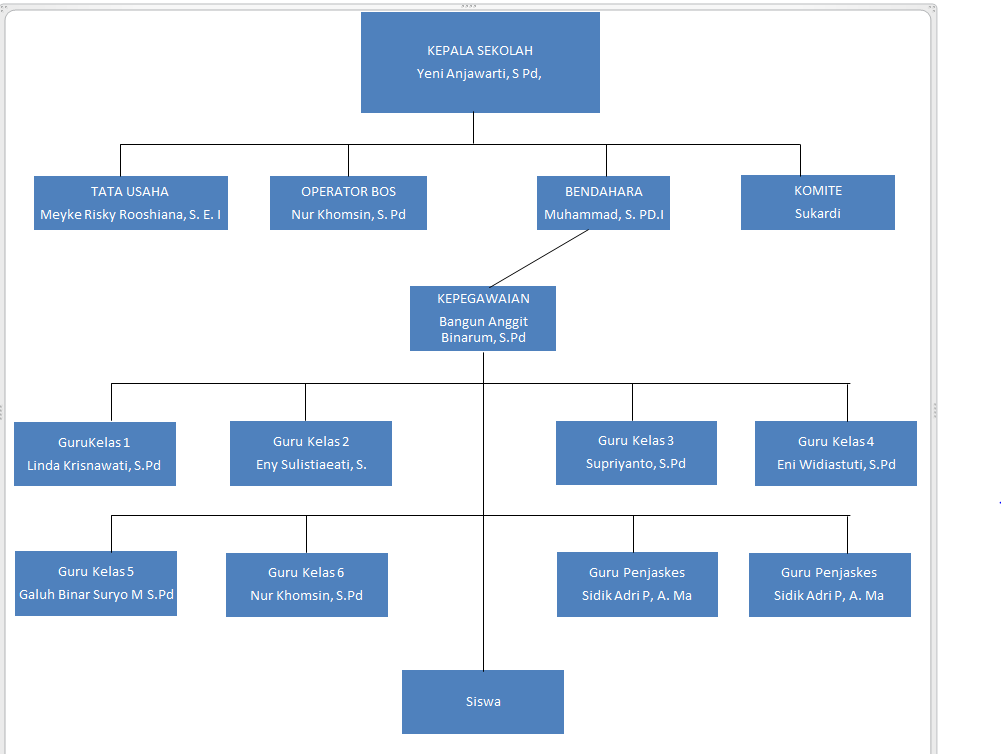
d. Memberikan motivasi kepada masyarakat (Ortu, Toga, Toma, dan unsur lain) untuk ikut serta dalam meningkatkan mutu pendidikan.

**2.3 Struktur Organisasi**

Pengorganisasian merupakan suatu fungsi manajemen yang dipandang sebagai alat yang dicapai oleh orang-orang atau organisasi untuk mencapai tujuan bersama efektif. Suatu instansi yang menginginkan kemajuan dan sistem kerja yang teratur harus mempunyai susunan organisasi yang teratur pula.

**STRUKTUR ORGANISASI SEKOLAH**

**SD NEGERI KARANGAYU 02 SEMARANG**



*Gambar 2.2 Struktur Organisasi SD Negeri Karangayu 02*

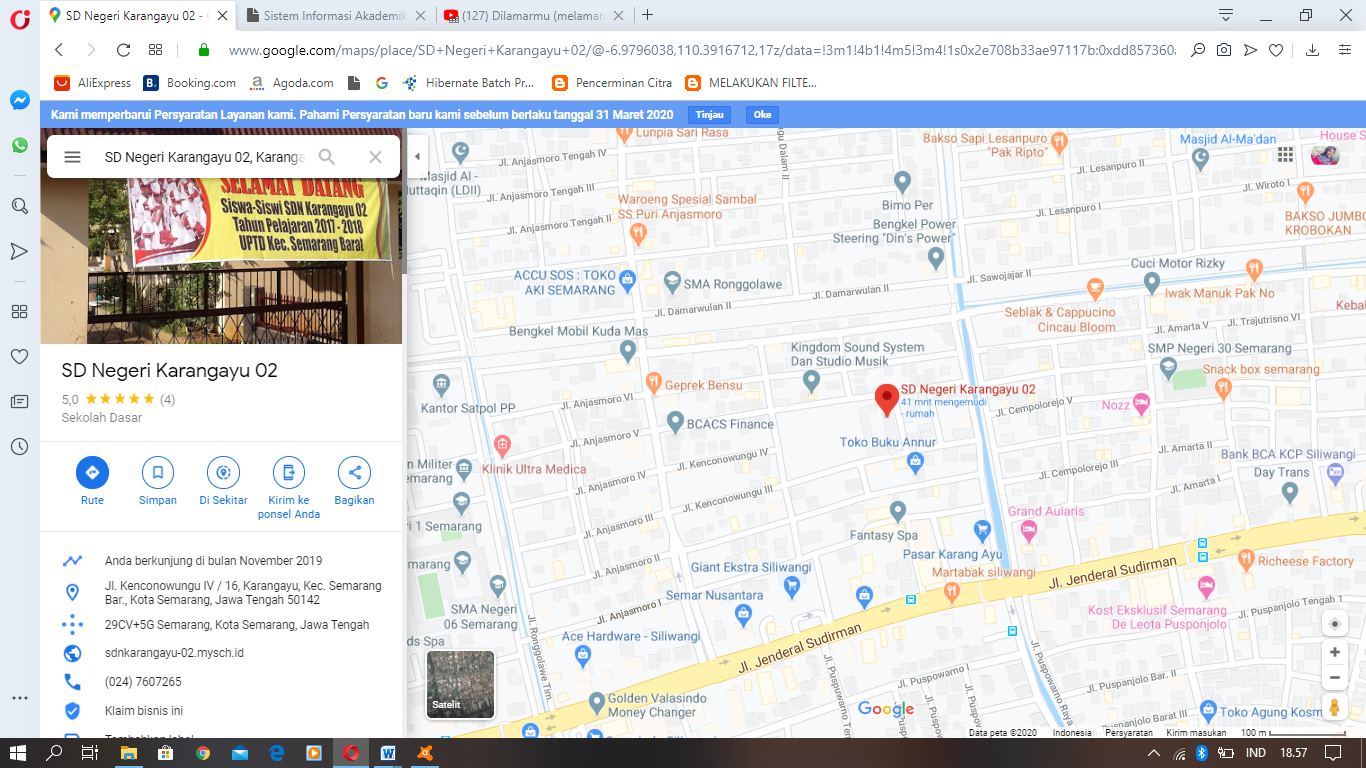
* 1. **Fungsi dan Tugas Sekolah**

*Tabel 2.1 Fungsi dan Tugas Sekolah*

|  |  |
| --- | --- |
| Jabatan | Tugas |
| Kepala Sekolah | 1. Mengelola guru dan staf dalam rangka pendayagunaan sumber daya manusia secara optimal 2. Melakukan perencanaan keuangan, dengan mengusulkan dan mengesahkan anggaran belanja dan juga anggaran pendapatan sekolah 3. Menyusun aturan dan juga tata tertib bagi guru, staff, dan juga murid secara adil dan objektif. |
| Tata Usaha | 1. Penyususnan program kerja tata usaha sekolah 2. Pengelolaan keuangan sekolah 3. Pengurusan administrasi ketenagaan dan siswa 4. Pembinaan dan pengembangan karir pegawai tata usaha sekolah 5. Penyusunan administrasi perlengkapan sekolah 6. Penyusunan dan oenyajian data atau statistic sekolah 7. Mengkoordinasi dan melaksanakan 7K 8. Penyusunan laporan pelaksanaan kegiatan pengurusan ketatausahaan secara berkala |
| Bendahara | 1. Membuat file keuangan sesuai dengan dana pembangunan 2. Membuat laporan dana pembangunan pada akhir tahun anggaran 3. Membuat laporan Rancangan Anggaran Pendapatan Bantuan Sekolah ( RAPBS ) |
| Kepegawaian | 1. Membuat kelengkapan mengajar dengan baik dan lengkap 2. Melaksanakan kegiatan pembelajaran 3. Melaksanakan kegiatan penilaian proses belajar, ualangan, dan ujian |
| Guru Kelas | 1. Mengelola kelas 2. Menyusun pembuatan statistic bulanan siswa 3. Pengisian daftar kumpulan nilai siswa 4. Pembuatan catatan khusus tentang siswa 5. Pencatatan mutasi siswa 6. Pengisian buku laporan penilaian hasil belajar 7. Pembagian buku lapor penilaian hasil belajar |

**2.5 Lokasi SD Negeri Karangayu 02**

Lokasi SD N Karangayu 02 sangatlah mudah ditemukan karena letaknya yang strategis dan mudah dijangkau yaitu berlokasi di Jl. Kenconowungu IV/16, Kota Semarang, Jawa Tengah, 50142 No. Telp (024) 7607265 Email: [sdnkarangayu02@yahoo.com](mailto:sdnkarangayu02@yahoo.com)



*Gambar 2.3 Lokasi SD Negeri Karangayu 02*

**BAB III**

**LANDASAN TEORI**

1. **Sistem**

Sistem berasal dri bahasa Latin (systema) dan bahasa Yunani (sustema) adalah suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi dimana suatu model matematika seringkali bisa dibuat. (Zaky, 2018)

Sedangkan menurut Murdick, R.G (2000) Sistem merupakan sekumpulan elemen yang terdiri dari prosedur atau bagan pengolahan untuk mencari tujuan bersama atau tujuan bagian dengan cara mengoperasikan barang atau data pada waktu tertentu agar bisa menghasilkan informasi, energi atau data yang diinginkan.

1. **Aplikasi**

Menurut (Widianti, 2000) Aplikasi merupakan sebuat *software* (perangkat lunak) yang bertugas sebagai *front end* pada sebuah sistem yang dipakai untuk mengolah berbagai macam data sehingga menjadi sebuah informasi yang bermanfaat untuk penggunaannya dan juga sistem yang berkaitan.

1. **Raport**

Raport adalah hasil yang dicapai dalam bentuk angka-angka skor setelah diberikan tes hasil belajar pada setiap akhir pembelajaran. Raport adalah buku yang berisi keterangan mengenai nilai kepandaian dan prestasi belajar murid di sekolah, yang biasanya dipakai sebagai laporan guru kepada orangtua siswa (Dimyati & Mudjiono, 2006).

**3.3.1 Fungsi Raport**

1. Bagi Siswa : Mengetahui kemajuan hasil belajar diri, konsep-konsep atau teori-teori yang belum dikuasai, untuk memotivasi diri untuk belajar lebih baik dan memperbaiki strategi belajar.

2. Bagi Orang Tua : Mengetahui perkembangan anaknya sehingga orang tua dapat membantu anaknya belajar, memotivasi untuk meningkatkan hasil belajar dan melengkapi fasilitas belajar di rumah.

3. Bagi Guru Mata Pelajaran : Sebagai penilaian yang digunakan guru untuk mengetahui kekuatan dan kelemahan siswa dalam satu kelas. Hasil penilaian harus dapat mendorong guru agar mengajar lebih baik, dan membantu guru untuk menentukan strategi mengajar yang lebih tepat.

4. Bagi Wali Kelas : Melalui raport, wali kelas dapat mengetahui kekuatan dan kelemahan siswa dalam kelas yang diampunya. Wali kelas dapat menentukan strategi dalam pengelolaan kelas yang menjadi tanggung jawabnya misalnya dengan menata ulang pengaturan tempat duduk, pembagian anggota kelompok belajar dan langkah strategis lainnya untuk membantu siswa meningkatkan kompetensi siswa atau membantu mengatasi kesulitan belajar siswa yang lemah.

1. **Web**

*World Wide Web* (*WWW*) atau yang lebih dekenal dengan web ditemukan oleh seorang berkebangsaan Inggris yang bernama *Sir Timothy John* “Tim” *Berners-Lee* sekitar tahun 1980-an. Awalnya web ini dibuat dengan tujuan untuk mempermudah tukar menukar dan memperbarui informasi kepada sesama peneliti di tempat dia berkerja.(Deni, 2013).

Web merupakan kumpulan infromasi pada server komputer yang terhubung satu sama lain dalam jaringan internet maupun intranet. Sedangkan aplikasi berbasis web (web based) secara prinsip menyerupai aplikasi dalam komputer biasa. Yang membedakan adalah aplikasi web based menggunakan tag-tag HTML (*Hypertext Markup Language)* sebagai dasar tampilan, sedangkan aplikasi program komputer menggunakan berbagai platform bahasa pemrograman.

Sejarah perkembangan bahasa pemrograman web dimulai dengan munculnya HTML. Kemudian dikembangkan dengan mucnulnya CSS (*Cascading Style Sheet)* yang bertujuan memperindah tampilan website.

1. ***Hyper Text Markup Language* (HTML)**

Menurut Efendi (2015) HTML adalah sebuah bahasa yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah web Internet dan pemformatan hiperteks sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi.

Dokumen HTML adalah file teks murni yang dapat dibuat dengan editor teks sembarang. Dokumen ini disebut dengan *webpage*. HTML sendiri adalah suatu dokumen teks biasa yang mudah dimengerti dibanding bahasa pemrograman lainnya. HTML dikeluarkan oleh *World Wide Web Consortium*, setiap terjadi perkembangan level HTML, harus dievaluasi ketat dan disetujui oleh *World Wide Web Consortium*.

1. ***Cascading Style Sheet* (CSS)**

*Cascading Style Sheet* merupakan salah satu bahasa standar pemrograman web. *Style Sheets* merupakan feature yang sangat penting dalam membuat *Dynamic* HTML. *Style Sheets* mendeskripsikan bagaimana tampilan dokumen HTML di layar.

1. **Bootstrap**

Bootstrap adalah sebuah *library framework* CSS yang dibuat khusus untuk bagian pengembangan front-end website. Bootstrap merupakan framework atau tools untuk membuat aplikasi web ataupun situs web responsive secara cepat, mudah dan gratis. Bootstrap terdiri dari CSS dan HTML untuk menghasilkan *Grid, Layout, Typography, Table, Form, Navigation,* dan lain-lain.

Dengan bantuan Bootstrap, kita bisa membuat responsive website dengan cepat dan mudah dan dapat berjalan sempurna pada browser-browser populer seperti Chrome, Firefox, Opera dan Internet Explorer. Bootstrap diciptakan oleh dua orang programmer di Twitter, yaitu Mark Otto dan Jacob Thornton pada tahun 2011.

1. **JavaScript**

Javascript menurut Efendi (2015) merupakan bahasa pemrograman komputer yang dinamis. Biasanya sering digunakan pada web brwoser untuk menciptakan halaman web yang menarik, interaktif, serta menerapkan berbagai fungsi pada halaman web. Javascript merupakan salah satu pemrograman web yang harus kita pelajari (selain HTML dan CSS).

Javascript bukan sebuah *compiled language,* artinya javascript tidak memerlukan sebuah compiler agar kode yang ada di dalamnya bisa dijalankan. Kode dari Javascript langsung diterjemahkan oleh web browser. Javascript biasanya ditulis pada dokumen HTML atau dengan membuat file terpisah yang kita hubungkan dengan dokumen HTML.

1. **JQuery**

JQuery adalah JavaScript *library* yang bersifat *opensource* dimana fungsinya adalah untuk menyederhanakan interaksi antara HTML dengan JavaScript.

JQuery dikembangkan pertama kali oleh John Resig, yang dibuat lebih ramping dari *libraryi prototype* yang menjadi inspirasi dari *library* jQuery ini. Secara pemrograman jQuery ini memiliki kemiripan seperti prototype.

JQuery adalah library nya, dan JavaScript adalah bahasanya.

1. **PHP**

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 perintah-perintah *PHP* akan dieksekusi diserver kemudian hasilnya akan dikirimkan 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.

**3.10.1 PHP *Native***

PHP *native* merupakan pemrograman web perpaduan bahasa pemprograman yang didasari dengan bahasa pemrograman PHP yang mana disisipi oleh text Javascript, CSS, Bootstrap dan lain-lain. *Native* sendiri artinya asli, yakni pemrograman php yang murni disusun dan di *coding* / dibangun oleh para programmer sendiri tanpa ada istilah tambahan settingan/konfigurasi lainnya.

Penggunaan PHP *Native* tidak cocok untuk membuat website besar seperti pemerintahan, online shop dan lain-lain dikarenakan file-file PHP yang dibuat oleh programmer akan dipisah dan memakan waktu yang sangat lama. PHP *Native* juga tidak memiliki fitur keamanan layaknya *CodeIgniter* dan Laravel yang mendeteksi dan mencegah serangan XSS, SQLi ataupun CSRF sebaliknya jika ingin ada fitur tersebut maka programmer harus membuat fitur itu sendiri.

PHP *Native* lebih cocok digunakan untuk pemakaian personal atau bisa di implementasikan untuk membuat website sekolah walau sudah adanya framework. PHP *Native* masih digunakan hingga saat ini walau tidak begitu banyak apalagi untuk mencari pekerjaan yang rata-rata harus memiliki persyaratan kemampuan di bidang framework.

PHP *Native* ini cocok digunakan untuk membuat aplikasi yang cukup sederhana. Performa merupakan salah satu indikator terpenting dari keberhasilan sebuah framework, jika programmer ingin mengejar performa, PHP *Native* dapat menghasilkan performa yang bisa lebih tinggi jika dibuat tanpa framework

Kekurangan php native tanpa framework adalah programmer harus membuat semuanya dari awal tapi karena ada bootstrap dan css jadi lebih memudahkan programmer native ini, untuk keamanan programmer mengamankan kodenya sendiri.

1. ***Structured Query Language* (SQL)**

SQL menurut Situmorang (2013) adalah sebuah bahasa yang digunakan untuk mengakses data dalam basis data relasional. Bahasa ini merupakan bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini hampir semua server basis data yang ada mendukung bahasa ini untuk melakukan manajemen datanya.

SQL sering juga disebut dengan istilah query. Di dalam *website* SQL akan sangat erat hubungannya dengan PHP, karena PHP merupakan struktur pemrograman pada sisi server sehingga server membutuhkan koneksi ke dalam database, dan SQL digunakan untuk menghubungkan website (dengan struktur bahasa PHP) ke dalam database menggunakan SQL. Contoh bahasa pemrograman SQL adalah MySQL, JQuery, dll.

1. **MySQL**

*MySQL* adalah sebuah perangkat lunak sistem manajemen basis data atau DBMS (*Database Management System)* yang *multithread* dan multi-user. DBMS merupakan perangkat lunak yang dipakai untuk membangun basis data yang berbasis komputerisasi. DBMS juga dapat membantu dalam memelihara serta pengolahan data dalam jumlah yang besar. MySQL menggunakan bahasa SQL untuk mengakses databaase nya. Lisensi MySQL adalah FOSS *License Exception* dan ada juga yang versi komersial nya. (Budi, 2011).

1. **Alat Bnatu Dalam Perancangan**

***UML (Unified Modelling Language)***

Pada perkembangan teknik berorientasi objek, munculah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek (A. S & Shalahuddin, Reakayasa Perangkat Lunak Terstruktur dan Berorientasi Objek, 2018).

Ada beberapa diagram yang disediakan dalam UML, antara lain sebagai berikut :

**3.13.1 *Use Case Diagram***

*Use* *Case* atau diagram merupakan pemodelan untuk sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Berikut adalah simbol-simbol yang ada pada diagram *use case*:

*Tabel 3.1. Simbol Use Case Diagram* (A. S & Shalahuddin, Reakayasa Perangkat Lunak Terstruktur dan Berorientasi Objek, 2018)*.*

|  |  |
| --- | --- |
| SIMBOL | DESKRIPSI |
| *Use case* | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling tertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama *Use Case* |
| Aktor / *actor*  **nama aktor** | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tetapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor |
| Asosiasi / *association* | Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor |
| Ekstensi/ *extend*  <<extend>> | Relasi *use case* tambahan ke sebuah *use case* dinamakan *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu; mirip dengan prinsip *inheritance* pada pemrograman berorientasi objek; biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambahkan |
| Generalisasi / *generalization* | Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang lebih umum dari lainnya |
| Menggunakan / *include* / *uses*  <<include>>  <<uses>> | Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini |

* + 1. ***Class Diagram***

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan didalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Banyak berbagai kasus, perancangan kelas yang dibuat tidak sesuai dengan kelas-kelas yang dibuat pada perangkat lunak, sehingga tidaklah ada gunanya lagi sebuah perancangan karena apa yang dirancang dan hasil jadinya tidak sesuai.

Berikut adalah simbol-simbol yang ada pada diagram kelas:

*Tabel 3.2. Simbol Class Diagram* (A. S & Shalahuddin, Reakayasa Perangkat Lunak Terstruktur dan Berorientasi Objek, 2018)*.*

|  |  |
| --- | --- |
| SIMBOL | DESKRIPSI |
| |  | | --- | | nama-kelas | | + atribut | | + operasi () |   Kelas | Kelas pada struktur sistem |
| antarmuka / *interface*  **nama\_interface** | Sama dengan konsep *interface* dalam pemrograman berorientasi objek |
| asosiasi / *association* | Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity* |
| asosiasi berarah / *directed association* | Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity* |
| generalisasi | Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus) |
| kebergantungan / *dependency* | Relasi antarkelas dengan makna kebergantungan antarkelas |
| agregasi / *aggregation* | Relasi antarkelas dengan makna semua-bagian (*whole-part*) |

* + 1. ***Activity Diagram***

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas yang dapat dilakukan oleh sistem.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

*Tabel 3.3. Simbol Activity Diagram* (A. S & Shalahuddin, Reakayasa Perangkat Lunak Terstruktur dan Berorientasi Objek, 2018)*.*

|  |  |
| --- | --- |
| SIMBOL | DESKRIPSI |
| status awal | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal |
| aktivitas  aktivitas | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja |
| percabangan / *decision* | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu |
| penggabungan / *join* | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu |
| status akhir | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir |
| Nama swimlane  Nama swimlane |  |

* + 1. ***Squence Diagram***

Diagram *squence* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram *squence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram squence juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Banyaknya diagram squence yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Berikut adalah simbol-simbol yang ada pada diagram squence:

*Tabel 3.4. Simbol Sequence Diagram* (A. S & Shalahuddin, Reakayasa Perangkat Lunak Terstruktur dan Berorientasi Objek, 2018)*.*

|  |  |
| --- | --- |
| SIMBOL | DESKRIPSI |
| Aktor  **nama aktor**  atau  nama aktor  tanpa waktu aktif | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tetapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor |
| Garis hidup / *lifeline* | Menyatakan kehidupan suatu objek |
| Objek  Nama objek : nama kelas | Menyatakan objek yang berinteraksi dengan pesan |
| Waktu Aktif | Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya |
| Pesan tipe *create*  *<<create*>> | Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat |
| Pesan tipe *call*  1 : nama\_metode() | Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi/metode, karena memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi |
| Pesan tipe *send*  1 : masukan | Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang menerima kembalian |
| Pesan tipe *return*  1 : keluaran | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian |
| Pesan tipe *destroy*  *<<destroy*>> | Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada *create* maka ada *destroy* |

**BAB IV**

**ANALISA DAN PEMBAHASAN**

**4.1 Analisa Sistem**

Perencanaan sistem dilakukan saat suatu kegiatan akan berjalan. Perencanaan sistem mengidentifikasikan pengolahan raport di SD N Karangayu 02 untuk mengetahui data nilai siswa dengan menggunakan aplikasi raport siswa. Perencanaan aplikasi ini dimulai dari pembuatan sebuah laporan perencanaan sebuah sistem yang menggunakan sumber informasi langsung dari SD N Karangayu 02 yang berhubungan dengan Raport Siswa.

## 4.2 Analisa Kebutuhan

**4.2.1 Analisa Kebutuhan *User***

Dalam melakukan tahap perencanaan, diperlukan untuk melakukan tahapan analisa kebutuhan User. Dalam tahap tersebut menghasilkan kesimpulan mengenai kebutuhan User yakni, dibutuhkannya aplikasi yang dapat membantu dalam pengolahan nilai raport siswa. Hal itu dibutuhkan karena berdasarkan pengalaman yang telah dialami wali kelas, yaitu tidak terkontrol dan kurang efektifnya dalam membuat laporan raport siswa.

**4.2.2 Analisa Kebutuhan *Software***

Perangkat lunak (*software*) yang digunakan dalam membangun aplikasi penggajian ini adalah sebagai berikut :

1. Sistem Operasi *Microsoft Windows* *10*
2. Xampp.
3. Notepad ++
4. Database MySQL
5. Google Chrome

**4.2.3** **Analisa Kebutuhan *Hardware***

Perangkat keras (*hardware*) yang dibutuhkan dalam menjalankan rancangan sistem ini adalah sebagai berikut :

1. Laptop dengan sistem operasi 64-bit
2. Processor AMD Ryzen 3 3200U with Radeon Vega Mobile Gfx 2.60 GHz
3. Memori RAM 4 GB
4. Mouse
5. Keyboard

## 4.3 Perancangan Sistem

Perancangan sistem didalam aplikasi penggajian ini penulis menggunakan metode *Unifield Modeling Language* (UML), yang terdiri dari beberapa diagram yang akan menggambarkan sistem secara keseluruhan yaitu *use case diagram, squence diagram, activity diagram dan class diagram.*

**4.3.1 Use case Diagram**



*Gambar 4.1 Use Case Diagram*

* + 1. **Skenario Use Case** 
       1. **Skenario Use Case Kelola Data Guru**

Interaksi antara Admin dengan *use case* dalam skenario *use case*

KelolaData Guru dapat dilihat di Tabel 4.1.

*Tabel 4.1 Skenario Use Case Kelola Data Guru*

|  |  |  |
| --- | --- | --- |
| Nama *Use Case* | KelolaData Guru | |
| Aktor | Admin | |
| Diskripsi | Admin melakukan kelola data guru yang akan dibuat maupun di update.  Terdapat fitur tambah guru, edit dan hapus. | |
| Prakondisi | Admin harus login terlebih dahulu sebagai Admin | |
| Bidang khas *event*: | Kegiatan Aktor | Respon Sistem |
|  | Memilih menu Guru | Menampilkan halaman menu data Guru |
|  | Klik Tambah Data | Menampilkan halaman input data Guru |
|  | Input data Guru kemudian klik simpan | Sistem akan memproses input data Guru ke database |
|  |  | Data berhasil di simpan |
|  | Klik edit | Sistem akan menampilkan form edit data Guru |
|  | Masukan data Guru yang ingin di ganti/ubah |  |
|  | Klik edit | Sistem akan memproses edit data |
|  |  | Data berhasis di edit |
|  | Klik hapus | Sistem akan memproses hapus data |
|  |  | Data berhasil di Hapus |
| Kesimpulan | Yang dapat di lakukan pada menu Guru yaitu tambah data Guru, edit data Guru, dan hapus data Guru | |
| Post Kondisi | Data Guru tersimpan ke dalam database. | |

* + - 1. **Skenario Use Case Kelola Data Siswa**

Interaksi antara Admin dengan *use case* dalam skenario *use case*

Kelola Data Siswa dapat dilihat di Tabel 4.2.

*Tabel 4.2 Skenario Use Case Kelola Data Siswa*

|  |  |  |
| --- | --- | --- |
| Nama *use case* | Kelola Data Siswa | |
| Aktor | Admin | |
| Diskripsi | Admin melakukan Kelola Data Siswa yang akan diinputkan maupun di update  Terdapat fitur tambah data, edit dan hapus. | |
| Prakondisi | Admin harus login terlebih dahulu sebagai Admin | |
| Bidang khas *event*: | Kegiatan Actor | Respon Sistem |
|  | Admin memilih menu Siswa | Sistem merespon dengan menampilkan halaman Data Siswa |
|  | Klik Tambah Data | Sistem akan menampilkan halaman input data Siswa |
|  | Admin melakukan input data Siswa dan klik simpan | Sistem pemroses data tersebut |
|  |  | Data berhasis di simpan |
|  | Admin klik edit data pada salah satu data siswa | Sistem menampilkan halam untuk edit data Siswa |
|  | Admin klik hapus data pada salah satu data |  |
|  |  | Sistem akan menghapus data tersebut dari database |
| Kesimpulan | Yang dapat di lakukan pada menu Siswa yaitu Tambah data Siswa, edit data Siswa, dan hapus data Siswa | |
| Post Kondisi | Data Siswa akan tersimpan di database. | |

* + - 1. **Skenario Use Case Kelola Data Akademik**

Interaksi antara *actor* Admin dengan *use case* dalam skenario *use case* Kelola Data Akademik dapat dilihat di Tabel 4.3.

*Tabel 4.3 Skenario Use Case Kelola Data Akademik*

|  |  |  |
| --- | --- | --- |
| Nama *use case* | Kelola Data Akademik | |
| Aktor | Admin. | |
| Diskripsi | Admin dapat melakukan Kelola Data Akademik | |
| Prakondisi | Admin harus login terlebih dahulu | |
| Bidang khas *event* | Kegiatan Aktor | Respon Sistem |
|  | Admin memilih menu Akademik | Sistem merespon dengan menampilkan halaman Kelas, Mapel, Nilai dan Tahun Ajaran |
|  | Klik Tambah Data | Sistem akan menampilkan halaman input data Kelas, Mapel |
|  | Admin melakukan input data Kelas dan Mapel kemudian klik simpan | Sistem pemroses data tersebut |
|  |  | Data berhasil di simpan |
|  | Admin klik edit data pada salah satu data Kelas dan Mapel | Sistem menampilkan halam untuk edit data Kelas dan Mapel |
|  | Admin klik hapus data pada salah satu data | Sistem akan memroses penghapusan |
| Kesimpulan | Yang dapat di lakukan pada halaman ini yaitu melakukan tambah data, update dan hapus data Kelas, Mapel. | |
| Post Kondisi | Data Kelas, Mapel di ambil dari database | |

* + - 1. **Skenario Use Case Kelola Data Nilai**

Interaksi antara aktor Guru dengan *use case* dalam skenario *use case* Kelola Data Nilai dapat dilihat di table 4.6

*Tabel 4.4 Skenario Use Case Kelola Data Nilai*

|  |  |  |
| --- | --- | --- |
| Nama *Use Case* | Kelola Data Nilai | |
| Aktor | Guru | |
| Diskripsi | Guru dapat melakukan Kelola Data Nilai | |
| Prakondisi | Guru harus login dulu | |
| Bidang khas *event* | Kegiatan Aktor | Respon Sistem |
|  | Klik menu Kelas | Sistem akan menampilkan halaman Kelas |
|  | Klik lihat kelas | Sistem otomatis menampilkan kelas sesuai yang di pilih |
|  | Memilih siswa yang akan di isi nilainya | Sistem akan menampilkan form input nilai |
|  | Klik simpan | Sistem akan pemproses data tersebut |
|  | Klik edit | Sistem akan mengupdate data tersebut |
| Kesimpulan | Guru hanya melakukan input dan update isi nilai siswa. | |
| Post Kondisi | Data di ambil dan di simpan di database | |

* + - 1. **Skenario Use Case Cetak Raport**

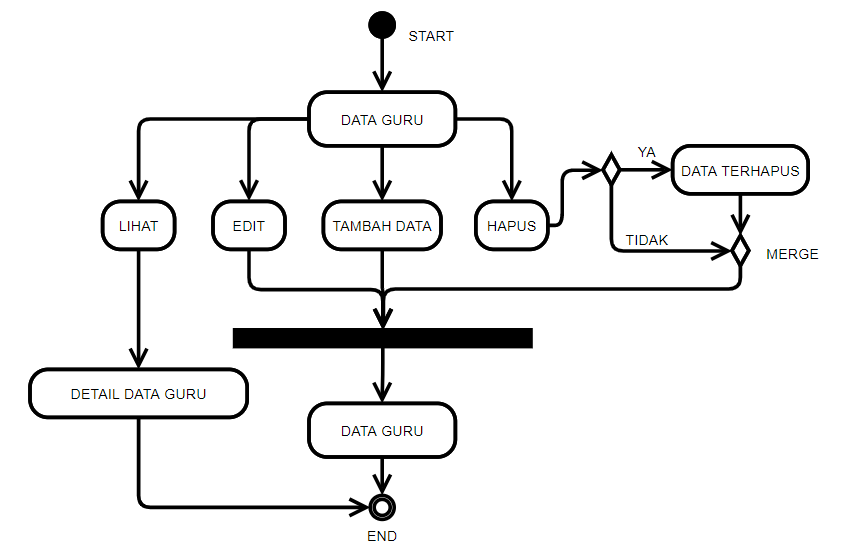
Interaksi antara actor Guru dan siswa dengan *use case* dalam skenario *use case* Cetak Raport dapat dilihat di table 4.5

*Tabel 4.5 Skenario Use Case Cetak Raport*

|  |  |  |
| --- | --- | --- |
| Nama *Use Case* | Cetak Raport | |
| Aktor | Guru dan Siswa | |
| Diskripsi | Guru dan Siswa melakukan cetak Cetak Raport | |
| Prakondisi | Guru dan Siswa harus login terlebih dahulu | |
| Bidang khas *event* | Kegiatan Aktor | Respon Sistem |
|  | Klik menu Cetak Raport | Sistem akan menampilkan halaman Cetak Pdf |
|  |  | Raport Siap di Cetak |
| Kesimpulan | Guru dan Siswa hanya melakukan Cetak Raport Siswa | |
| Post Kondisi | Data di ambil dari database | |

* + 1. ***Activity Diagram***
       1. ***Activity Diagram* Kelola Data Guru**

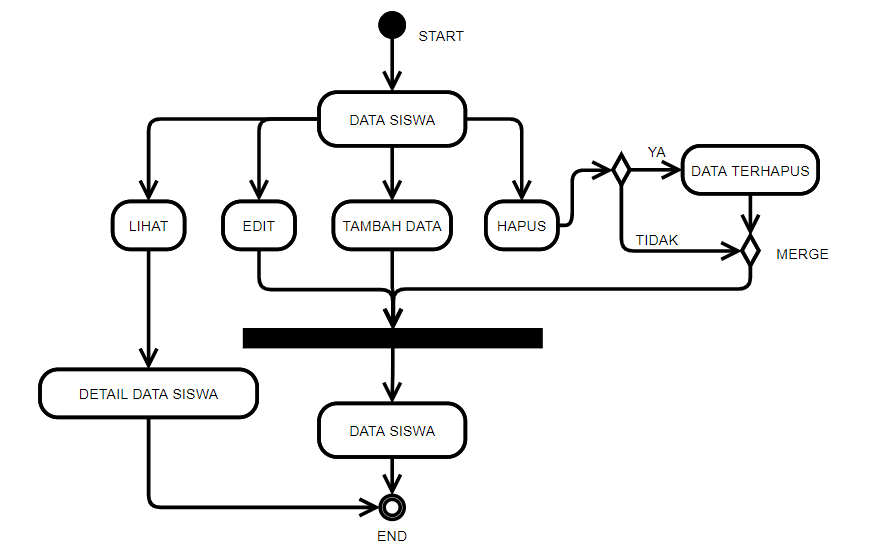
Berikut adalah *Activity Diagram* Kelola Data Guru, dimana Admin dapat Kelola Data Guru dengan cara, *update*, atau *delete* data guru ditunjukkan pada Gambar 4.2.



*Gambar 4.2 Activity Diagram Kelola Data Guru*

* + - 1. ***Activity Diagram* Kelola Data Siswa**

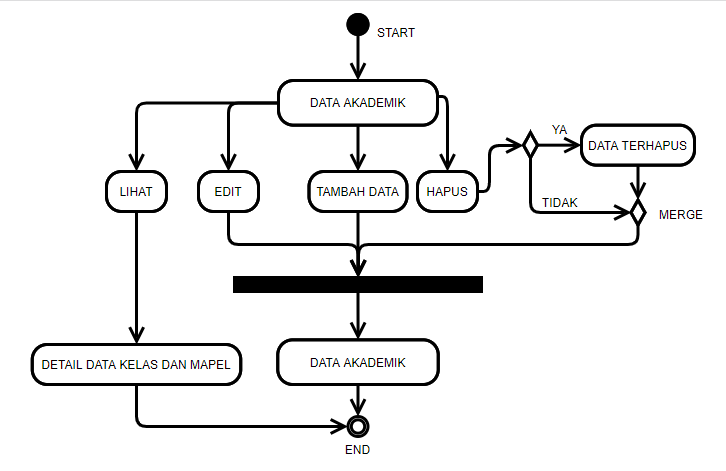
Berikut adalah *Activity Diagram* Kelola Data Siswa, dimana admin dapat kelola data siswa dengan cara, *update*, atau *delete* data siswa ditunjukkan pada Gambar 4.3.



*Gambar 4.3 Activity Diagram Kelola Data Siswa*

* + - 1. ***Activity Diagram* Kelola Data Akademik**

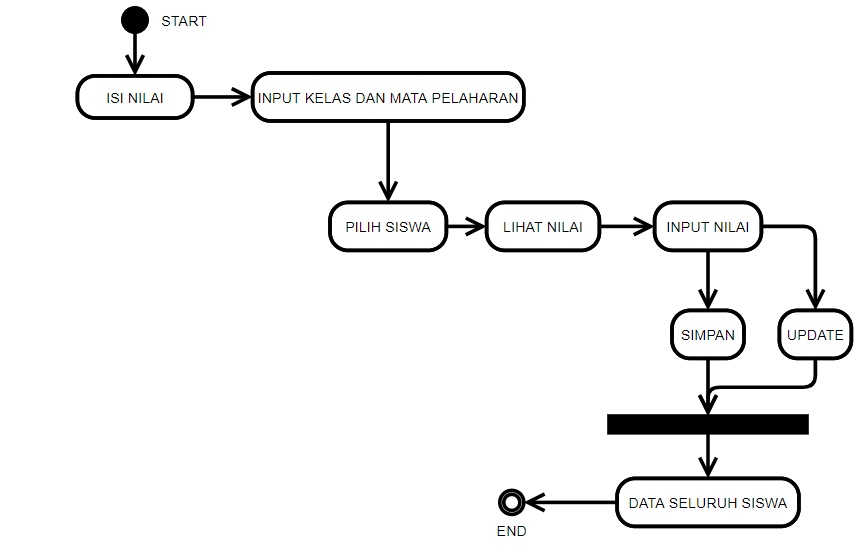
Berikut adalah *Activity Diagram* Kelola Data Akademik dimana Admin dapat mengelola data akademik dengan cara, *update*, atau *delete* data kelas dan mapel ditunjukkan pada Gambar 4.4.



*Gambar 4.4 Activity Diagram Kelola Data Akademik*

* + - 1. **Activity Diagram Kelola Data Nilai**

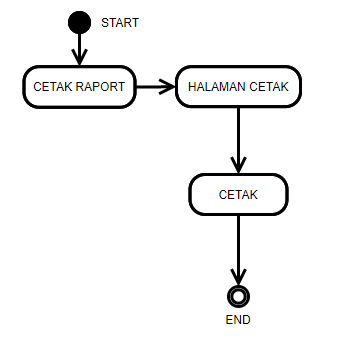
Berikut adalah *Activity Diagram* Kelola Data Nilai, Guru dapat Mengelola Data Nilai Siswa dengan cara, *input*, atau *update* data nilai ditunjukkan pada Gambar 4.5.



*Gambar 4.5 Activity Diagram Kelola Data Nilai*

* + - 1. **Activity Diagram Cetak Raport Siswa**

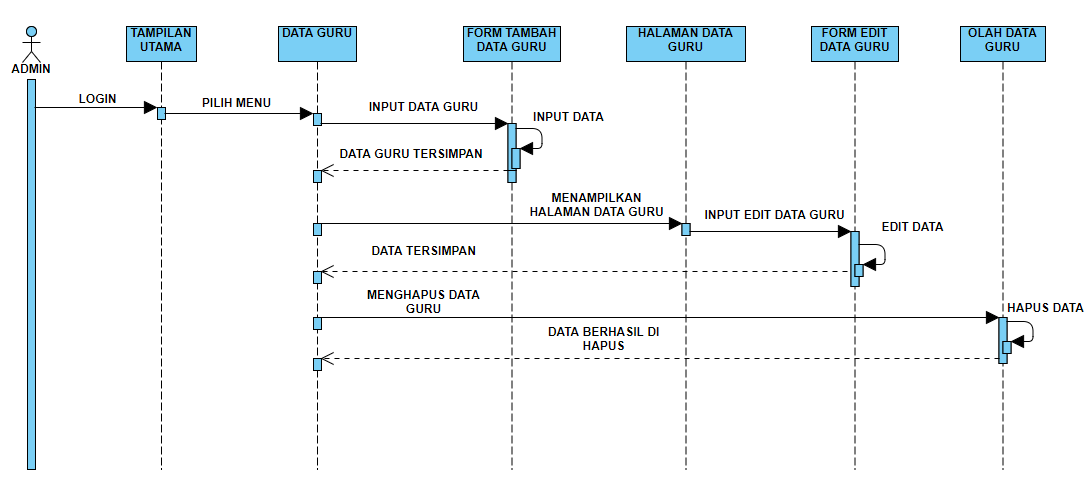
Berikut adalah *Activity Diagram* Cetak Raport Siswa, dimana Guru dan Siswa dapat melakukan Cetak Raport ditunjukkan pada Gambar 4.6.



*Gambar 4.6 Activity Diagram Cetak Raport Siswa*

* + 1. ***Sequence Diagram***
       1. ***Sequence Diagram* Kelola Data Guru**

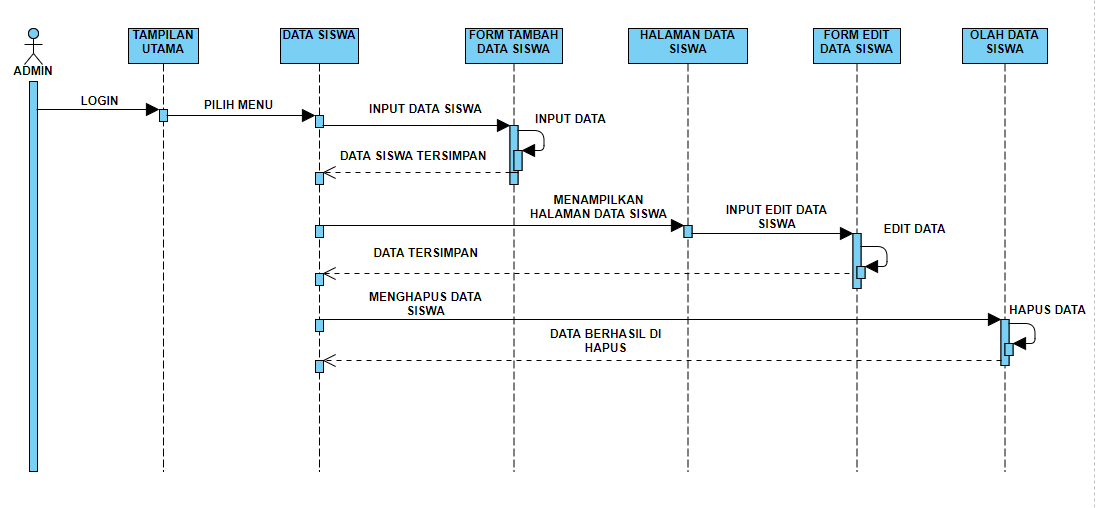
Berikut adalah *Sequence Diagram* Kelola Data Guru, dimana Admin dapat mengelola data guru dengan cara, tambah data, *edit* dan hapus data guru ditunjukkan pada Gambar 4.7.



*Gambar 4.7 Sequence Diagram Kelola Data Guru*

* + - 1. ***Sequence Diagram* KelolaData Siswa**

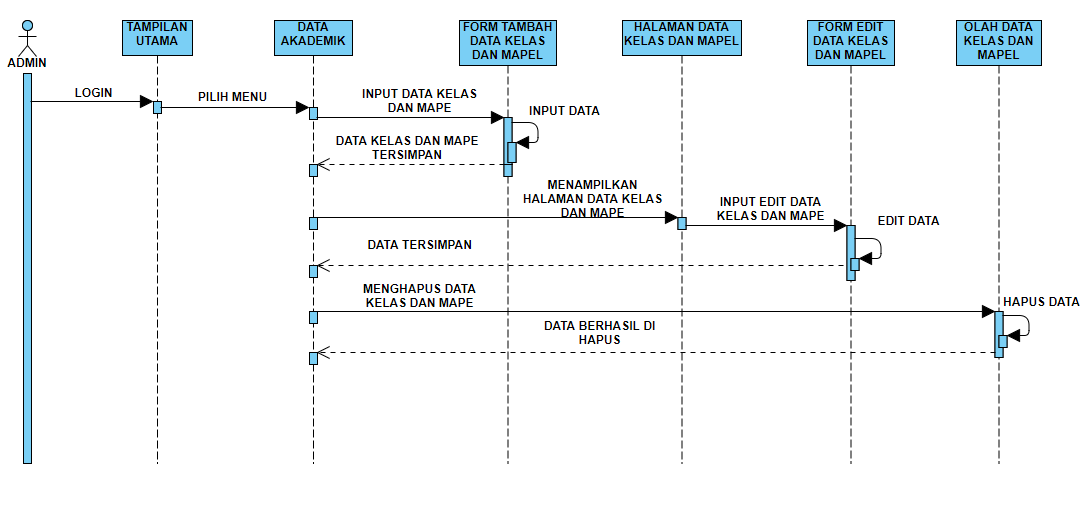
Berikut adalah *Sequence Diagram* Kelola Data Siswa, dimana Admin dapat mengelola data siswa dengan cara, tambah data, *edit* dan hapus data siswa ditunjukkan pada Gambar 4.8.



*Gambar 4.8 Sequence Diagram Kelola Data Siswa*

* + - 1. ***Sequence Diagram* Kelola Data Akademik**

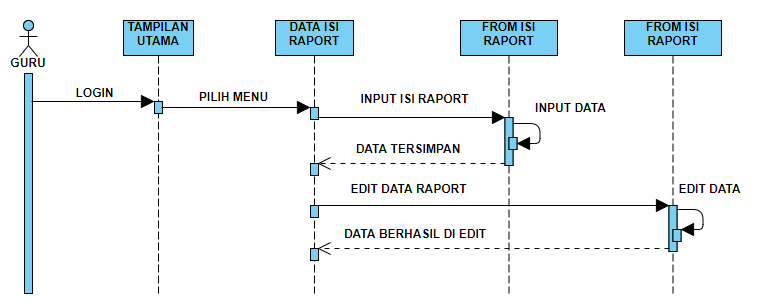
Berikut adalah *Sequence Diagram* Kelola Data Akademik, dimana Pegawai dapat mengelola data kelas dan mapel dengan cara, tambah data, *edit* dan hapus data kelas dan maple ditunjukkan pada Gambar 4.9.



*Gambar 4.9 Sequence Diagram Kelola Data Akademik*

* + - 1. **Sequence Diagram Kelola Data Nilai**

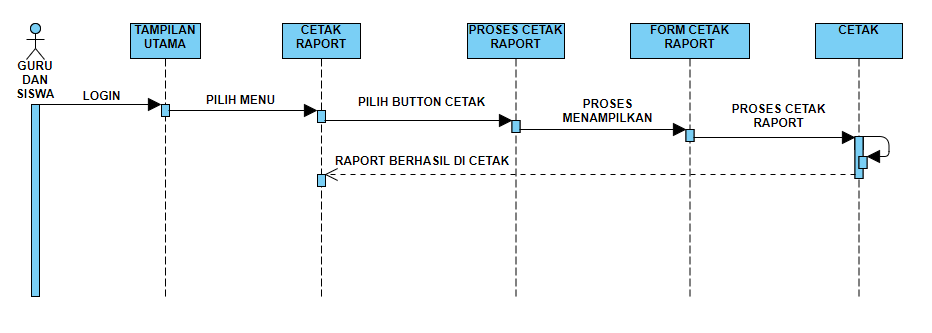
Berikut adalah *Squence Diagram* Kelola Data Nilai, dimana Guru dapat mengelola isi raport dengan cara, tambah datadan hapus isi raport ditunjukkan pada Gambar 4.10.



*Gambar 4.10 Sequence Diagram Kelola Data Nilai*

* + - 1. **Sequence Diagram Cetak Raport Siswa**

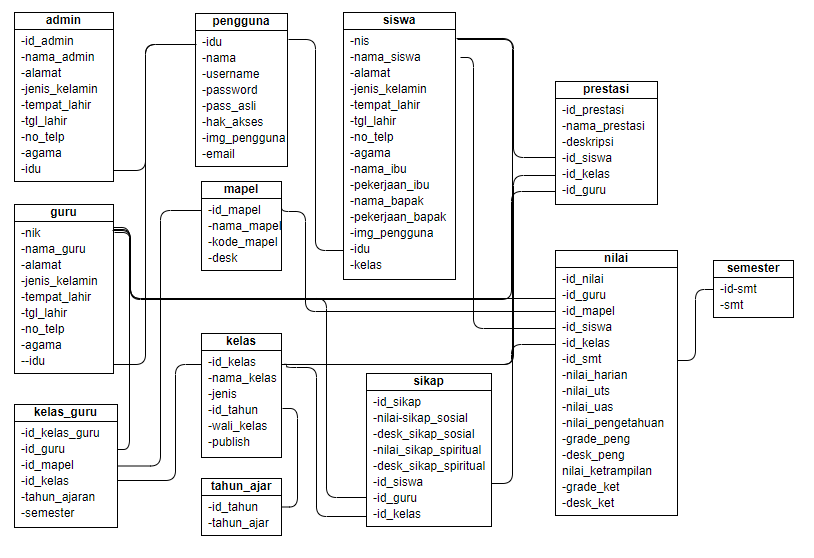
Berikut adalah *Squence Diagram* Cetak Rapor Siswat, dimana Guru dan Siswa dapat melakuakn cetak raport siswa ditunjukkan pada Gambar 4.11.



*Gambar 4.11 Sequence Diagram Cetak Raport Siswa*

* + 1. **Class Diagram**

Berikut adalah *Class Diagram* ditujukkan pada Gambar 4.12.



*Gambar 4.1**2 Class Diagram*

* 1. **Perancangan Struktur Database**

Perancangan database merupakan proses untuk menentukan isi dan pengaturan data yang dibutuhkan untuk mendukung berbagai rancangan sistem dalam aplikasi ini.

* + 1. **Tabel Admin**

*Tabel 4.6 Tabel Admin*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama *Field*** | **Tipe Data (Size)** | **Keterangan** |
| 1. | *Id\_admin* | *Varchar (11)* | *Primary Key* |
| 2. | nama\_admin | *Varchar (50)* |  |
| 3. | Alamat | *Varchar (50)* |  |
| 4. | Jenis\_kelamin | *Enum(laki-laki,perempuan)* |  |
| 5. | Tempat\_lahir | *text* |  |
| 6. | Tgl\_lahir | *date* |  |
| 7. | No\_telp | *Varchar (15)* |  |
| 8. | Agama | *Varchar (100)* |  |
| 9. | idu | *Varchar (30)* | *Foreign Key* |

* + 1. **Tabel Guru**

*Tabel 4.7 Tabel Guru*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama *Field*** | **Tipe Data (Size)** | **Keterangan** |
| 1. | Nik | *Varchar (18)* | *Primary Key* |
| 2. | Nama\_guru | *Varchar (20)* |  |
| 3. | Alamat | *Varchar (50)* |  |
| 4. | Jenis\_kelamin | *Enum(laki-laki,perempuan)* |  |
| 5. | Tempat\_lahir | *text* |  |
| 6 | Tgl\_lahir | *date* |  |
| 7. | No\_telp | *Varchar (15)* |  |
| 8. | Agama | *Varchar (20)* |  |
| 9 | Idu | *Varchar (25)* | *Foreign Key* |

* + 1. **Tabel Kelas**

*Tabel 4.8 Tabel Kelas*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama *Field*** | **Tipe Data (Size)** | **Keterangan** |
| 1. | Id\_kelas | *Int (11)* | *Primary Key* |
| 2. | Nama\_kelas | *Varchar (100)* |  |
| 3. | Jenis | *Int (11)* |  |
| 4. | Id\_tahun | *Int (11)* | *Foreign Key* |
| 5. | Wali\_kelas | *Varchar (100)* |  |
| 6. | publish | *Enum(ya,tidak)* |  |

* + 1. **Tabel kelas\_guru**

*Tabel 4.9 Tabel Kelas\_Guru*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama *Field*** | **Tipe Data (Size)** | **Keterangan** |
| 1. | Id\_kelas\_guru | *Int (11)* | *Primary Key* |
| 2. | Id\_guru | *Int (11)* | *Foreign Key* |
| 3. | Id\_mapel | *Int (11)* | *Foreign Key* |
| 4. | Id\_kelas | *Int (11)* | *Foreign Key* |
| 5. | Tahun\_ajar | *Int (50)* | *Foreign Key* |
| 6 | Semester | *Int (19)* | *Foreign Key* |

* + 1. **Tabel Mapel**

*Tabel 4.10 Tabel Mapel*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama *Field*** | **Tipe Data (Size)** | **Keterangan** |
| 1. | Id\_mapel | *Int (11)* | *Primary Key* |
| 2. | Nama\_mapel | *Varchar (50)* |  |
| 3. | Kode\_mapel | *Varchar (50)* |  |
| 4. | desk | *Varchar (30)* |  |

* + 1. **Tabel Nilai**

*Tabel 4.11 Tabel Nilai*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama *Field*** | **Tipe Data (Size)** | **Keterangan** |
| 1. | Id\_nilai | *Int (11)* | *Primary Key* |
| 2. | Id\_guru | *Varchar (50)* | *Foreign key* |
| 3. | Id\_mapel | *Int (11)* | *Foreign key* |
| 4. | Id\_siswa | *Varchar (50)* | *Foreign key* |
| 5. | Id\_kelas | *Int (11)* | *Foreign key* |
| 6 | Id\_smt | *Int (11)* | *Foreign key* |
| 7. | Nilai\_harian | *double* |  |
| 8. | Nilai\_uts | *double* |  |
| 9. | Nilai\_uas | *double* |  |
| 10. | Nilai\_pengetahuan | *double* |  |
| 11. | Grade\_peng | *Varchar (5)* |  |
| 12. | Desk\_peng | *Varchar (5)* |  |
| 13. | Nilai\_ketrampilan | *double* |  |
| 14. | Grade\_ket | *Varchar (5)* |  |
| 15. | Desk\_ket | *Varchar (5)* |  |

* + 1. **Tabel Pengguna**

*Tabel 4.12 Tabel Pengguna*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama *Field*** | **Tipe Data (Size)** | **Keterangan** |
| 1. | idu | *Varchar (25)* | *Primary Key* |
| 2. | Nama | *Varchar (100)* |  |
| 3. | Username | *Varchar (100)* |  |
| 4. | Password | *Varchar (100)* |  |
| 5. | Pass\_asli | *Varchar (100)* |  |
| 6 | Hak\_akses | *Enum*  *(admin,siswa,guru,walikelas)* |  |
| 7. | Img\_pengguna | *Varchar (100)* |  |
| 8. | email | *Varchar (100)* |  |

* + 1. **Tabel Prestasi**

*Tabel 4.13 Tabel Prestasi*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama *Field*** | **Tipe Data (Size)** | **Keterangan** |
| 1. | Id\_prestasi | *Int (11)* | *Primary Key* |
| 2. | Nama\_prestasi | *Varchar (100)* |  |
| 3. | Desktipsi | *Varchar (100)* |  |
| 4. | Id\_siswa | *Varchar (50)* | *Foreign key* |
| 5. | Id\_kelas | *Int (11)* | *Foreign key* |
| 6 | Id\_guru | *Varchar (50)* | *Foreign key* |

* + 1. **Tabel Semester**

*Tabel 4.14 Tabel Semester*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama *Field*** | **Tipe Data (Size)** | **Keterangan** |
| 1. | Id\_smt | *Int (11)* | *Primary Key* |
| 2. | smt | *Varchar (10)* |  |

* + 1. **Tabel Sikap**

*Tabel 4.15 Tabel Sikap*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama *Field*** | **Tipe Data (Size)** | **Keterangan** |
| 1. | Id\_sikap | *Int (11)* | *Primary Key* |
| 2. | Nilai\_sikap\_sosial | *Varchar (5)* |  |
| 3. | Desk\_sikap\_sosial | *text* |  |
| 4. | Nilai\_sikap\_spiritual | *Varchar (5)* |  |
| 5. | Desk\_sikap\_spiritual | *text* |  |
| 6 | Id\_siswa | *Varchar (50)* | *Foreign key* |
| 7. | Id\_guru | *Varchar (50)* | *Foreign key* |
| 8. | Id\_kelas | *Int (11)* | *Foreign key* |

* + 1. **Tabel Siswa**

*Tabel 4.16 Tabel Siswa*

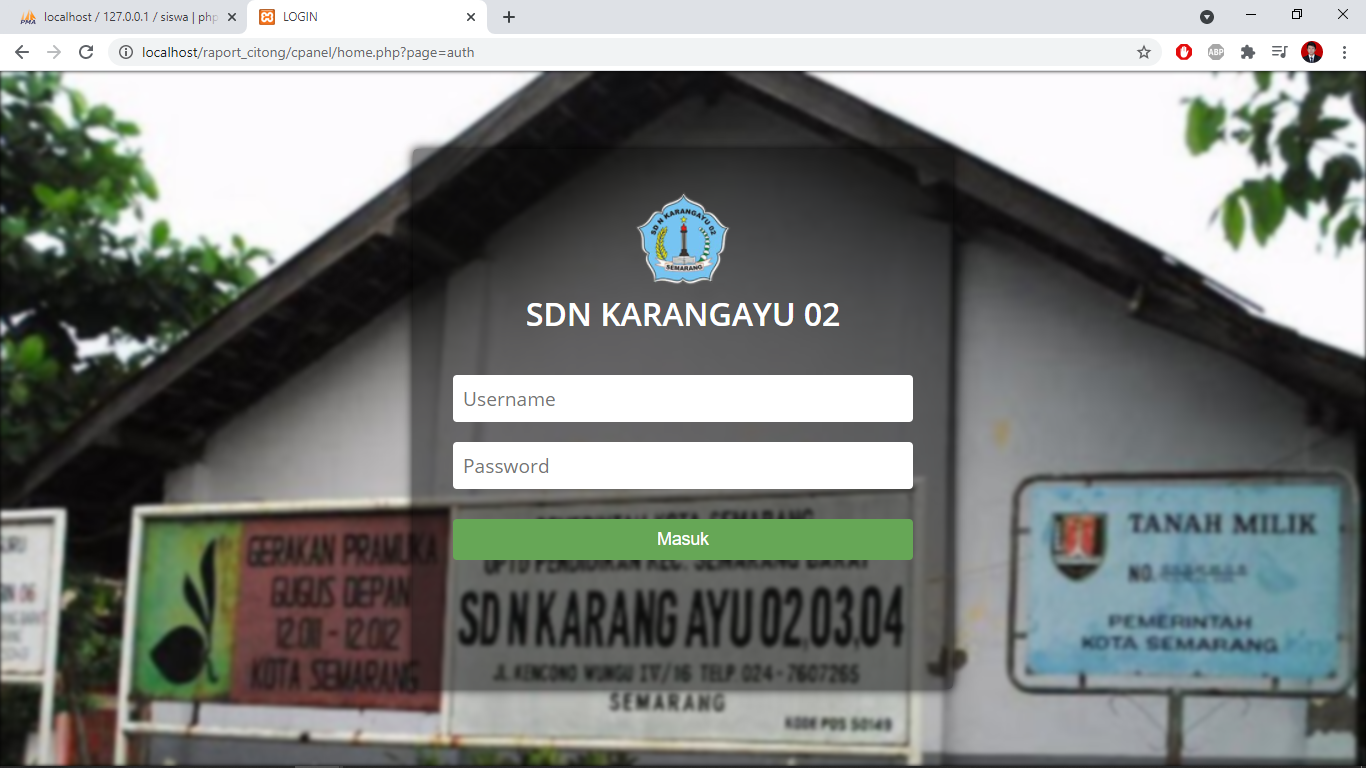
|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama *Field*** | **Tipe Data (Size)** | **Keterangan** |
| 1. | Nis | *Varchar (11)* | *Primary Key* |
| 2. | Nama\_siswa | *Varchar (100)* |  |
| 3. | Alamat | *text* |  |
| 4. | Jenis\_kelamin | *Varchar (50)* |  |
| 5. | Tempat\_lahir | *text* |  |
| 6 | Tgl\_lahir | *date* |  |
| 7. | No\_telp | *Varchar(15)* |  |
| 8. | Agama | *Varchar (100)* |  |
| 9. | Nama\_ibu | *Varchar (100)* |  |
| 10. | Pekerjaan\_ibu | *Varchar (100)* |  |
| 11. | Nama\_bapak | *Varchar (100)* |  |
| 12. | Pekerjaan\_bapak | *Varchar (100)* |  |
| 13. | Img\_pengguna | *Varchar (100)* |  |
| 14. | Idu | *Varchar* (25) | *Foreign key* |
| 15. | kelas | *Int(11)* | *Foreign key* |

* + 1. **Tabel Tahun Ajar**

*Tabel 4.17 Tabel Tahun\_ajar*

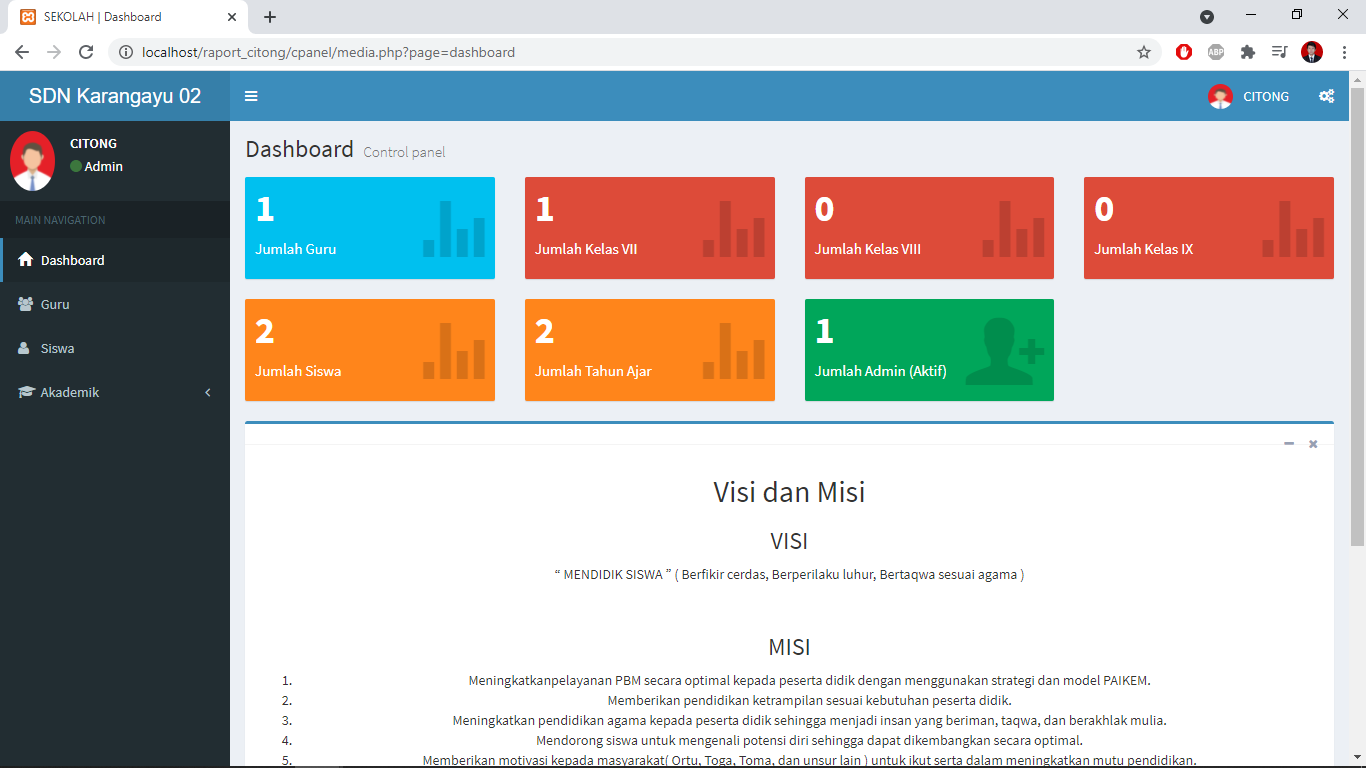
|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama *Field*** | **Tipe Data (Size)** | **Keterangan** |
| 1. | Id\_tahun | *Int (11)* | *Primary Key* |
| 2. | Tahun\_ajar | *Varchar (50)* |  |

* 1. **Perancangan Antar Muka** 
     1. **Desain Tampil Menu Utama**

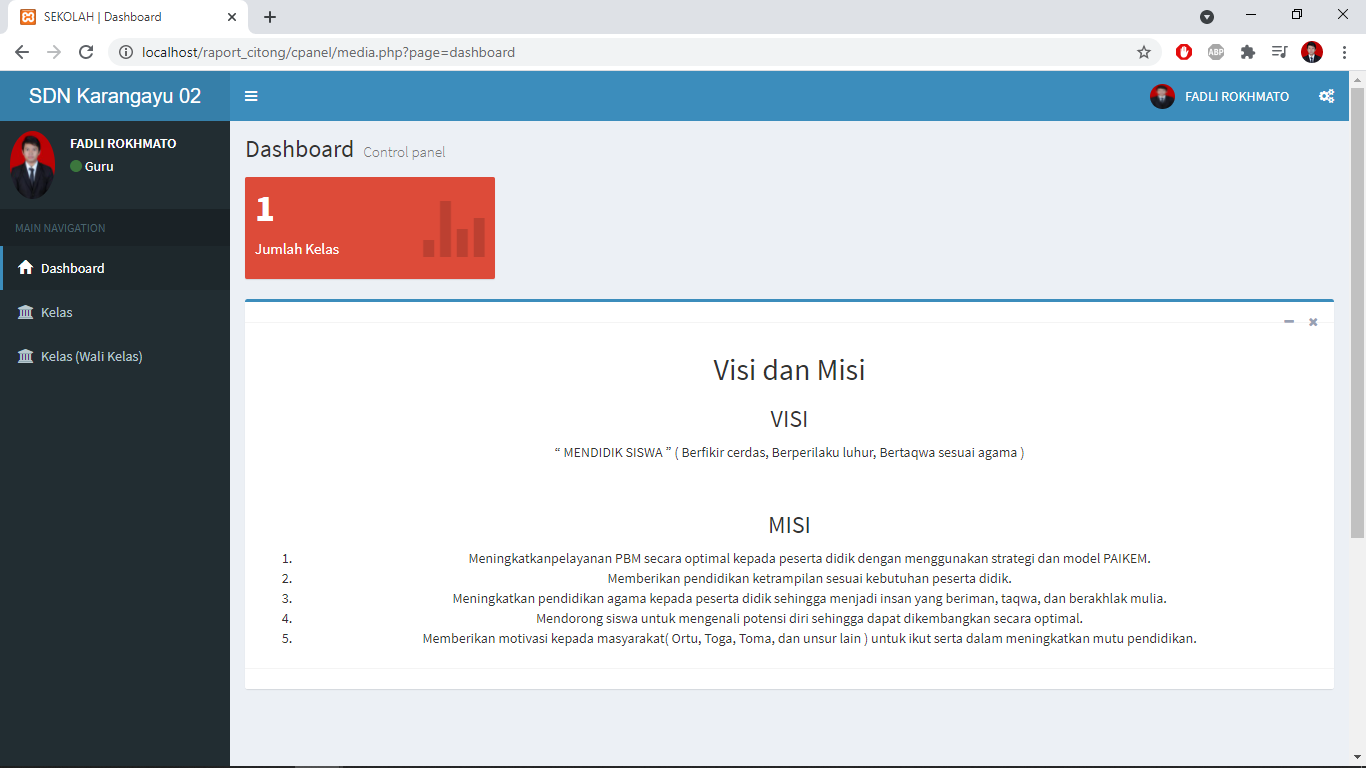


*Gambar 4.13 Desain Tampil Menu Utama*

* + 1. **Desain Tampilan Menu Admin**

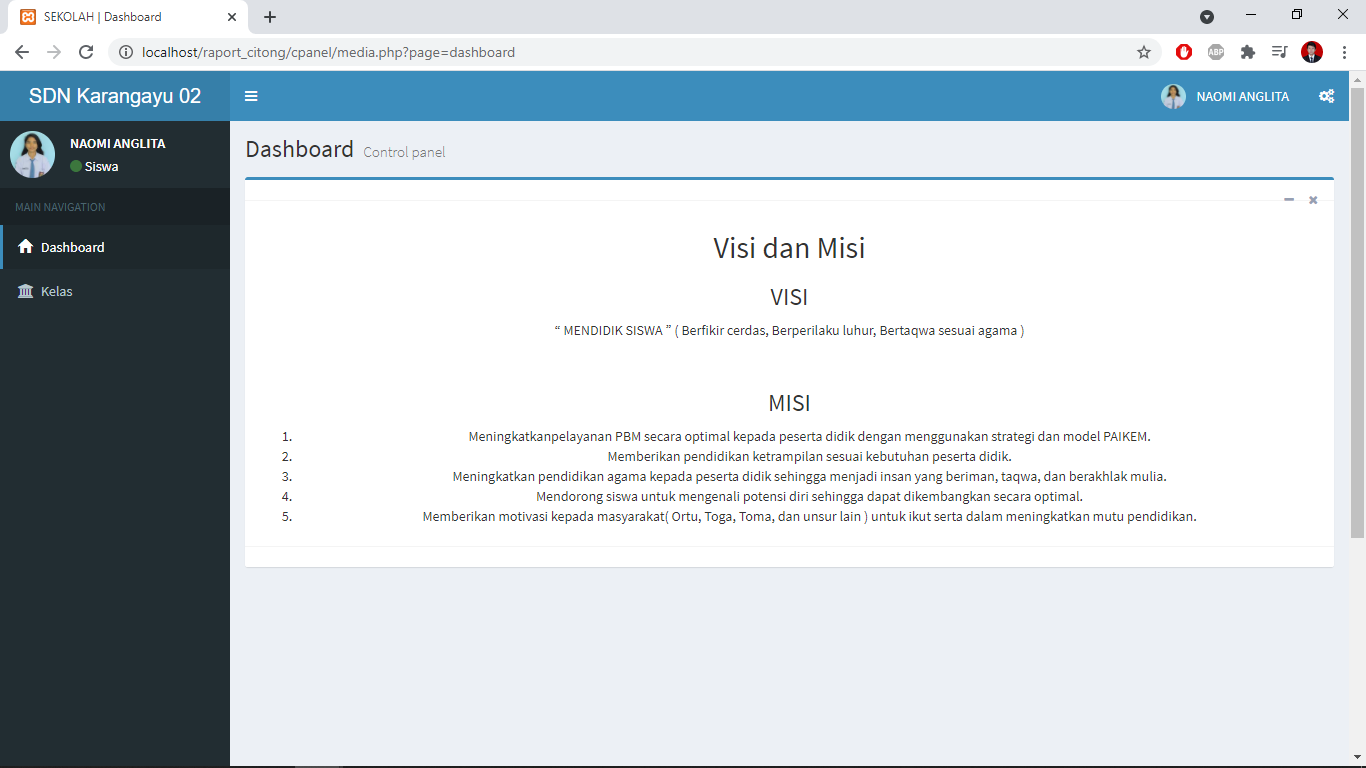
 *Gambar 4.14 Desain Tampil Menu Admin*

* + 1. **Desain Tampilan Menu Guru**



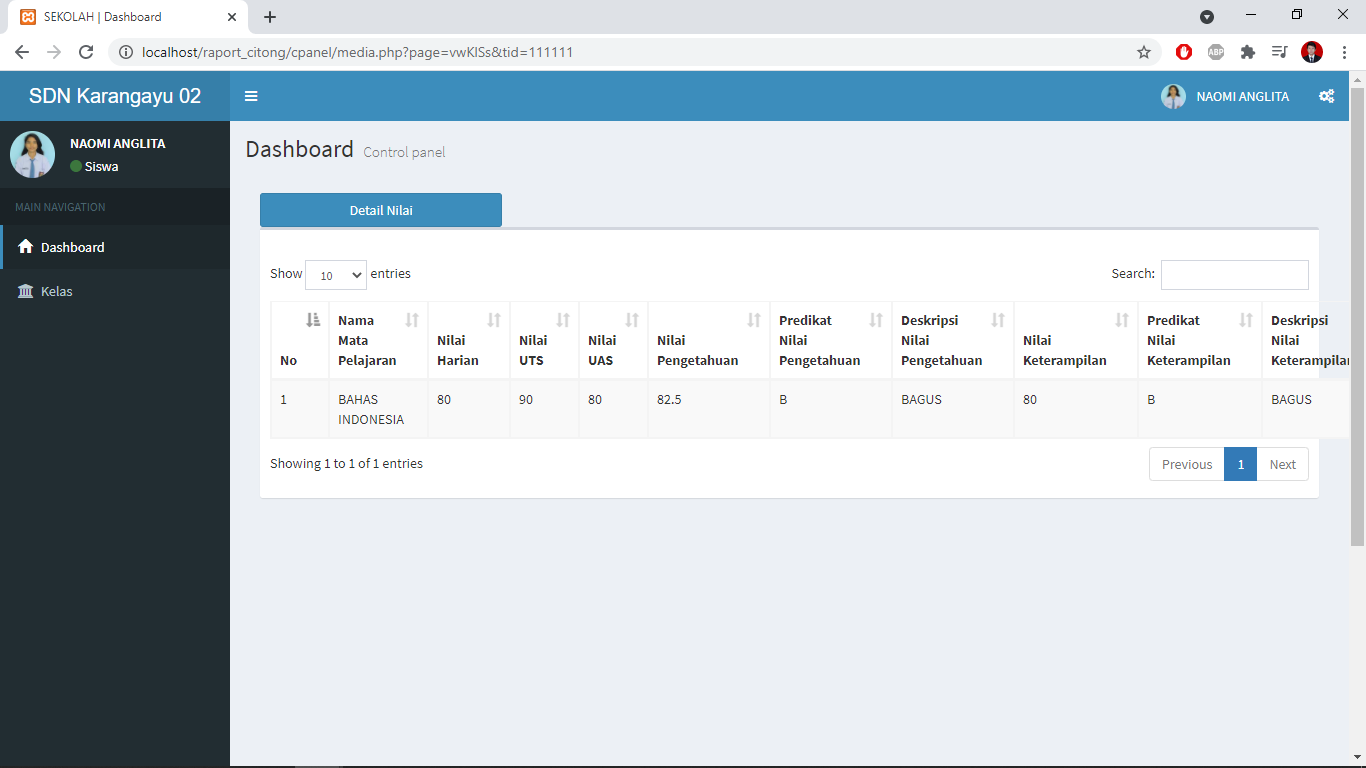
*Gambar 4.15 Desain Tampilan Menu Guru*

* + 1. **Desain Tampilan Menu Siswa**



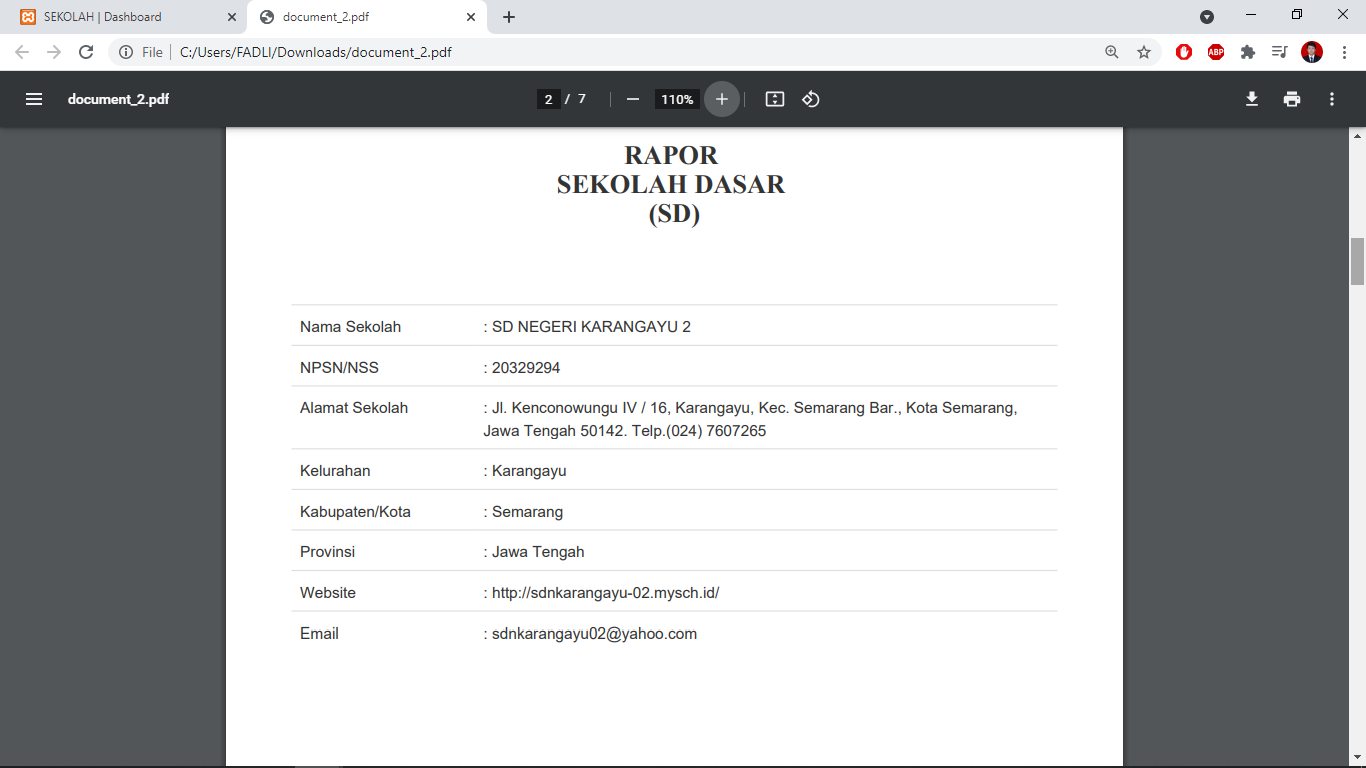
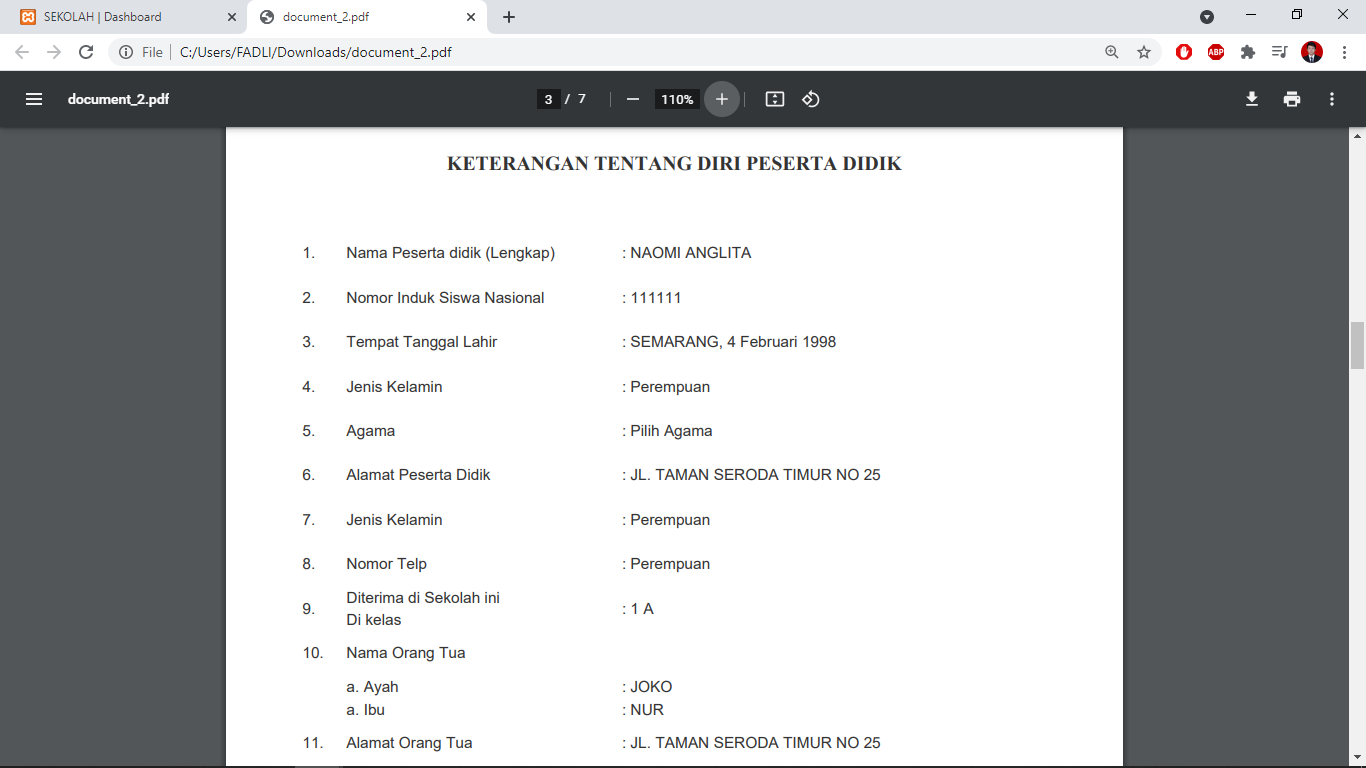
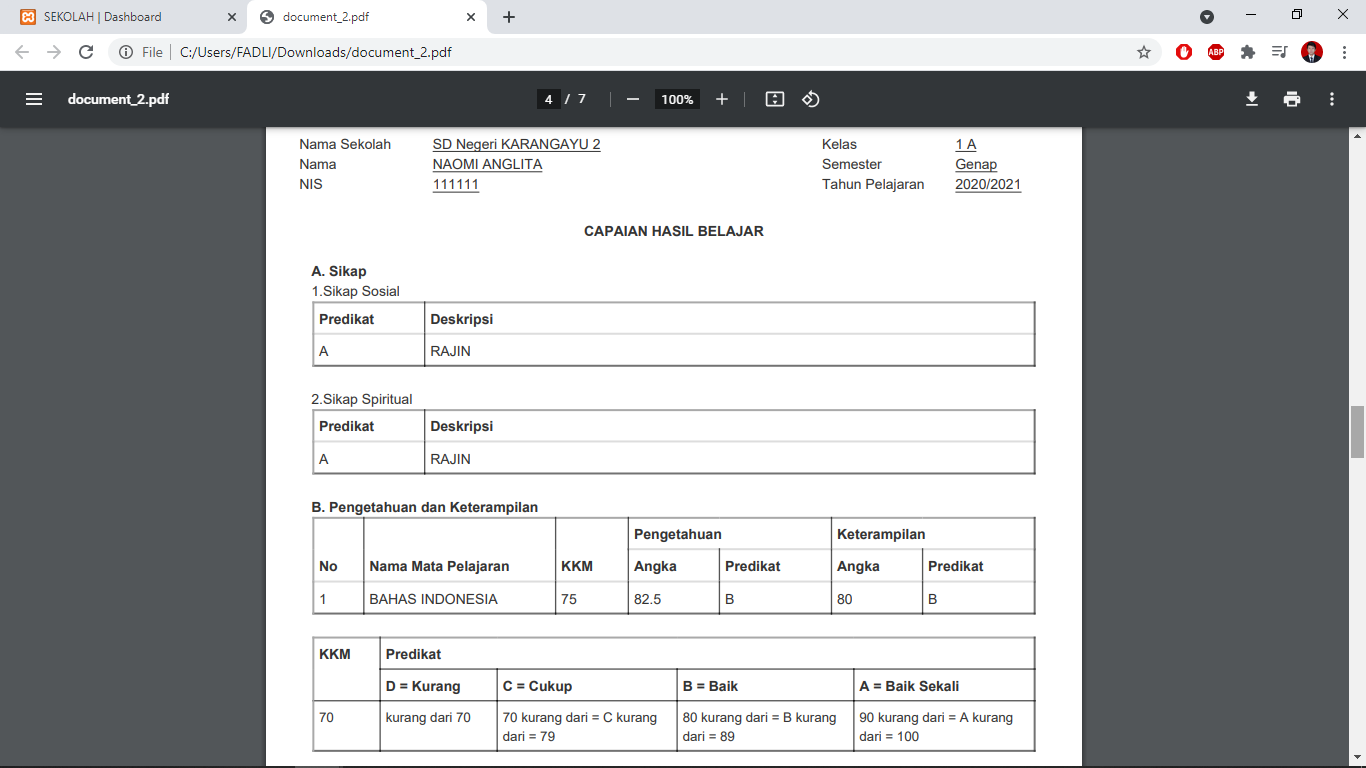
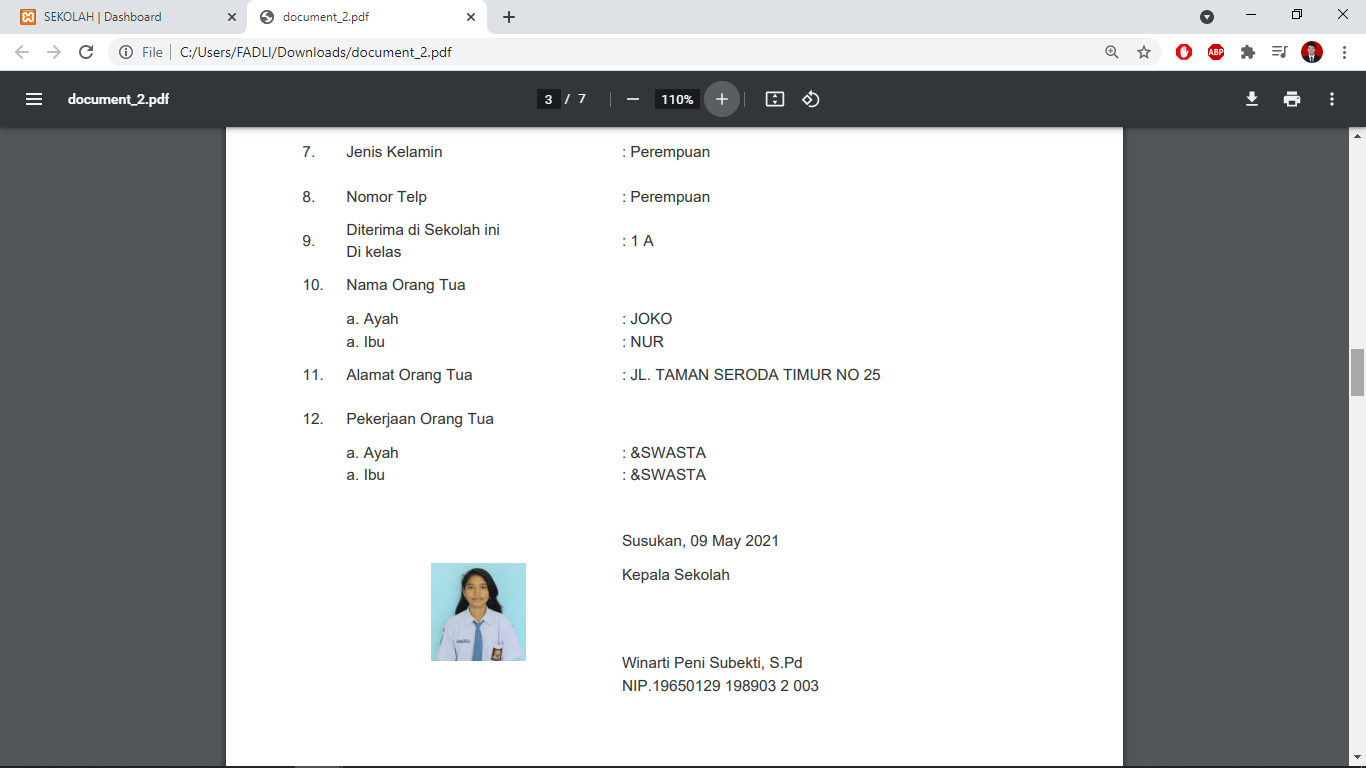
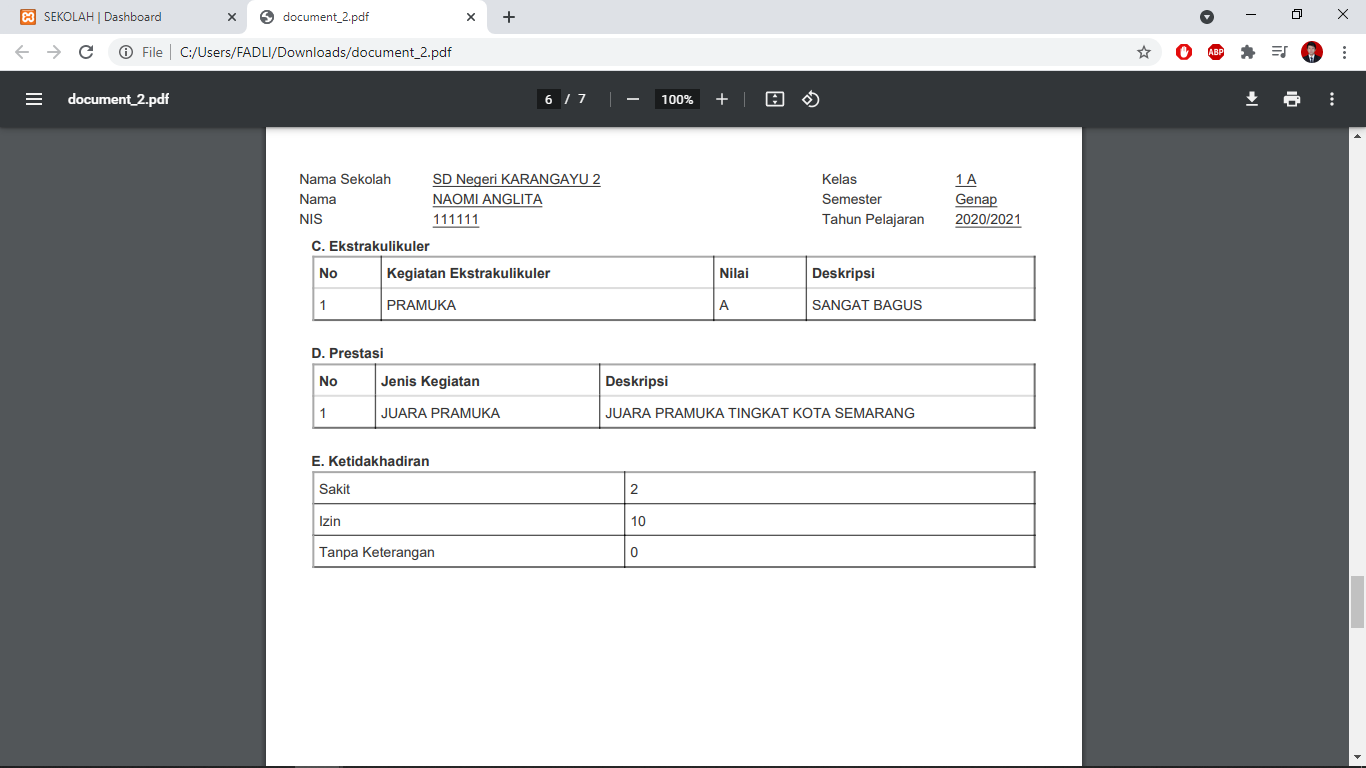
*Gambar 4.16 Desain Tampilan Menu Siswa*

* + 1. **Tampilan Nilai Siswa**



*Gambar 4.17 Tampilan Nilai Siswa*

* + 1. **Tampilan Raport Siswa**

*Gambar 4.18 Tampilan Raport Siswa*

**BAB V**

**PENUTUP**

**5.1. Kesimpulan**

Berdasarkan hasil pembuatan Pengembangan Model Sistem Raport Online SD Negeri Karangayu 02 Semarang, maka penulis dapat mengambil kesimpulan sebagai berikut :

1. Sistem Raport Siswa SD Negeri Karangayu 02 yang telah dirancang untuk memperbaiki kekurangan dari sistem manual yang diterapkan saat ini, dengan mempertimbangkan faktor efisiensi dan ketelitian dalam pencatatan manual.
2. Adanya sistem raport ini mempermudah instansi dalam pengelolaan nilai siswa.
3. Membantu dan memudahkan kinerja guru wali dalam pencatatan nilai raport yang cepat dan efisien.
4. Memudahkan wali murid untuk melihat raport tanpa adanya batasan ruang dan waktu.
5. Dari segi pelayanan raport online yang masih terbilang baru ini masih seringkali terjadi *trouble* dalam mengakses karena jaringan atau *server* yang tidak *connect* dan juga penyajian raport online menurut sebagian orang sulit dimengerti karena harus memasukkan username dan password.

**5.2. Saran**

Berdasarkan kesimpulan yang dikemukakan diatas, maka ada beberapa saran yang penulis sampaikan antara lain sebagai berikut :

1. Untuk pengembangan sistem ini perlu menggunakan sistem keamanan yang lebih rumit agar data yang digunakan tidak mudah di manipulasi.
2. Proses pengambilan data dari *server* perlu dibatasi agar tidak mudah mengakses langsung ke *database.*
3. Dilakukannya sosialisasi secara periodik (per semester) dari sekolah untuk wali siswa mengenai raport online ini terkait mengatasi berbagai macam kendala saat mengakses.
4. Perlu ditambahkan fumgsi untuk memback-up file-file yang ada di sistem, agar jika terjadi kehilangan file-file tersebut masih ada file *back-up* yang tersimpan.
5. Agar ada jaminan konektivitas raport *online* tidak sering *loading,* ada perbaikan *server* agar lebih adaptif dengan *user*

# DAFTAR PUSTAKA

A. S, R., & Shalahuddin, M. (2013). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek.* Bandung: Informatika.

A. S, R., & Shalahuddin, M. (2018). *Reakayasa Perangkat Lunak Terstruktur dan Berorientasi Objek.* Jakarta: INFORMATIKA.

Arief, M. R. (2011). *Pemrograman Web Dinamis menggunakan PHP dan MySQL.* Yogyakarta: C.V ANDI OFFSET.

Budiman, D., Nugraha, D., & Margahayu, S. (2019). Aplikasi Raport Online Berbasis Web Menggunakan Framework Codeigniter. *Jurnal Computer & Bisnis*, 112-121.

Dimyati, & Mudjiono. (2006). *Belajar dan Pembelajaran.* Jakarta: PT Rineke Cipta.

Endra, R., Aprilinda, Y., Dharmawan, Y., & Ramadhan, W. (2002). Analisis Perbandingan Bahasa Pemrograman PHP Laravel dengan PHP Native pada Pengembangan Website. 48-55.

Indarjit. (2001). Sistem Merupakan Satu Kesatuan Prosedur. *Journal of Chemical Information and Modelling*, 1689-1699.

Padmanaba, A., Kumalasari, E., & Andayati, A. (2020). Komparasi Penggunaan Framework Codeigniter Vs PHP Native Pada Sistem Informasi Manajemen Surat Sekretariat DPRD Pemalang. *Jurnal SCRIPT*, 1-6.

Prakasa, C., Mukaromah, S., & Arifiyanti, A. (2020). Aplikasi Raport Online Berbasis Web. 92-98.

Roekminiati, S., Nidn, M., Kamariyah D. S., Nidn, M., Astuti, D., Nidn, M., et al. (2016). Laporan Hasil Penelitian Dosen Program Studi Evaluasi Terhadap Pelayanan Rapor Online di SMP Negeri 12 Surabaya . *Tim Peneliti*.

Wardani, S. (2013). Sistem Informasi Pengolahan Data Nilai Siswa Berbasis Web Pada Sekolah Menengah Atas (SMA) Muhammadiyah Pacitan. *Indonesian Jurnal on Networking and Security*, 2301-5700.

Widianti, S. (2000). *Pengantar Basis Data.* Jakarta: Fajar.

()