

APLIKASI RAPORT ONLINE DI SMP TELKOM BANDUNG

Otto Prayono Santoso Siahaan

6301134091

Jurusan Manajemen Informatika

Fakultas Ilmu Terapan Telkom

University

Jl. Telekomunikasi No.01, Terusan Buah Batu, Bandung, Jawa Barat, 40257

ottoprayono@gmail.com

ABSTRAK

Teknologi Informasi adalah sebuah teknologi yang dipergunakan untuk mengelola data sehingga dapat menghasilkan informasi yang berkualitas dan bernilai guna tinggi. SMP Telkom Bandung merupakan salah satu sekolah menengah pertama yang sudah mengikuti kemajuan teknologi, dimana sekolah ini sudah memiliki sebuah website akademik. Di SMP Telkom Bandung dalam Pengisian nilai dilakukan menggunakan *form* nilai pada selembar kertas sebagai media pengisian nilai sehingga sangat rentang terjadi kehilangan data. Pengisian nilai juga membutuhkan waktu yang cukup lama karena terkadang beberapa guru mata pelajaran tidak tepat waktu dalam memberikan nilai pada wali kelas. Berdasarkan permasalahan tersebut maka dibuat sebuah Aplikasi Rapor Online dimana aplikasi dapat mengolah data nilai siswa, menyajikan informasi mengenai data nilai dan data siswa. Aplikasi ini digunakan 4 pengguna yaitu bagian kurikulum, guru, wali kelas dan kepala sekolah. Metode yang digunakan dalam pembuatan aplikasi ini adalah metode *prototype*. Bahasa pemrograman yang digunakan adalah PHP dengan *framework codeigniter* dan *database* yang digunakan adalah MySQL. Aplikasi ini dibangun untuk membantu guru matapelajaran dalam melakukan pengisian nilai siswa dan aplikasi ini juga dapat membantu wali kelas dan kepala sekolah dalam melihat perkembangan nilai siswa.

Kata Kunci : Teknologi Informasi, Nilai, Aplikasi

ABSTRACT

Information technology is a technology that is used to manage the data so that it can produce high quality information and high use value. SMP Telkom Bandung is one of the first secondary school that has followed the progress of technology, in which the school already has an academic website. In secondary school, Telkom Bandung in value Charging is done using form values on a piece of paper as a medium for charging value so highly the range of data loss occurs. Charging value also takes quite a long time because sometimes some subject teachers are not timely in delivering value to the homeroom. Based on the problems it created an Online Report Application where applications can process data value of students, provide information about the value of data and student data. This application is used 4 user that is part of the curriculum, teachers, homeroom teacher and principals. The method used in this app maker is the method prototype. The programming language used is PHP with CodeIgniter framework and database used is MySQL. This app was built to help teachers lesson

in charging the students 'grades and these applications can also help homeroom and principals to see the development of students' grades.

Keywords: Information Technology, Value, Applications

1. PENDAHULUAN

1.1 Latar Belakang Masalah

Teknologi Informasi adalah sebuah teknologi yang dipergunakan untuk mengelola data sehingga dapat menghasilkan informasi yang berkualitas dan bernilai guna tinggi. Perkembangan Teknologi Informasi dan Komunikasi (TIK) pun terus meningkat seiring dengan meningkatnya kebutuhan manusia, termasuk dalam bidang pendidikan. Perkembangan ini dapat dilihat dari pemakaian sistem informasi akademik sekolah yang berbasis *website*. SMP Telkom Bandung merupakan salah satu sekolah menengah pertama yang sudah mengikuti kemajuan teknologi, dimana sekolah ini sudah memiliki sebuah *website* akademik, tetapi di SMP Telkom Bandung dalam hal pencatatan nilai masih dilakukan secara manual. Maka dari itu, perlu dilakukan pengembangan sistem pengolahan data nilai di SMP Telkom Bandung agar dapat menggunakan teknologi juga.

Sistem pembuatan atau penilaian rapor di SMP Telkom Bandung masih bersifat manual atau belum ada sistem komputer dalam pengolahan data nilai. Data nilai rapor siswa hanya tertulis pada buku rapor dan di bagikan ke siswa atau wali murid. Pembuatan data tersebut dapat mengakibatkan kehilangan data dan data yang ingin dicari sulit untuk ditemukan. Selain itu untuk menulis dan mendata nilai siswa membutuhkan waktu yang cukup lama.

Berdasarkan permasalahan tersebut diperlukan suatu aplikasi pengolahan data nilai siswa yang dapat menyajikan informasi mengenai data nilai secara tepat. Aplikasi ini nantinya dapat melakukan pengolahan data dan dapat diakses dimana saja dan kapan saja. Pada aplikasi ini nantinya penilaian yang dilakukan akan

disesuaikan dengan kurikulum 2013. Aplikasi ini dapat melakukan cetak laporan hasil nilai siswa oleh wali kelas.

Berdasarkan uraian diatas dapat disimpulkan bahwa pembuatan aplikasi rapor *online* SMP Telkom Bandung sangat diperlukan untuk media dalam memantau nilai para siswa dalam pendidikan.

1.2 Tujuan

Adapun tujuan yang ingin dicapai pada tugas akhir ini adalah membangun aplikasi yang :

- a. Dapat membantu pihak sekolah dalam mengolah data nilai siswa sehingga dapat meminimalisir kesalahan dalam memasukkan data dan menghitung nilai siswa.
- b. Memiliki fitur mencetak raport siswa secara *online*.

1.3 Rumusan Masalah

Dari latar belakang diatas maka dapat diambil perumusan masalah, yaitu :

- a. Bagaimana membantu pihak sekolah khususnya wali kelas dan guru dalam mengolah data nilai siswa?
- b. Bagaimana wali kelas dapat mencetak rapor dan hasil nilai siswa secara *online*?

1.4 Batasan Masalah

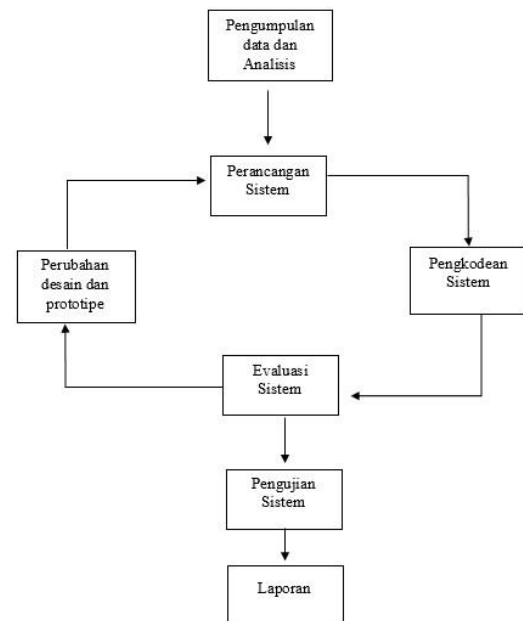
Batasan masalah dapat berisi:

1. Data nilai siswa yang ditampilkan mencakup nilai ulangan, nilai Rata-Rata harian, nilai ulangan tengah semester, nilai Rata-Rata Tugas, nilai akhir semester, nilai keterampilan, nilai sikap dan total nilai akhir siswa.
2. Tidak membahas absensi siswa.

3. Hak akses aplikasi ini yaitu admin, wali kelas, guru mata pelajaran dan kepala sekolah. Hak akses yang diberikan berbeda-beda setiap *user*.
4. Studi kasus hanya kelas VII.
5. Matapelajaran yang diberikan di kelas VII Bahasa Indonesia, Bahasa Inggris, Matematika, IPA, Agama, IPS, Bahasa Sunda, Teknologi Infomasi Komunikasi (TIK), Pendidikan Kesehatan dan Jasmani (PENJAS) dan Pendidikan Kewarganegaraan (PKN).
6. Jumlah guru yang mengajar di kelas VII ada sebanyak 13 orang dan jumlah mata pelajaran yang diajarkan ada 10.
7. Nilai keterampilan yang ditampilkan hanya berupa nilai praktek siswa dan nilai sikap diberikan berdasarkan proses belajar mengajar.
8. Aplikasi ini tidak menangani penjadwalan mata pelajaran.
9. Pembagian kelas dan pembagian guru mata pelajaran setiap kelas di luar sistem.

1.5 Metode Pengerjaan

Dalam penyusunan Proyek Akhir yang berjudul “ Aplikasi *Raport Online* “, metode yang dipakai adalah metode *prototyping* yaitu pengembangan dan pelanggan dapat saling berinteraksi selama proses pembuatan sistem. (Simarmata, Janner, Rekayasa Perangkat Lunak 2010) Berikut adalah tahap pengerjaannya :



Gambar 1

a) Pengumpulan data

Dalam pembuatan aplikasi rapor *online* akan dilakukan pengumpulan data. Pengumpulan data dilakukan untuk mengumpulkan informasi data yang dibutuhkan. Pengumpulan data dilakukan dengan melakukan wawancara kepada bagian kurikulum, kepala sekolah dan guru – guru. Selain melakukan wawancara dilakukan pengumpulan data siswa, wali kelas, guru dan *form* nilai atau data nilai siswa.

b) Perancangan sistem

Setelah melakukan pengumpulan data, maka tahap selanjutnya adalah membuat perancangan sistem. Untuk menggambarkan pemahaman proses bisnis *tools* yang digunakan dalam perancangan sistem adalah *use case diagram* dan teknik perancangan menggunakan UML.

c) Pengkodean sistem

Dalam pengkodean menggunakan MySQL sebagai databasanya dan menggunakan *framework codeigneter*.

d) Evaluasi sistem

Setelah pembuatan perancangan sistem dibuat maka akan diberikan kepada *client* untuk mengetahui

apakah ada perubahan atau penambahan pada rancangan sistem yang dibuat dan jika sudah sesuai maka akan melangkah ke tahap selanjutnya.

e) **Perubahan Desain dan Prototipe**

Apabila terjadi kekurangan, maka pada tahap ini akan direvisi kembali dan mengulangi langkah sebelumnya.

f) **Pengujian Sistem**

Pada tahap ini dilakukan uji coba sistem yang telah dirancang untuk memastikan bahwa sistem yang dibangun dapat digunakan dengan baik dan benar sesuai dengan yang diharapkan. Pengujian ini dilakukan dengan *black box testing*.

g) **Pembuatan laporan**

Setelah semua program berjalan dengan baik dan tidak ada kesalahan maka dibuat sebuah laporan untuk dijadikan acuan untuk aplikasi selanjutnya.

2. LANDASAN TEORI

1.1 Metode Prototype

Sebelum melakukan metode *prototype* dilakukan dari tahap komunikasi. Setelah dari tahap komunikasi maka akan dilanjutkan ke tahap pengumpulan data dan analisis. Pada tahap ini bisa dilakukan dengan wawancara pengguna atau *client*. Selanjutnya akan dilakukan perancangan sistem kemudian dilanjutkan ke tahap pengkodean sistem. Setelah dilakukan pengkodean sistem maka akan dilanjutkan ke tahap evaluasi sistem. Pada tahap ini dilakukan untuk mengetahui apakah sistem dan model yang dibuat sudah sesuai dengan keinginan *client*, setelah sudah sesuai akan dilanjutkan ke tahap pengujian sistem jika sistem atau model yang dibuat tidak sesuai maka akan lanjut ke tahap perubahan desain dan protitipe dimana akan kembali ke tahap perancangan sistem dan pengkodean sistem. Setelah dilakukan pengujian sistem maka akan lanjut ke tahap pembuatan laporan. [8].

a) **Pengumpulan data**

Sumber pengumpulan data terdiri dari data primer dan data sekunder, dimana sumber pengumpulan data primer dihasilkan dari :

1. *Observasi* yaitu melakukan peninjauan langsung ke SMP Telkom Bandung.
2. *Interview* yaitu dengan mengajukan pertanyaan secara langsung kepada kepala sekolah dan bagian kurikulum SMP Telkom Bandung mengenai objek yang sedang diteliti sedangkan sumber pengumpulan data sekunder dihasilkan dari :

1. Dokumen Sekolah yaitu pengumpulan data yang dilakukan dengan mempelajari dokumen-dokumen yang ada untuk memperoleh data dan informasi dalam penelitian ini.
2. Studi Pustaka yaitu mempelajari masalah-masalah yang ada pada SMP Telkom Bandung melakukan riset dengan merujuk pada buku-buku yang berhubungan dengan judul tugas akhir ini, guna untuk memecahkan masalah-masalah yang ada pada SMP Telkom Bandung.

b) **Perancangan sistem**

Membangun sistem dengan membuat perancangan sementara yang berfokus kepada penyajian kebutuhan aplikasi. Dan untuk menggambarkan pemahaman proses bisnis berjalan menggunakan *flowmap* sedangkan *tools* yang digunakan *use case*

diagram dan teknik perancangan antar muka menggunakan UML.

c) Pengkodean sistem

Setelah dilakukan perancangan sistem maka dilakukan dengan pembuatan kode program dengan database menggunakan *mySQL* dan menggunakan *framework codeigneter*.

d) Evaluasi sistem

Setelah dilakukan pengkodean dan perancangan sistem maka dilakukan evaluasi untuk mengetahui apakah sistem yang sudah sesuai dengan apa yang diharapkan. Jika sudah sesuai maka akan melangkah ke tahapan pengujian sistem apabila belum sesuai maka akan kembali ke tahap perubahan desain dan *prototype*.

e) Perubahan Desain dan Prototipe

Pada tahap ini melakukan perbaikan sesuai dengan kekurangan pada tahap evaluasi sistem. Pada tahap ini akan direvisi kembali dan mengulangi langkah perancangan sistem dan pengkodean sistem.

f) Pengujian Sistem

Pada tahap ini dilakukan uji coba sistem setelah selesai dari tahap evaluasi sistem. Uji coba sistem dilakukan untuk memastikan bahwa sistem yang dibangun dapat digunakan dengan baik dan benar sesuai dengan yang diharapkan. Pengujian ini dilakukan dengan *black box testing*.

g) Pembuatan laporan


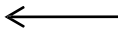




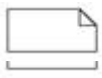
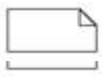
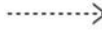

Setelah semua program berjalan dengan baik dan tidak ada kesalahan maka dibuat sebuah laporan untuk dijadikan acuan untuk aplikasi selanjutnya.

1.2 Usecase Diagram

Diagram *use case* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Sebuah *use case* merepresentasikan sebuah interaksi antara *actor* dengan sistem. Seorang *actor* adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem

untuk melakukan pekerjaan-pekerjaan tertentu. [6]

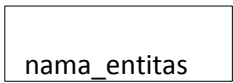
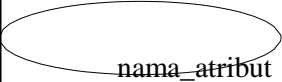
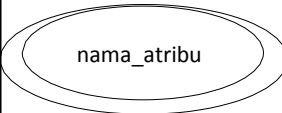

Tabel 1
Simbo – Simbol Usecase [7]

| N O | GAMBAR | NAMA | KETERANGAN |
|-----|--|-----------------------|--|
| 1 |  | <i>Actor</i> | Orang, proses, atau aplikasi lain yang berinteraksi dengan aplikasi informasi yang akan dibuat diluar aplikasi informasi yang akan dibuat itu sendiri. |
| 2 |  | <i>Generalization</i> | Hubungan generalisasi dan spesialisasi antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. |
| 3 |  | <i>Include</i> | Fungsi atau syarat yang dijalankan <i>use case</i> ini. |
| 4 |  | <i>Extend</i> | Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan. |
| 5 |  | <i>Association</i> | Komunikasi antara <i>actor</i> dan <i>use case</i> yang berpartisipasi pada <i>use case</i> . |
| 6 |  | <i>UseCase</i> | Fungsionalitas yang disediakan aplikasi sebagai unit-unit yang saling bertukar pesan antar unit atau <i>actor</i> . |
| 7 |  | <i>System</i> | Menspesifikasikan paket yang menampilkan aplikasi secara terbatas. |
| 8 |  | <i>Note</i> | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi. |
| 9 |  | <i>Dependency</i> | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>). |
| 10 |  | <i>Collaboration</i> | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |

1.3 Entity Relationship Diagram (ERD)

ERD (*Entity Relationship Diagram*) merupakan sebuah diagram yang digunakan untuk merancang hubungan antar tabel-tabel dalam basis data. Berikut adalah simbol-simbol yang sering digunakan pada ERD untuk merancang tabel-tabel beserta relasinya dalam basis data. [9]

Tabel 2
Simbol Simbo ERD

| Simbol | Deskripsi |
|---|--|
| Entitas / <i>entitiy</i>  | Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data |
| Atribut  | <i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas |
| Atribut kunci primer <u>nama_kunci_primer</u> | <i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id |
| Atribut multivalai/ <i>mutivalue</i>  | <i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu |
| Relasi  | Relasi yang menghubungkan antara entitas; biasanya diawali dengan kata kerja |
| Asosiasi / <i>association</i> I 0..* | Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian |


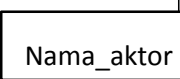
1.4 Sequence Diagram




Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antara objek. Oleh karena itu untuk menggambar *sequence diagram* maka harus diketahui objek – objek yang terlibat dalam sebuah *use case* beserta metode – metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.



Banyaknya *sequence diagram* yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan

interaksi jalannya pesan sudah dicakup pada *sequence diagram* sehingga semakin banyak *use case* yang didefinisikan maka *sequence diagram* yang harus dibuat juga semakin banyak. [10]

Tabel 3

| | | |
|---|---|---|
| 1 | Aktor  Atau  Tanpa waktu aktif | Orang, proses, atau sistem yang berinteraksi dengan sistem informasi yang akan dibuat di informasi yang dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor. |
|---|---|---|

| | | |
|---|--|---|
| 2 | Garis hidup/ <i>lifeline</i> | Menyatakan kehidupan suatu objek. |
| 3 | Objek Nama_objek : nama_kelas | Menyatakan objek yang berinteraksi dengan objek lain. |
| 4 | Waktu aktif  | Menyatakan objek dalam keadaan aktif dan berinteraksi pesan. |
| 5 | Pesan tipe <i>create</i> «create»  | Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat. |
| 6 | Pesan tipe <i>call</i> 1 : nama_method() | Menyatakan suatu objek memanggil operasi/ metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi/ metode, karena ini memanggil operasi/ metode yang dipanggil harus ada pada diagram kelas sesuai dengan objek yang berinteraksi. |
| 7 | Pesan tipe <i>send</i> 1 : masukan  | Menyatakan bahwa suatu objek mengirimkan data/ masukan/ |

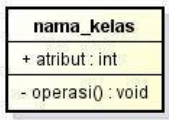






| | | |
|---|--|---|
| | | informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim. |
| 8 | Pesan tipe <i>return</i> 1 : keluaran  | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian. |
| 9 | Pesan tipe <i>destroy</i> «destroy»  | Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy. |

1.5 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur dari segi pendefinisian kelas-kelas yang dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi [11].

1. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Tabel 4
Simbol Simbol Class Diagram

| Simbol | Deskripsi |
|---|---|
|  | Kelas pada struktur sistem |
|  | Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek |
|  | Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> |
|  | Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> |
|  | Relasi antarkelas dengan makna generalisasi-generalisasi (umum khusus) |
|  | Kebergantungan antarkelas |
|  | Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>) |

1.6 Codeigniter

CodeIgniter (selanjutnya disebut CI) adalah sebuah *framework* yang digunakan untuk

membuat sebuah aplikasi berbasis *web* yang disusun dengan menggunakan bahasa PHP. Di dalam CI ini terdapat beberapa macam kelas yang berbentuk *library* dan *helper* yang berfungsi untuk membantu pemrogram dalam mengembangkan aplikasinya. CI sangat mudah dipelajari oleh seorang pemrogram *web* pemula sekalipun. Alasannya, Karena CI mempunyai file dokumentasi yang sangat memadai untuk menjelaskan setiap fungsi yang ada pada *library* dan *helper*. File dokumentasi ini disertakan secara langsung pada saat anda mengunduh paket *framework* CI. [12]

1.7 Javascript

JavaScript merupakan bahasa *script* populer yang dipakai untuk menciptakan halaman *web* yang dapat berinteraksi dengan pengguna dan dapat merespon *event* yang terjadi pada halaman. *JavaScript* dapat digunakan untuk mengendalikan tampilan halaman. *JavaScript* dipakai untuk melakukan validasi apa yang diketikkan pengguna ke dalam sebuah *form* sebelum pengiriman *form* ke *server* dilakukan. *JavaScript* berperan sebagai bahasa pemrograman, yang memiliki konstruksi-konstruksi dasar seperti variabel dan tipe data, *look* kontrol, *statemen if/else*, *statemen switch*, fungsi, dan objek. *JavaScript* dapat dipakai untuk perhitungan aritmatik, pemanipulasian tanggal dan waktu, pemodifikasian *array*, *string* dan objek [1].

JavaScript adalah bahasa yang digunakan untuk membuat program yang digunakan agar dokumen HTML yang ditampilkan dalam browser menjadi lebih interaktif. *JavaScript* merupakan bahasa *script*, bahasa yang tidak memerlukan kompiler untuk menjalankannya, cukup dengan *interpreter*. Program *JavaScript* disisipkan ke dalam dokumen HTML dengan ditandai dengan *tag* yang diawali dengan `<script ... >` dan diakhiri dengan `</script>`. *JavaScript* dapat membuat *cookies* (suatu kode yang digunakan menyimpan dan mengambil data atau informasi dari *browser* pengunjung), membuat aplikasi *client/server*, melakukan validasi data, memberikan reaksi kepada suatu kejadian di atas halaman *web* yang sedang ditampilkan [2].

1.8 Hypertext Preprocessor (PHP)

PHP merupakan bahasa pemrograman berbasis *web* yang memiliki kemampuan untuk

melakukan proses dan mengolah data secara dinamis. PHP bersifat *open source* dan berada dibawah lisensi *GNU Public License(GPL)*. PHP dapat dikatakan sebagai sebuah *server-side embedded script language* artinya semua sintaks dan perintah program yang ditulis akan sepenuhnya dijalankan oleh server, tetapi dapat disertakan pada halaman HTML biasa. Kode PHP diawali dengan sintaks “<?” dan diakhiri dengan sintaks “>”. PHP memiliki kelebihan yaitu cara koneksi dan *query* database yang sederhana, mudah digunakan dan biaya yang dibutuhkan tidak terlalu mahal dan bahkan gratis, *security system* yang cukup tinggi, dan waktu eksekusi yang lebih cepat. PHP memiliki kelemahan yang utamanya yaitu tidak adanya *namespace*. *Namespace* adalah sebuah cara untuk mengelompokkan fungsi atau nama variabel dalam susunan hierarki [3].

PHP adalah salah satu bahasa pemrograman *script* yang dirancang untuk membangun aplikasi *web*. Program yang ditulis dengan PHP akan di-*parsing* di dalam dokumen HTML dan akan ditampilkan kembali ke *web browser*. Kode PHP dapat juga disisipkan ke dalam kode HTML. PHP memiliki delapan tipe data. Empat nilai skalar (*single-value*) : *integer*, *floating-point*, *string*, dan *boolean*; dua nilai koleksi : *array* dan objek. Sisanya adalah tipe khusus : *resource* dan *NULL*. Dalam PHP, variabel didefinisikan dengan cara membubuhkan tanda \$ (*dollar*) di depannya. Dalam PHP, Operator dapat dikategorikan menjadi beberapa kelompok yaitu operator *assignment* (pengisian), aritmetika, *relational* dan logika [4].

1.9 Hypertext Markup Language (HTML)

HTML (*Hypertext Markup Language*) adalah pemrograman yang mengatur bagaimana kita menyajikan informasi di dunia internet dan bagaimana informasi itu membawa kita melompat dari satu tempat ke tempat lain.

Contoh HTML:

```
<html>
<head>
<title>Judul halaman</title>
</head>
<body>Hello World</body>
</html>
```

Sama seperti sebuah *software*, HTML memiliki beberapa versi. Versi terakhir yang diperkenalkan adalah HTML 5. HTML 5 memiliki kapabilitas baru dibandingkan versi sebelumnya, HTML 4. Contoh tag dalam HTML 5 <audio> dan <video> yang khusus berfungsi untuk memasang suara atau musik dan video ke dalam *website*. Walaupun belum semua *browser* mampu menginterpretasikan *tag-tag* dalam HTML 5 secara utuh, namun perlahan tapi pasti versi terbaru dari *Hypertext Markup Language* ini akan mendominasi desain *website* [7].

1.10 MySQL

MySQL adalah salah satu program yang dapat digunakan sebagai database, dan salah satu *software* untuk *database server* yang banyak digunakan. MySQL bersifat *Open Source* dan menggunakan SQL. MySQL bisa dijalankan di berbagai *platform* misalnya *Windows*, *Linux*, dan lain sebagainya.

MySQL memiliki beberapa kelebihan, antara lain:

1. MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah.
2. MySQL memiliki kecepatan yang bagus dalam menangani *query* sederhana.
3. MySQL memiliki operator dan fungsi secara penuh dan mendukung perintah *Select* dan *where* dalam perintah *query*.
4. MySQL memiliki keamanan yang bagus karena beberapa lapisan sekuritas seperti level *subnmask*, nama *host*, dan izin akses *user* dengan sistem perijinan yang mendetail serta sandi terenskripsi.
5. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*record*) lebih dari 50 juta dan 60 ribu

table serta kurang lebih 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

6. MySQL dapat melakukan koneksi dengan *client* menggunakan **protocol TCP/IP, Unix Soket (UNIX), atau Named PipesI (NT)**.
7. MySQL dapat mendeteksi pesan kesalahan pada *client* dengan menggunakan lebih dari 20 bahasa.
8. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti *Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga*, dan masih banyak lagi.
9. MySQL didistribusikan secara *open source*, di bawah lisensi GPL sehingga dapat digunakan secara gratis.

Koneksi ke *database* digunakan untuk mengakses data-data yang ada dalam *database* tersebut. Data tidak bisa diakses tanpa ada koneksi terlebih dahulu. Berikut ini *script* untuk koneksi ke *database*:
MySQL_Connect(nama_host, nama_user, password);

Keterangan:

- a. **Nama host** adalah lokasi tempat MySQL dipublikasikan, dalam latihan di buku ini nama *host* isi dengan localhost.
- b. **Nama user** yaitu nama *user* yang terdaftar dalam MySQL yang digunakan untuk mengakses data yang ada dalam MySQL, untuk menggunakan latihan di buku ini, nama *user* isi dengan **root**, yaitu *user* yang digunakan untuk membuka **phpMyAdmin**.
- c. **Password** adalah *password* yang digunakan untuk membuka *database* (phpMyAdmin), isi dengan *password* yang dibuat waktu instalasi AppServ.[5]

1.11 Cascading Style Sheet (CSS)

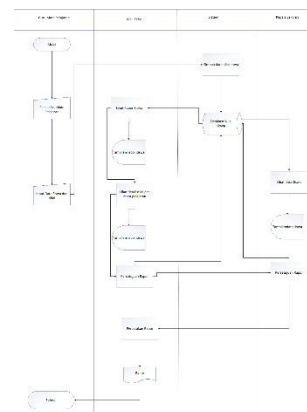
Salah satu bahasa pemrograman desain *web* (*style sheet language*) yang mengontrol format tampilan sebuah halaman *web* yang ditulis dengan menggunakan bahasa penanda (*markup language*). Biasanya CSS digunakan untuk mendesain halaman *HTML* dan *XHTML*, tetapi sekarang bahasa pemrograman CSS bisa diaplikasikan untuk segala dokumen *XML*, termasuk *SVG* dan *XVL*. CSS merupakan salah satu bahasa *style sheet style sheet website* yang sekarang sedang populer dipakai pada berbagai aplikasi *web*. CSS dibuat untuk membuat *style* pada *web* lebih bervariasi dan mudah digunakan [1].

1.12 Black Box Testing

Pengujian Black Box berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian black box memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Black box merupakan pengujian suatu program yang kemungkinan besar menangkap kesalahan dibandingkan dengan pengujian yang lainnya[16].

3. Analisis dan Perancangan

3.1 Sistem Pengolahan Nilai Yang diusulkan



Gambar 2

Keterangan :

- a. Guru mata pelajaran terlebih dahulu memilih data kelas dan semester yang akan diisi, kemudian guru mata pelajaran memasukkan nilai siswa kedalam aplikasi *raport online*.

- b. Setelah data nilai dimasukkan maka akan diproses oleh sistem dan akan disimpan pada *database*, untuk ditampilkan pada wali kelas masing – masing.
- c. Wali kelas akan melakukan cek nilai pada seluruh siswa, apakah nilai siswa sudah terpenuhi maka setelah nilai terpenuhi wali kelas akan menyetujui nilai-nilai.
- d. Kemudian kepala sekolah akan melihat nilai seluruh siswa dan menyetujui.
- e. Wali kelas akan mencetak *raport* siswa setelah disetujui oleh kepala sekolah.

3.2 Kebutuhan Perangkat Keras dan Perangkat Lunak

Berikut adalah kebutuhan perangkat keras yang menunjang pembuatan proyek akhir meliputi :

- a) *Processor* : Intel(R) Core(TM) i3-2350 CPU @2350 2.30 GHz
- b) *Memory* : 4 GB RAM
- c) *Hardisk* : 500 GB

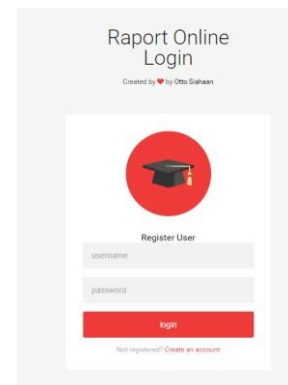
Berikut adalah kebutuhan perangkat keras yang menunjang pembuatan proyek akhir meliputi :

- a) *Sistem Operasi* : Windows 7 Home Premium
- b) *Sistem Manajemen Basis Data* : MySQL, XAMPP
- c) *Script Editor* : Sublime Text 3
- d) *Bahasa Pemrograman* : HTML, CSS, Javascript, PHP
- e) *Framework* : Codeigniter 3.1

4. IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Antarmuka Halaman Login

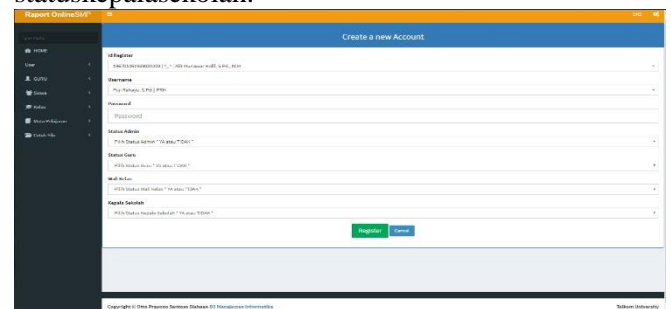
Halaman *login* merupakan halaman utama ketika sistem pertama kali dijalankan pada *web browser*. *Form login* dapat digunakan sekaligus oleh bagian kurikulum, guru, walikelas, dan kepala sekolah.



Gambar 3

4.2 Implementasi Antarmuka Halaman Admin Tambah User

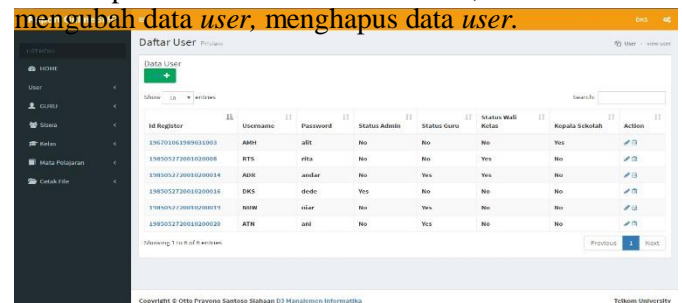
Halaman ini digunakan untuk menambah *user* bagi para guru agar bisa dapat melakukan *login*. Adapun *form* yang harus diisi ketika mendaftarkan *user* yang baru adalah idregister (sesuai dengan nip), username, password, statusadmin, statusguru, statuswali, statuskepalaSekolah.



Gambar 4

4.3 Implementasi Antarmuka Halaman Admin View User

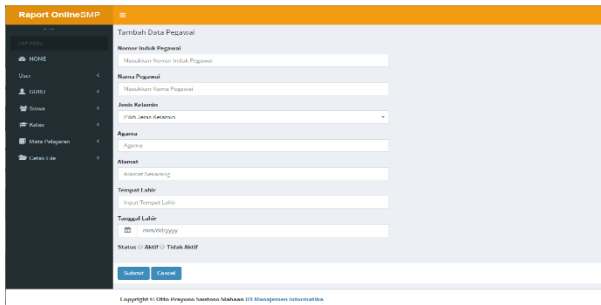
Halaman ini digunakan untuk menampilkan data *user* yang sudah terdaftar. Pada halaman ini terdapat fitur untuk menambah *user*, **mengubah data user, menghapus data user.**



Gambar 5

4.4 Implementasi Antarmuka Halaman Admin Tambah Guru

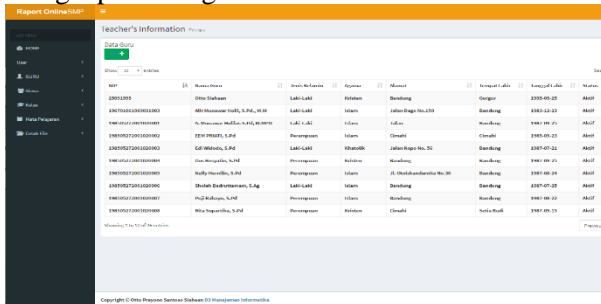
Halaman ini digunakan untuk menambah data guru. Data yang harus diisi meliputi nomor induk pegawai(nip), nama pegawai, jenis kelamin, agama, alamat, tanggal lahir, tempat lahir dan status.



Gambar 6

4.5 Implementasi Antarmuka Halaman Admin View Data Guru

Halaman ini digunakan untuk menampilkan data guru yang sudah ditambahkan. Pada halaman ini terdapat fitur untuk menambah data guru baru, mengubah data guru, menghapus data guru.



Gambar 7

4.6 Implementasi Antarmuka Halaman Cetak Data Guru

Pada halaman ini digunakan untuk mencetak data guru yang sudah ada. Hasilnya nanti akan disimpan dalam bentuk file pdf.

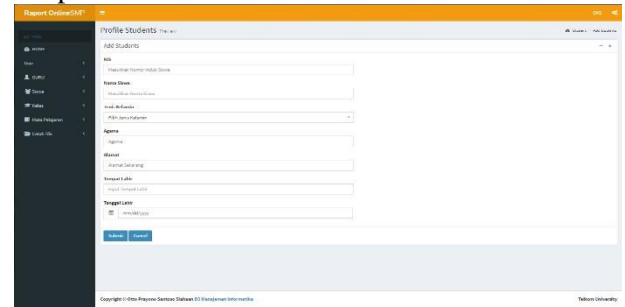


Gambar 8

4.7 Implementasi Antarmuka Halaman Tambah Siswa

Halaman ini digunakan untuk mengisi data siswa yang baru. Data siswa yang harus diisi meliputi nomor induk siswa(nis), nama siswa,

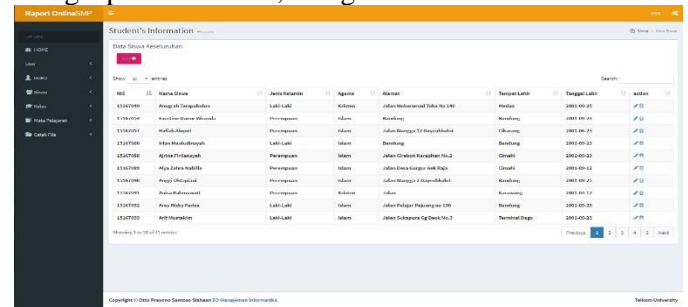
jenis kelamin, agama, alamat, tanggal lahir dan tempat lahir.



Gambar 9

4.8 Implementasi Antarmuka Halaman View Siswa

Halaman ini digunakan untuk menampilkan data siswa yang sudah ada. Pada halaman ini terdapat fitur untuk menambah data siswa, menghapus data siswa, mengubah data siswa.



Gambar 10

4.9 Implementasi Antarmuka Halaman Cetak Siswa

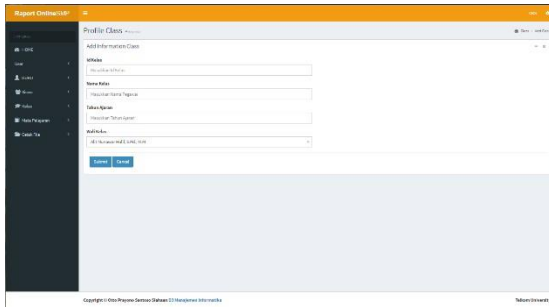
Halaman ini digunakan untuk melakukan cetak data siswa. Halaman melakukan cetak berdasarkan kelas siswa.



Gambar 11

4.10 Implementasi Antarmuka Halaman Tambah Data Kelas Halaman

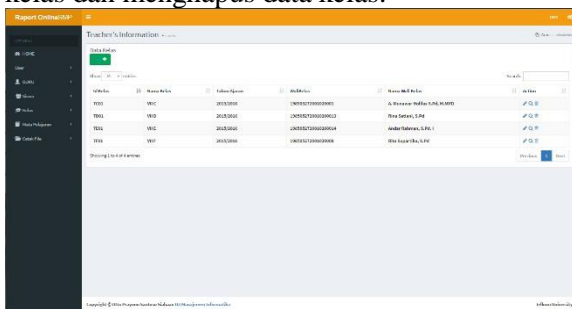
tambah kelas ini digunakan untuk menambah data kelas. Data yang harus diisi ketika melakukan tambah kelas adalah idkelas, nama kelas, tahun ajaran dan wali kelas.



Gambar 12

4.11 Implementasi Antarmuka Halaman View Kelas

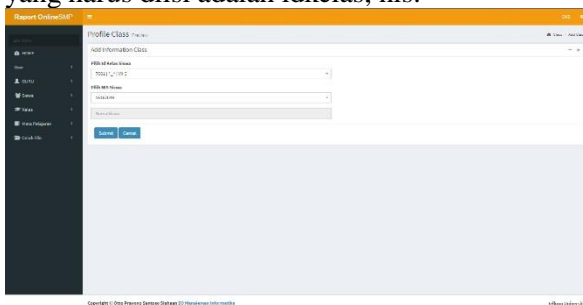
Halaman ini menampilkan data kelas yang sudah ada. Pada halaman ini terdapat fitur untuk menambah data kelas, mengubah data kelas dan menghapus data kelas.



Gambar 13

4.12 Implementasi Antarmuka Halaman Tambah Kelas Siswa

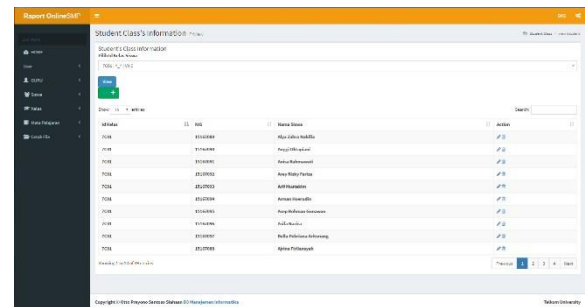
Halaman ini digunakan untuk menambah daftar siswa pada suatu kelas. Dimana data yang harus diisi adalah idkelas, nis.



Gambar 14

4.13 Implementasi Antarmuka Halaman View Kelas Siswa

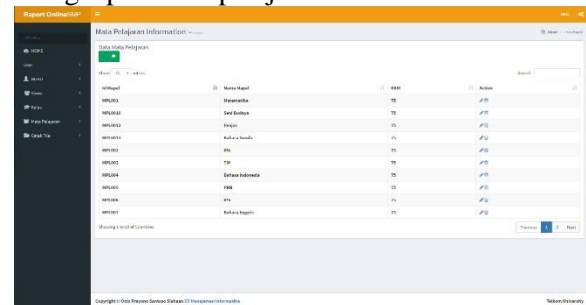
Halaman ini menampilkan data setiap kelas yang ada. Halaman ini memiliki fitur untuk menampilkan data siswa pada kelas tertentu dan dapat melakukan tambah siswa pada kelas tertentu.



Gambar 15

4.14 Implementasi Antarmuka Halaman Mata Pelajaran

Halaman ini menampilkan daftar mata pelajaran yang ada. Pada halaman ini terdapat fitur menambah mata pelajaran, mengubah mata pelajaran, dan menghapus mata pelajaran.



Gambar 16

4.15 Implementasi Antarmuka Tambah Mata Pelajaran

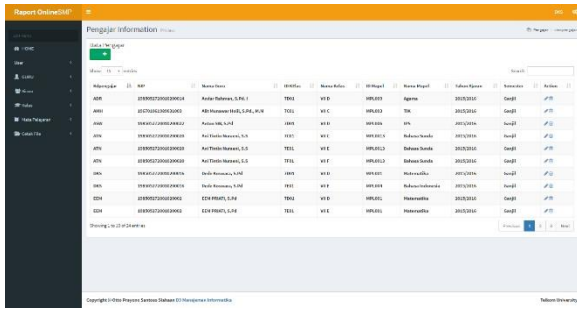
Halama ini digunakan untuk menambah data mata pelajaran dimana data yang harus diisi adalah idmapel, nama mapel dan kkm.



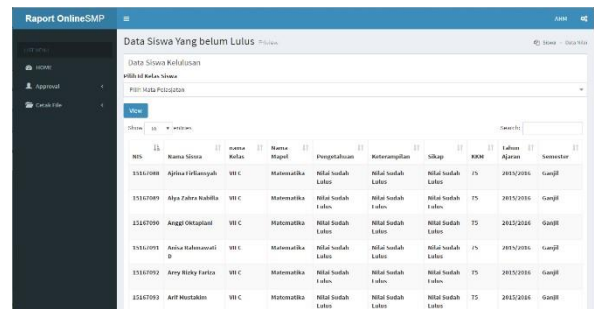
Gambar 17

4.16 Implementasi Antarmuka View Pengajar

Halaman ini menampilkan data pengajar pada setiap kelas yang ada. Pada halaman ini terdapat fitur untuk menambah pengajar, mengubah pengajar dan menghapus data pengajar.



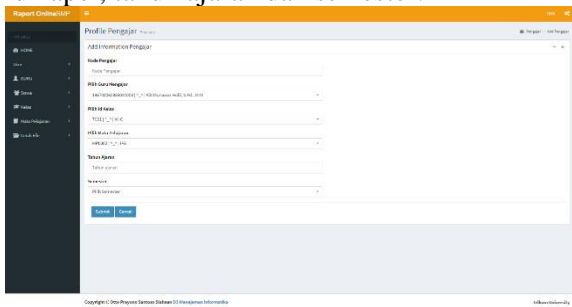
Gambar 18



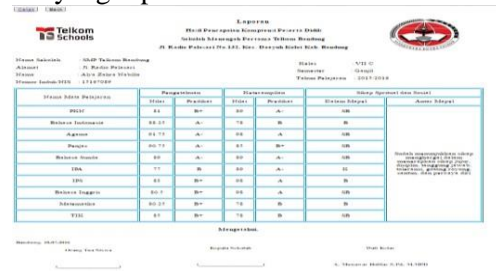
Gambar 21

4.17 Implementasi Antarmuka Tambah Pengajar
 Halaman ini digunakan untuk menambah data pengajar. Data yang harus dimasukkan adalah kode pengajar, nip guru pengajar, idkelas, idmapel, tahun ajaran dan semester.

4.20 Implementasi Antarmuka Halaman Wali Cetak Report
 Halaman ini digunakan oleh wali kelas untuk melakukan cetak raport siswa sesuai dengan nis yang dipilih.



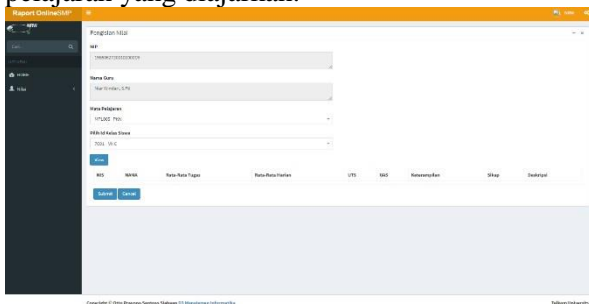
Gambar 19



Gambar 22

4.18 Implementasi Antarmuka Guru Tambah Nilai
 Halaman ini digunakan oleh guru untuk mengisi form nilai sesuai kelas dan mata pelajaran yang diajarkan.

4.21 Implementasi Antarmuka Wali Cetak Deskripsi
 Halaman ini digunakan oleh wali kelas untuk melakukan cetak deskripsi nilai siswa sesuai nis yang dipilih.



Gambar 20

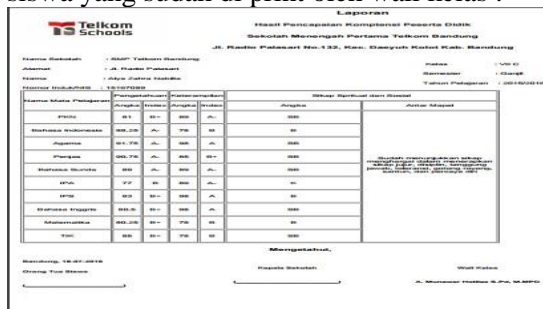


Gambar 23

4.19 Implementasi Antarmuka Wali View Kelulusan
 Halaman ini digunakan oleh wali kelas untuk melihat data nilai siswa yang belum lulus di setiap mata pelajaran.

4.22 Impelentasi Antarmuka Report Siswa
 Halaman ini digunakan oleh wali kelas untuk melakukan cetak raport siswa sesuai dengan nis yang dipilih.

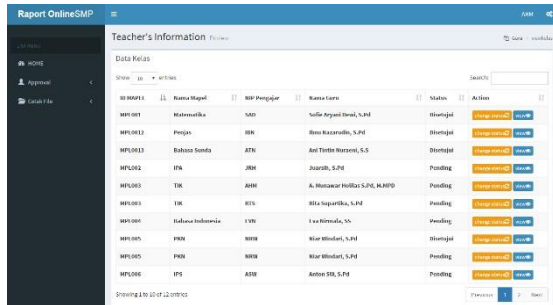
Halaman ini merupakan hasil cetak raport siswa yang sudah di print oleh wali kelas .



Gambar 24

4.23 Implementasi Antarmuka Approval Nilai

Halaman ini digunakan oleh wali kelas dan kepala sekolah dalam melakukan approval nilai sebelum dilakukan cetak nilai.



Gambar 25

5. Kesimpulan dan Saran

5.1 Kesimpulan

Setelah melakukan analisis, desain, pengembangan, implementasi, evaluasi dan pengujian terhadap proyek akhir ini maka dapat menyimpulkan:

1. Aplikasi ini dapat mengolah data nilai siswa dan menghitung nilai siswa secara tepat.

2. Aplikasi ini menyediakan pengisian raport secara *online* dan dapat melakukan cetak data.

5.2 Saran

Saran yang dapat dipertimbangkan untuk pengembangan Aplikasi Rapor Online ini di masa mendatang adalah :

1. Menambahkan fitur dalam pembagian kelas dan pembagian guru mata pelajaran setiap kelas.

2. Menambahkan fitur presensi siswa.

DAFTAR PUSTAKA

- [1] R. Sianipar, Pemrograman JavaScript Teori dan Implementasi, Bandung: Informatika, 2015.
- [2] B. Sidik, JavaScript, Bandung: Informatika, 2011.
- [3] C. W. Hermawan, ShortCourse: PHP Programming, Semarang: Andi, 2009.
- [4] B. Raharjo and I. Heryanto, Modul Pemrograman WEB (HTML, PHP, dan YSQL), Bandung: Modula, 2010.
- [5] K. Peranginangin, Aplikasi WEB dengan PHP dan MYSQL, Yogyakarta: CV. ANDI OFFSET (Penerbit Andi), 2006.
- [6] A. Yulianto, Ardhian and dkk, Praktikum Analisis dan Desain Sistem Informasi, Bandung: Politeknik Telkom, 2009.
- [7] J. Enterprise, Step by Step HTML 5, Jakarta: PT Elex Media, 2011.
- [8] J. Simarmata, Rekayasa Perangkat Lunak, Yogyakarta: CV. Andi Offset, 2010.