

**SISTEM INFORMASI PENGOLAHAN NILAI SISWA
PADA SEKOLAH MENENGAH**

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat mencapai gelar Ahli Madya Program
Diploma III Ilmu Komputer Fakultas Matematika dan Ilmu pengetahuan Alam
Universitas Sebelas Maret



Disusun oleh :

CANDRA TRI HARJANTO

NIM. M3108026

**PROGRAM DIPLOMA III ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SEBELAS MARET**

SURAKARTA
commit to user
2011

HALAMAN PERSETUJUAN

**SISTEM INFORMASI PENGOLAHAN NILAI SISWA
PADA SEKOLAH MENENGAH**

Disusun Oleh



CANDRA TRIHARJANTO

NIM M3108026

Proposal Tugas Akhir ini telah disetujui untuk disusun dan ditindak lanjuti
pada tanggal _____

Pembimbing

Ristu Saptono, S.Si.,M.T

NIP. 19790210 200212 1 001

commit to user

HALAMAN PENGESAHAN

**SISTEM INFORMASI PENGOLAHAN NILAI SISWA
PADA SEKOLAH MENENGAH**

Disusun oleh :

CANDRA TRI HARJANTO

NIM M3108026

Dibimbing oleh :
Pembimbing Utama

Ristu Saptono, S.Si.,M.T

NIP. 19790210 200212 1 001

Tugas Akhir ini telah diterima dan disahkan
Oleh dewan penguji Tugas Akhir
Program Diploma III Ilmu Komputer
pada hari _____ tanggal _____

Dewan Penguji

1. Penguji 1. Ristu Saptono, S.Si.,M.T ()
NIP. 19790210 200212 1 001
2. Penguji 2. Sakroni, S.Kom ()
NIP. 19700603 2001121001
3. Penguji 3. Fendi Aji Purnomo, S.Si ()
NIDN. 0626098402

Disahkan Oleh :

Dekan
Fakultas MIPA UNS

Ketua Program Studi
DIII Ilmu Komputer UNS

Ir. Ari Handono Ramelan, M.Sc (Hons), Ph.D

NIP. 19610223 198601 1 001

Drs. YS. Palgunadi, M.Sc

NIP. 19560407 198303 1 004

ABSTRACT

CANDRA TRI HARJANTO, 2011. GRADES PROCESSING INFORMATION SYSTEM AT HIGH SCHOOL. Diploma Degree Computer Science, Mathematics and Natural Sciences Faculty, Sebelas Maret University of Surakarta.

Grade processing information system was development to make proses submission and processing of grade become faster, precise, and easy in operation. The purpose of making this system is to develop the grade processing information system not only for junior high school but also for senior high school fastly, precise, and accurate and to support teaching and learning process very well.

The methods used in completing this final project were observation and interview methods of data collection techniques using primary and secondary. The program was developed using PHP and MySQL-based orientation with CodeIgniter.

This application is a web based so that it can be run using a browser facility and the local server. The additional facility is a useful for admins to manage users which give the level of the user. In addition, there are facilities to change the password on each user.

Key words: grade processing system, high school.

ABSTRAK

CANDRA TRI HARJANTO, 2011. SISTEM INFORMASI PENGOLAHAN NILAI SISWA PADA SEKOLAH MENENGAH. Diploma III Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sebelas Maret Surakarta.

Sistem Informasi Pengolahan Nilai ini dikembangkan untuk membuat proses penyerahan dan pengolahan nilai menjadi lebih cepat, tepat, dan mudah dalam pengoperasiannya. Tujuan dari pembuatan sistem ini adalah untuk mengembangkan Sistem Informasi Pengolahan Nilai untuk Sekolah Menengah Pertama maupun Atas secara cepat, tepat, dan akurat serta mendukung proses belajar mengajar dengan baik.

Metode yang digunakan dalam menyelesaikan tugas akhir ini adalah metode observasi dan wawancara dengan menggunakan teknik pengumpulan data primer dan sekunder. Program ini dikembangkan dengan menggunakan PHP berbasis Orientasi dan MySQL dengan Codeigniter.

Aplikasi ini berbasis web sehingga dapat dijalankan menggunakan fasilitas *browser* dan *local server*. Adapun fasilitas tambahan adalah fasilitas untuk admin yang berguna mengelola pengguna yang memberi level kepada pengguna. Selain itu, terdapat fasilitas untuk merubah sandi pada masing-masing pengguna.

Kata kunci : sistem pengolahan nilai, sekolah menengah.

MOTTO

"Kepercayaan adalah sumber dari kekuatan"

"Berusahalah diimbangi dengan berdoa"

"Tidak ada kata yang tidak mungkin didunia ini"

"Perjuangkan semua apa yang pantas kau perjuangkan"

"Last not but least"



commit to user

KATA PENGANTAR

Assalammu'alaikum Wr. Wb.

Alhamdulillahirobbilalamin, puji syukur kehadirat Allah SWT atas segala rahmat dan hidayahNya, sehingga penulis dapat menyelesaikan penulisan laporan Tugas Akhir ini dengan baik.

Tugas Akhir ini disusun sebagai salah satu syarat untuk memenuhi persyaratan guna memperoleh derajat Ahli Madya Ilmu Komputer Jurusan Manajemen Informatika Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sebelas Maret Surakarta.

Dalam pembuatan laporan ini, penulis mengucapkan terimakasih kepada semua pihak yang telah membantu dalam penyelesaian laporan Tugas Akhir ini, antara lain :

1. Drs. YS. Palgunadi, M.Sc. selaku ketua jurusan DIII Ilmu Komputer FMIPA UNS.
2. Ristu Saptono, S.Si.,M.T selaku dosen pembimbing dari jurusan DIII Ilmu Komputer yang telah memberikan pengarahan dan masukan dalam penyusunan laporan kerja magang mahasiswa ini.
3. Ayah, Bunda, dan kakak yang selalu memberi dukungan.
4. Teman-teman D3 Ilmu Komputer, yang pantang menyerah dan selalu meluangkan waktu untuk membantu menyelesaikan kesulitan yang dihadapi penulis.

Semoga amal dan kebaikan kita semua dibalas dengan nikmatNya yang tiada henti dan mudah-mudahan selalu diberikan ridhoNya.

Terima kasih

Wassalammu'alaikum Wr. Wb.

Surakarta,

2011

Penulis

commit to user

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERSETUJUAN.....	ii
HALAMAN PENGESAHAN.....	iii
ABSTRACT.....	iv
ABSTRAK.....	v
MOTTO.....	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
BAB I PENDAHULUAN.....	1
A. Latar Belakang Masalah.....	1
B. Perumusan Masalah.....	1
C. Batasan Masalah.....	1
D. Tujuan dan Manfaat.....	2
E. Metodologi Penelitian.....	3
F. Sistematika Penulisan.....	5
BAB II LANDASAN TEORI.....	7
A. <i>UML Diagram</i>	7
1. <i>Use Case Diagram</i>	7
2. <i>Class Diagram</i>	9
3. <i>Squence Diagram</i>	10
4. <i>Activity Diagram</i>	12
B. Web Programing.....	14
C. Desain Web.....	14
1. HTML.....	14
2. CSS.....	15
D. DBMS.....	16
BAB III ANALISIS KEBUTUHAN DAN PERANCANGAN.....	18
A. Deskripsi Kebutuhan.....	18

1. Kebutuhan Fungsional.....	18
2. Kebutuhan Non-Fungsional	19
B. UML <i>Diagram</i>	20
1. <i>Use Case Diagram</i>	20
2. <i>Diskripsi Use Case</i>	20
a. Hak dan Kewajiban Aktor	20
b. Use Case Glossary	21
3. <i>Use Case Naratif</i> atau <i>Use Case Skenario</i>	22
4. <i>Class Diagram</i> tahap <i>Analysis</i>	35
5. <i>Squence Diagram</i>	36
6. <i>Class Element Diagram</i>	42
C. Perancangan Database	45
D. Perancangan Antarmuka (<i>Interface</i>)	45
BAB IV IMPLEMENTASI DAN HASIL PENGUJIAN	52
A. Implementasi Antarmuka	52
B. Evaluasi Program	62
BAB V PENUTUP.....	66
A. Kesimpulan.....	66
B. Saran.....	66
DAFTAR PUSTAKA	55
LAMPIRAN.....	56

DAFTAR TABEL

	halaman
Tabel 2.1 : Simbol <i>Use Case Diagram</i>	8
Tabel 2.2 : Simbol <i>Class Diagram</i>	10
Tabel 2.3 : Simbol <i>Sequence Diagram</i>	11
Tabel 2.4 : Simbol <i>Activity Diagram</i>	13
Tabel 3.1 : Hak dan Kewajiban Aktor	20
Tabel 3.2 : <i>Use Case Glossary</i>	21
Tabel 3.3 : <i>Use Case Naratif Login</i>	22
Tabel 3.4 : <i>Use Case Naratif Mengelola User</i>	23
Tabel 3.5 : <i>Use Case Naratif Add Data Nilai Matapelajaran</i>	25
Tabel 3.6 : <i>Use Case Naratif View Data Nilai Akhir</i>	28
Tabel 3.7 : <i>Use Case Naratif Print Raport</i>	30
Tabel 3.8 : <i>Use Case Naratif Mengelola Profil</i>	31
Tabel 3.9 : <i>Use Case Naratif Search Nilai</i>	33
Tabel 4.1 : Evaluasi Program	62

DAFTAR GAMBAR

	halaman
Gambar 1.1 : Metode Penelitian.....	3
Gambar 3.1 : Diagram <i>Use Case</i>	20
Gambar 3.2 : <i>Class</i> Diagram Tahap Analisis.....	35
Gambar 3.3 : <i>Sequence</i> Login.....	36
Gambar 3.4 : <i>Sequence</i> View User.....	36
Gambar 3.5 : <i>Sequence</i> Tambah User.....	37
Gambar 3.6 : <i>Sequence</i> Update User.....	38
Gambar 3.7 : <i>Sequence</i> Mengelola Profil	39
Gambar 3.8 : <i>Sequence</i> Search Nilai.....	40
Gambar 3.9 : <i>Sequence</i> Update Nilai.....	41
Gambar 3.10 : <i>Sequence</i> Print Data Nilai Akhir.....	42
Gambar 3.11 : <i>Sequence</i> Print Raport.....	42
Gambar 3.12 : <i>Class Element</i> Login	43
Gambar 3.13 : <i>Class Element</i> User	43
Gambar 3.14 : <i>Class Element</i> Nilai Mata Pelajaran.....	44
Gambar 3.15 : <i>Class Element</i> Raport.....	44
Gambar 3.16 : Skema Diagram.....	45
Gambar 3.17 : Rancangan Form Login.....	45
Gambar 3.18 : Rancangan Form Guru Matapelajaran	46
Gambar 3.19 : Rancangan Form Wali Kelas	46
Gambar 3.20 : Rancangan Form Admin	47
Gambar 3.21 : Rancangan Form Profil <i>submit to user</i>	47

Gambar 3.22 : Rancangan Form Edit Profil.....	48
Gambar 3.23 : Rancangan Form Ganti <i>Password</i>	48
Gambar 3.24 : Rancangan Form <i>Select Input</i> Nilai	49
Gambar 2.25 : Rancangan Form <i>Input</i> Nilai	49
Gambar 2.26 : Rancangan Form <i>Select</i> Lihat Nilai	50
Gambar 2.27 : Rancangan Form Lihat Nilai.....	50
Gambar 2.28 : Rancangan Form Lihat Raport.....	51
Gambar 4.1 : Form Login	52
Gambar 4.2 : Form Menu User	53
Gambar 4.3 : Form Menu Wali Kelas.....	53
Gambar 4.4 : Form Menu Admin.....	54
Gambar 4.5 : Form Profil.....	55
Gambar 4.6 : Form Edit Profil	55
Gambar 4.7 : Form Ganti Profil	56
Gambar 4.8 : Form Selesct Input Nilai	56
Gambar 4.9 : Form Input Nilai.....	57
Gambar 4.10 : Form <i>Select</i> Lihat Nilai.....	58
Gambar 4.11 : Form Lihat Nilai.....	59
Gambar 4.12 : Form <i>Select Input</i> Raport.....	60
Gambar 4.13 : Form Lihat Raport.....	61

BAB I

PENDAHULUAN

A. LatarBelakang

Siswa di Sekolah Menengah dari kelas 1 sampai kelas 3 rata-rata berjumlah 960 siswa tiap semesternya. Banyak kendala bagi team evaluasi untuk mengelola nilai-nilai siswa yang jumlahnya 960 siswa ini, contoh pada salah satu Sekolah Menengah, Sekolah Menengah ini memiliki team evaluasi, yaitu nama untuk team pengelola nilai di sekolah menengah ini.

Salah satu kendala yang lain adalah proses pengumpulan nilai dari guru-guru mata pelajaran diserahkan kepada team evaluasi, maka pada saat itu guru-guru mata pelajaran memberikan daftar nilai siswa dalam bentuk tulisan tangan, setelah itu team evaluasi mengelola nilai-nilai siswa dari guru-guru mata pelajaran ini dengan menggunakan aplikasi *spread sheet*, sehingga team evaluasi harus menyalinnya lagi apalagi nilai-nilai yang diambil dari beberapa nilai yaitu : 4 kali nilai tugas, 4 kali nilai ulangan harian, ulangan mid semester dan ulangan semester, begitu banyak waktu dan tenaga yang terbuang untuk melakukan proses tersebut.

Sistem Informasi Pengolahan Nilai Siswa ini diharapkan menjadi solusi atas permasalahan pada Sekolah Menengah tersebut.

B. Perumusan Masalah

Berdasarkan latar belakang di atas dapat dibuat rumusan masalah bagaimana membangun Sistem Informasi Pengolahan Nilai pada Sekolah Menengah yang berbasis Kurikulum Tingkat Satuan Pendidikan (KTSP).

C. Batasan Masalah

Masalah dibatasi hanya pada pembuatan Sistem Pengolahan Nilai menggunakan *Framework Codeighter* yang akan melakukan proses Pengolahan Nilai dan Raport, sedangkan untuk masalah siswa, kelas dan nilai

commit to user

non akademis akan diatur pada sistem berbeda yaitu sistem BK, yang nantinya data siswa dan kelas akan diambil dari sistem BK.

D. Tujuan dan Manfaat

1. Tujuan

Membuat Sistem Informasi Pengolahan Nilai pada Sekolah Menengah untuk memudahkan proses pengolahan nilai.

2. Manfaat

Penyusunan tugas akhir ini diharapkan dapat memberikan manfaat sebagai berikut :

a. Bagi Akademik

- 1) Meningkatkan mutu pendidikan sehingga mahasiswa lebih siap dalam menghadapi persaingan kerja.
- 2) Menjadikan mahasiswa D3 Teknik Informatika Fakultas MIPA UNS berpengalaman dan profesional dalam bidangnya.
- 3) Terjalin kerjasama dan hubungan yang baik antara pihak D3 Teknik Informatika Fakultas MIPA UNS dengan perusahaan, instansi-instansi ataupun lembaga-lembaga.

b. Bagi Penulis

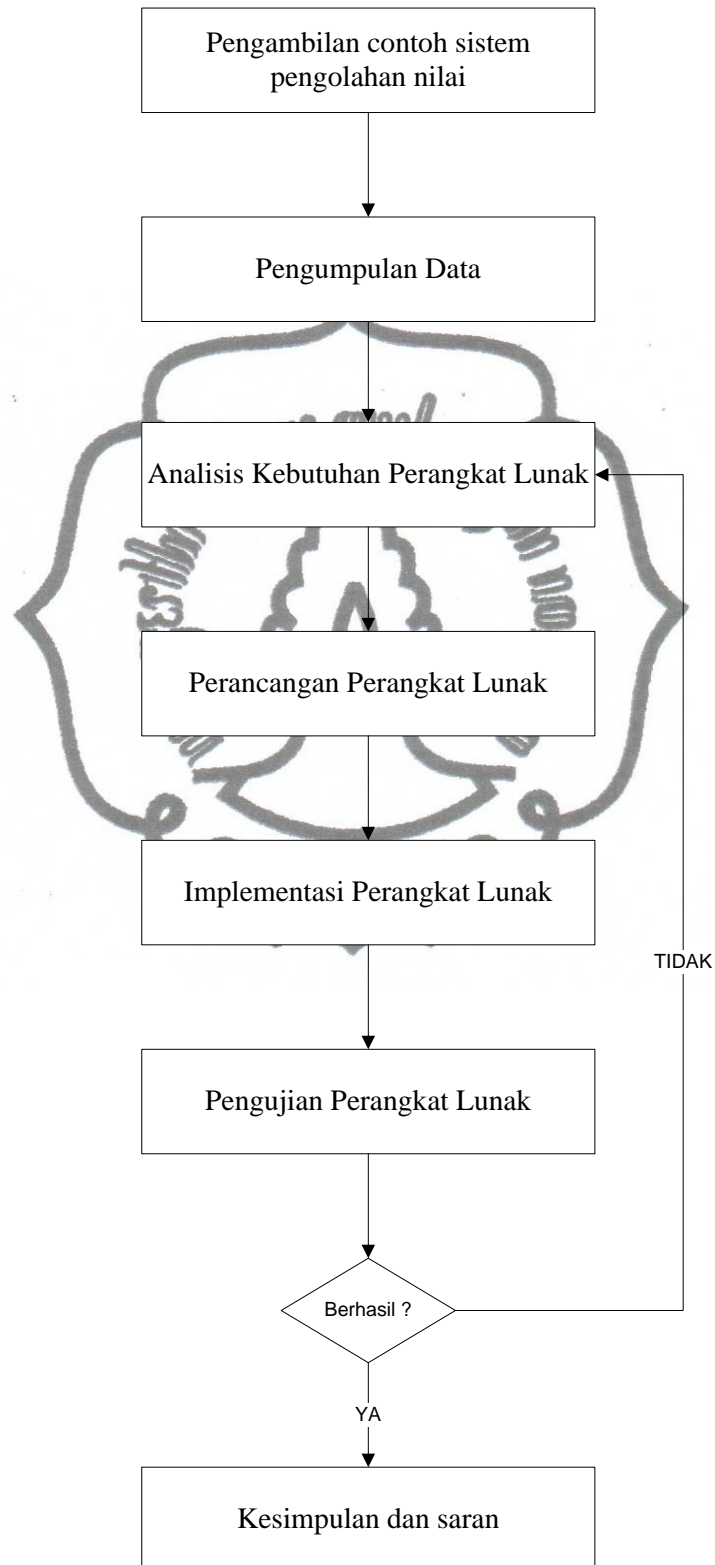
- 1) Mengembangkan dan menerapkan teori dan praktikum yang telah didapat selama mengikuti pendidikan di jurusan Manajemen Informatika D3 Teknik Informatika Fakultas MIPA UNS.
- 2) Meningkatkan pengetahuan dan pengalaman tentang teknik perancangan dan pembuatan sistem informasi.
- 3) Membantu dunia pendidikan dalam perkembangan teknologi informasi.

c. Bagi Pengguna(*User*)

Dapat membantu mempercepat proses pengolahan dan penyerahan nilai, sehingga dapat menghemat waktu yang terbuang dan dapat menyelesaikan pengelolaan tepat pada waktunya.

commit to user

E. Metode Penelitian



Gambar 1.1 Metode Penelitian

1. Pengambilan Contoh

Pengambilan contoh sistem pengolahan nilai ini merupakan tahap awal, dimana setiap sekolah menengah mempunyai alur sistem pengolahan yang berbeda-beda, meskipun sebagian besar sama, akan tetapi penulis berpandangan lebih baik mengambil contoh sistem pengolahan nilai pada salah satu sekolah menengah, supaya penulis mendapatkan sumber yang sudah jelas.

2. Pengumpulan Data

Dalam mengumpulkan data-data yang diperlukan terdapat dua metode yaitu :

1. Metode observasi adalah metode pengumpulan data dengan cara melakukan pengamatan dan pencatatan data-data serta berbagai hal yang akan dibutuhkan dalam proses penelitian.
2. Studi pustaka merupakan metode pencarian dan pengumpulan data dengan cara mencari referensi, literatur atau bahan-bahan teori yang diperlukan dari berbagai sumber wacana yang berkaitan dengan pembuatan aplikasi ini. Studi pustaka yang dilakukan oleh penulis yaitu dengan cara pengumpulan data melalui internet dan mencari buku-buku referensi yang berhubungan dengan tema dalam penyusunan tugas akhir.

3. Analisis Kebutuhan Perangkat Lunak

Pada tahapan ini penulis sudah mendapatkan data – data yang dibutuhkan, sehingga penulis sudah mulai membuat perancangan sistem yang baru, perancangan ini berisi meliputi : Deskripsi Kebutuhan, Diskripsi *Use Case*, *Diagram Use Case*, *Class Diagram* tahap analisis, *Squence Diagram*, dan *Class Element Diagram*.

4. Perancangan Perangkat Lunak

Pada tahap ini penulis akan mulai merancang perangkat lunak, langkah pertamanya dimulai dengan membuat perancangan database, dan perancangan antarmuka (*interface*).

commit to user

5. Implementasi Perangkat Lunak

Pada tahap ini penulis mulai mengerjakan pembuatan perangkat lunak dengan membuat *script program* dan desain untuk perangkat lunak, setelah merancanganya.

6. Pengujian Perangkat Lunak

Setelah aplikasi telah jadi, maka waktunya penulis menguji aplikasi, apakah sudah seperti yang diinginkan penulis.

F. Sistematika Penulisan

Untuk memberikan gambaran tentang pembahasan tugas akhir ini, maka secara garis besar pembahasan studi akhir ini adalah sebagai berikut:

BAB I : Pendahuluan meliputi latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan manfaat, metodologi penelitian, dan sistematika penulisan.

BAB II : Landasan teori, meliputi dasar-dasar teoritis yang menjadi landasan pemecahan masalah yang meliputi sistem, informasi, sistem informasi, basis data, pengertian-pengertian perancangan sistem, dan pengertian SQL.

BAB III : Analisis dan Perancangan, menguraikan tentang data yang diperlukan untuk membangun Sistem Informasi Pengolahan Nilai. Bab ini berisi deskripsi kebutuhan program, UML diagram yang terdiri dari *Diagram Use Case*, *Class Diagram* tahap *analisis*, *Sequence Diagram*, dan *Class Element Diagram*.

BAB IV : Implementasi dan Hasil Pengujian, menguraikan hasil akhir dari pembangunan Sistem Informasi Pengolahan Nilai berikut cara penggunaan Sistem Informasi Pengolahan Nilai agar pengguna lebih mudah memahami alur dari program yang digunakan.

BAB V : Kesimpulan dan Saran, memuat kesimpulan yang berisi tentang pernyataan singkat, tepat yang dijabarkan dari hasil penelitian dan pembahasan serta penutup memuat saran yang berisi tentang sumbang saran pemikiran yang didasarkan pada kesimpulan yang diperoleh untuk penyempurnaan dan pengembangan di masa mendatang.



BAB II

LANDASAN TEORI

A. UML Diagram

Unified Modelling Language (UML) adalah bahasa yang digunakan untuk menjelaskan dan memvisualisasikan artifak dari proses analisis dan disain berorientasi obyek. UML menyediakan standar pada notasi dan diagram yang bisa digunakan untuk memodelkan suatu sistem. UML dikembangkan oleh 3 pendekar ‘berorientasi obyek’, yaitu Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. UML menjadi bahasa yang bisa digunakan untuk berkomunikasi dalam perspektif obyek antara user dengan developer, antara developer dengan developer, antara developer analisis dengan developer disain, dan antara developer disain dengan developer pemrograman.

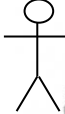
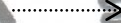

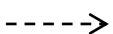



UML memungkinkan developer melakukan pemodelan secara visual, yaitu penekanan pada penggambaran, bukan didominasi oleh narasi. Pemodelan visual membantu untuk menangkap struktur dan kelakuan dari obyek, mempermudah penggambaran interaksi antara elemen dalam sistem dan mempertahankan konsistensi antara disain dan implementasi dalam pemrograman.




1. Use Case Diagram

Use Case Diagram (UCD) menjelaskan apa yang akan dilakukan oleh sistem yang akan dibangun dan siapa yang berinteraksi dengan sistem. UCD menjadi dokumen kesepakatan antara Customer, User dan Developer. User menggunakan dokumen UCD ini untuk memahami sistem dan mengevaluasi bahwa benar yang dilakukan sistem adalah untuk memecahkan masalah yang *user* ajukan atau sedang dihadapi. Developer menggunakan dokumen UCD ini sebagai rujukan yang benar dalam pengembangan sistem (Julius, H. 2004). *Use case diagram* pada umumnya tersusun dari elemen *actor*, *use case*, *dependency*, *generalization*, dan *commit to user*

association. UCD ini memberikan gambaran statis dari sistem yang sedang dibangun dan merupakan artifak dari proses analisis (Julius, H. 2004).

Tabel 2.1 Simbol *Use Case Diagram* (netindonesia.net)

No	Gambar	Nam	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<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>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi



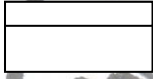
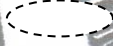

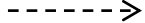
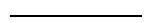
			yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

2. Class Diagram

Class diagram merupakan diagram yang selalu ada di pemodelan sistem berorientasi obyek. *Class diagram* menunjukkan hubungan antar class dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.

Class diagram umumnya tersusun dari elemen *Class*, *Interface*, *Dependency*, *Generalization* dan *Association*. Relasi *dependency* menunjukkan bagaimana ketergantungan terjadi antar *class* yang ada. Relasi *generalization* menunjukkan bagaimana suatu *class* menjadi *subclass* dari *class* tersebut. Relasi *association* menggambarkan navigasi antar *class*, berapa banyak obyek lain bisa berhubungan dengan satu obyek (*multiplicity* antar *class*), dan apakah suatu *class* menjadi bagian dari *class* lainnya (*aggregation*). *Class diagram* digunakan untuk menggambarkan disain statis dari sistem yang sedang dibangun (Julius, H. 2004).

Tabel 2.2 Simbol *Class Diagram* (netindonesia.net)

No	Gambar	Nama	Keterangan
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<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.
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

3. Sequence Diagram

Sequence diagram menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case*. Interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi. Pembuatan

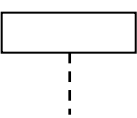
sequence diagram merupakan aktivitas yang paling kritikal dari proses disain karena artifak inilah yang menjadi pedoman dalam proses pemrograman nantinya dan berisi aliran kontrol dari program. Oleh karena itu berharga untuk meluangkan waktu lebih lama di pembuatan *sequence diagram* ini untuk menghasilkan *sequence diagram* yang terdisain dengan baik (Julius, H. 2004).


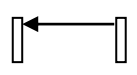
Sequence diagram biasanya tersusun dari elemen Obyek, *Interaction* dan *Message*. *Interaction* menghubungkan 3 obyek dengan pesannya. Diagram ini menjelaskan aspek dinamis dari sistem yang sedang dibangun.

Cara mudah yang bisa diikuti untuk memulai pembuatan *sequence diagram* adalah sebagai berikut :

- a. Gambarkan aktor dan kelas yang terlibat ke dalam *sequence diagram*.
- b. Urutkan sebagai berikut : *Actor* – obyek dari *boundary class* – obyek dari *control class* – obyek *entity class*.
- c. Ubah dari tipe *analysis class* menjadi *design class*.
- d. Ikuti urutan seperti dalam *use case spesification* dan mulai identifikasi operasi yang diperlukan untuk mengeksekusi suatu baris aktivitas dalam *use case spesification*. Operasi ini akan bergerak bertahap, dari *actor* ke *boundary class*, dari *boundary class* ke *control class*, dari *control class* ke satu atau beberapa *entity class*.
- e. Dari masing-masing operasi tersebut, identifikasi informasi apa saja yang perlu dipindahkan dari *actor* ke *bounday class* ke *control class* hingga ke *entity class* dan informasi apa yang harus dikembalikan dari *entity class* ke *boundary class* (Julius, H. 2004).

Tabel 2.3 Simbol *Sequence Diagram* (netindonesia.net)

No	Gambar	Nama	Keterangan
1		<i>LifeLine</i> <i>commit to user</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.

2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

4. Activity Diagram

Activity diagram menggambarkan berbagai aliran aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*) (Harmi, P. 2006).

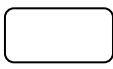




Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu (Pender, A Thomas. 2002).

Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu (Harmi, P. 2006).

Tabel 2.4 Simbol *Activity Diagram* (netindonesia.net)

No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan.
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

B. Web Programing

PHP adalah bahasa pemrograman yang memiliki sistem *interpreter* bukan sebagai *compiler*. Bahasa *interpreter* adalah bahasa pemrograman yang tidak memerlukan proses perubahan ke dalam bentuk *source code*, sehingga pada saat menjalankan program kode dasar akan segera langsung dijalankan (Nugroho, 2004).

Codeigniter adalah sebuah framework untuk web yang dibuat dalam format PHP, framework sendiri adalah sebuah susunan atau rangkaian kerja yang tetap dan dibuat sedemikian rupa yang kemudian dapat kembali dalam sebuah aktifitas kerja yang lain tetapi tetap dalam satu area kerja dengan rangkaian kerja yang sebelumnya (Komang Wiswakarma, 2010).

C. Desain Web

1. HTML

HTML (*Hypertext Markup Language*) merupakan salah satu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan di halaman web. Oleh karena itu, agar dapat membuat program aplikasi di atas halaman web, Anda terlebih dahulu harus mengenal dan menguasai HTML. Walaupun sekarang telah banyak paket aplikasi yang dapat digunakan untuk membuat halaman web secara WYSIWYG (*What you See Is What You Get*) seperti Frontpage, Dreamweaver, Netscape Composer, dan masih banyak lagi, namun Anda tetap harus menguasai *tag-tag* HTML karena mau tidak mau Anda akan bekerja dalam mode text editor bilamana hendak menyisipkan setiap script program dalam *script* HTML.

HTTP (*Hypertext Transfer Protocol*) merupakan protokol yang digunakan untuk mentransfer data antara *web server* ke *web browser*. Protokol ini mentransfer dokumen-dokumen web yang ditulis atau berformat HTML (*Hypertext Markup Language*).

Dikatakan *markup language* karena HTML berfungsi untuk memformat file dokumen teks biasa untuk bisa ditampilkan pada *web*

browser dengan bantuan tanda-tanda yang sudah ditentukan. Hal tersebut dapat dilakukan dengan menambahkan elemen atau yang sering disebut sebagai *tag* (Bima, S. 2001).

2. CSS

Kalau kita *browsing* situs-situs komersial kita lihat begitu cantik dan menariknya halaman demi halaman webnya. Bagaimana para pembuat situs tersebut merencangkannya? Tidak mungkin jika hanya menggunakan HTML saja. Mereka mengimprovisasi tampilan HTML itu dengan apa yang disebut *style sheet*, yang kemudian dikenal dengan CSS (*Cascading Style Sheet*) (Irwan, S. 2008).

Dengan CSS kita tidak hanya membuat *style* untuk teks, link, latar belakang. Lebih dari itu CSS dapat digunakan dalam membuat tampilan web yang cantik dan menarik dan memudahkan desainer web dalam mengganti-ganti tata letak dan nuansa (*theme*). Kelebihan lain menggunakan CSS, yaitu mengurangi penulisan berulang-ulang *style* yang sama di dalam halaman web.

Ada tiga cara menggunakan CSS dalam halaman web:

1. Dengan eksternal file yang di-*link*-an ke dalam halaman web. Nama file umumnya ditulis dengan ekstensi *.css*. Sintaks pemanggilan di halaman webnya adalah :

```
<link href="style.css" type="text/css" rel="stylesheet">
```

dimana *style.css* adalah nama file CSS.

2. Dengan menuliskan blok CSS di dalam halaman web itu sendiri. Penulisan ditempatkan di antara tag *<HEAD>* dan *</HEAD>*.

```
<head><style>
```

```
p { padding-bottom:12px; }
```

```
</style></head>
```

3. Dengan menuliskan kode CSS di sebelah kanan tag itu sendiri:

```
<p style="padding-bottom:12px;"> Your Text</p>
```

Kebanyakan para pengembang web menggunakan cara ke-1 dan cara ke-2 dalam menuliskan kode-kode CSS-nya karena lebih mudah

mengontrolnya dan menghindari penulisan *style* yang berulang-ulang (Irwan, S. 2008). Sintaks CSS dibangun oleh tiga bagian : *selector*, *property* dan *value*:

selector {property: value}

Selector elemen/tag HTML juga bisa didefinisikan. Setiap *property* adalah *attribute* yang bisa diganti dan setiap *attribute* mempunyai *value*. Antara *attribut* dan *value* dipisahkan oleh spasi.

body {color: black}

Jika *value* mengandung lebih dari satu kata penulisan diapit oleh tanda petik-ganda (“).

p {font-family: "sans serif"}

Jika ingin menuliskan beberapa properti, setiap properti dipisahkan dengan tanda titik-koma (;) (Irwan, S. 2008).

p {text-align:center;color:red}

D. DBMS

Kumpulan *file* yang saling berkaitan bersama dengan program untuk pengelolaannya disebut sebagai DBMS. Database adalah kumpulan datanya, sedangkan program pengelolaannya yang berdiri sendiri dalam satu paket program komersial untuk membaca data, mengisi data, menghapus data, melaporkan data dalam database (Kristanto, 2004).

DBMS ada 2 jenis, yaitu :

1. RDBMS (*Relational Database Management System*), adalah perangkat lunak yang didesain untuk mengatur/memanajemen sebuah basisdata sebagai sekumpulan data yang disimpan secara terstruktur, dan melakukan operasi-operasi atas data atas permintaan penggunaannya. Karakteristik dari RDBMS adalah :
 - 1) Menyediakan operator relasional untuk memanipulasi data dalam bentuk tabular.

commit to user

- 2) Menyajikan data pada pengguna dalam bentuk relasional (ditampilkan dalam bentuk tabular, sebagai koleksi dari tabel dimana setiap tabel berisi sekumpulan baris dan kolom).

Contoh dari RDBMS adalah MySQL, Microsoft Access, Microsoft SQL Server.

2. ORDBMS (*Object Relational Database Management System*), merupakan database sistem yang akan menggabungkan antara fitur pada model objek dan fitur yang ada pada model relasional, yaitu menerapkan konsep pada pemrograman berdasar objek sebagai perluasan dari model relasional, sehingga pada ORDBMS ini mempunyai *integrity constraint* yang tetap terjaga dan mendukung untuk data yang kompleks. Selain itu, penerapan model objek yang *reusable* dapat mempermudah pembuatan aplikasi basis data (Shalahuddin, 2005). Contoh dari ORDBMS adalah PostgreSQL, Oracle.

BAB III

ANALISIS KEBUTUHAN DAN PERANCANGAN

A. Deskripsi Kebutuhan

1. Kebutuhan Fungsional

a. Sistem bersifat

Web Hosting

b. Sistem dapat menampilkan seluruh informasi dari:

- 1) Guru Matapelajaran
Berkaitan dengan nilai awal matapelajaran
- 2) Wali Kelas
 - Berkaitan dengan nilai akhir matapelajaran
 - Raport siswa

Guru matapelajaran dan wali kelas masuk ke sistem melalui *login* sehingga memiliki session yang berbeda.

c. Sistem dapat menampilkan hasil:

- Nilai matapelajaran
- Raport siswa

d. Pengguna Sistem

Write : Administrasi, Guru matapelajaran, dan Wali kelas (*user*)

e. Setelah *user* login sesuai sessionnya, maka akan diberikan fungsi :

1) Admin

- Mengelola *user* : untuk mengelola *user* yang meliputi, *view user*, menambah *user* dan meng-*update user*
- *Search* Nilai : untuk mencari data nilai siswa sesuai dengan kriteria yang dipilih.

2) *User*

- *Add* data nilai matapelajaran : untuk memasukan data nilai matapelajaran. *commit to user*

- *Update* data nilai matapelajaran : untuk memperbaharui data nilai matapelajaran yang telah dimasukan.
- *View* data nilai akhir : untuk melihat data nilai akhir matapelajaran
- *Print* data nilai akhir : untuk mencetak data nilai akhir matapelajaran

3) *Manage* Raport

- *View* Raport : menampilkan data nilai akhir yang sudah di tulis pada format raport.
- *Print* Raport : mencetak raport

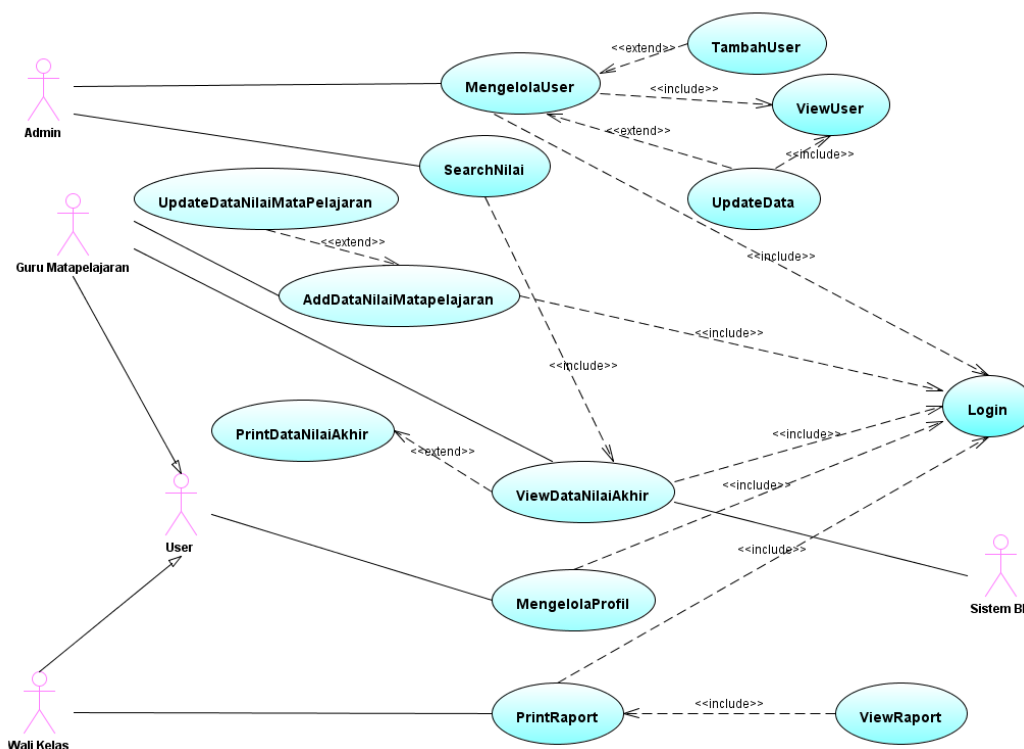
f. Setiap *user* diberikan fungsi untuk mengganti *password*.

2. Kebutuhan Non Fungsional

- Sistem mudah dipelajari dan digunakan
- Mengurangi sistem manual
- Dapat dibuka menggunakan *browser* standar
- Dapat dijalankan dengan sistem operasi windows maupun linux dan lainnya

B. UML Diagram

1. Use case Diagram



Gambar 3.1 Diagram Use case

2. Deskripsi Use case

a. Hak dan Kewajiban Aktor

Tabel 3.1 Hak dan Kewajiban Aktor

Nama	Hak dan Kewajiban
Admin	<ul style="list-style-type: none"> - Mengelola user yang meliputi : view user, tambah user, dan meng-update user - Mencari nilai dari keseluruhan siswa
Guru Matapelajaran	<ul style="list-style-type: none"> - Memasukan data nilai Matapelajaran dan meng-update nilai Matapelajaran - Melihat data nilai Matapelajaran - Mencetak data nilai Matapelajaran - Mengelola akun profilnya
Wali Kelas	<ul style="list-style-type: none"> - Melihat raport - Mencetak raport - Mengelola akun profilnya
Sistem BK	<ul style="list-style-type: none"> - Dapat melihat data nilai Matapelajaran tetapi aktor ini merupakan sistem sendiri.

b. Use case Glossary

Tabel 3.2 *Use case* Glossary

ID	Use case	Deskripsi
1	Login	Fungsi yang digunakan agar <i>user</i> dapat mengakses sistem sesuai dengan hak akses yang dimiliki pada masing-masing <i>user</i> .
2	Mengelola <i>User</i>	Fungsi untuk melakukan pengolahan terhadap <i>user</i> , yang meliputi <i>view user</i> , menambah <i>user</i> , meng- <i>update user</i> , dan memberi hak kepada <i>user</i> .
3	Add Data Nilai Matapelajaran	<i>Use case</i> yang hanya dapat diakses oleh guru matapelajaran untuk melakukan pengolahan data nilai matapelajaran, meliputi <i>add</i> dan <i>update</i> .
4	View Data Nilai Akhir	<i>Use case</i> yang hanya dapat diakses oleh guru matapelajaran dan sistem BK untuk melakukan pengolahan data nilai matapelajaran, meliputi <i>view</i> data nilai akhir dan <i>print</i> data nilai akhir.
5	Print Raport	<i>Use case</i> yang hanya dapat diakses oleh wali kelas untuk melakukan pengolahan raport, meliputi <i>view</i> dan <i>print</i> raport.
6	Mengelola Profil	<i>Use case</i> yang digunakan oleh <i>user</i> untuk mengelola profil mereka, meliputi mengganti <i>password</i> dan data diri.
7	Nilai	<i>Use case</i> yang digunakan oleh Admin untuk mencari dan melihat nilai dari semua siswa yang datanya

	berada di <i>database</i> .
--	-----------------------------

3. Use case Naratif atau Use case Skenario

Tabel 3.3 Use case Naratif Login

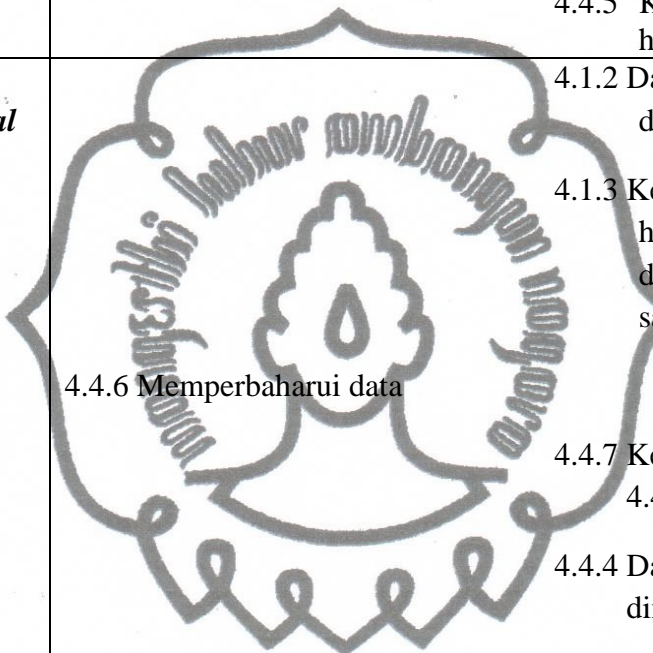
Use case Name	Login
ID	1
Impotance Level	High
Primary Actor	Admin, Guru Matapelajaran, Wali Kelas
Stakeholders and interest	Admin, Guru Matapelajaran, Wali Kelas : melakukan login untuk melakukan akses terhadap sistem
Brief Description	<i>Use case</i> ini berfungsi untuk login agar <i>user</i> dapat dikenali oleh sistem sehingga dapat mengakses sestem sesuai dengan hak yang diberikan kepadanya
Trigger	Admin, Guru Matapelajaran, Wali Kelas ingin menggunakan sistem lebih lanjut
Type	Internal
Relationship	Assosiation : Include : Extend : Generalitation :
Normal flow event	<p>Aksi Admin, Guru Matapelajaran, Wali Kelas</p> <p>1. Memasukan <i>Username</i> dan <i>Password</i></p> <p><i>commit to user</i></p> <p>Reaksi</p> <p>2. Mencocokkan <i>username</i> dan <i>password</i> yang dimasukkan dengan</p>

	database 3. Jika data ada, login sukses 4. Mengambil level <i>user</i> yang login dari database 5. Masuk ke sistem sesuai level <i>user</i>
Sub flow	
Alternate/ Exceptional flow	4.Data tidak ada, login gagal 5.Kembali ke langkah 1

Tabel 3.4 Use case Naratif Mengelola User

Use case Name	Mengelola User
ID	2
Impotance Level	High
Primary Actor	Admin
Stakeholders and interest	Admin : <i>view user</i> , menambah <i>user</i> , meng- <i>update user</i> , dan memberi hak kepada <i>user</i> , <i>user</i> meliputi guru matapelajaran, dan wali kelas dimana data yang dimiliki terdaftar didalam system.
Brief Description	<i>Use case</i> ini berfungsi untuk mengelola <i>user</i> seperti <i>view user</i> , menambah <i>user</i> , meng- <i>update user</i> , dan memberi hak kepada <i>user</i> .
Trigger	Admin memiliki data <i>user</i> baru atau data baru tentang <i>user</i>
Type	Internal
Relationship	Assosiation : admin

	<p>Include : <i>view user</i>, login</p> <p>Extend : <i>update data</i>, tambah <i>user</i></p> <p>Generalitation :</p>														
Normal flow event	<table border="0" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Aksi Admin</th> <th style="text-align: center;">Reaksi</th> </tr> </thead> <tbody> <tr> <td>1. Admin memilih menu tambah <i>user</i></td> <td></td> </tr> <tr> <td>2. Admin melakukan manipulasi data</td> <td>2. Sistem menampilkan data <i>user</i> yang harus diisi</td> </tr> <tr> <td>3. Admin melakukan manipulasi data</td> <td>4. Sistem menyimpan data</td> </tr> </tbody> </table>	Aksi Admin	Reaksi	1. Admin memilih menu tambah <i>user</i>		2. Admin melakukan manipulasi data	2. Sistem menampilkan data <i>user</i> yang harus diisi	3. Admin melakukan manipulasi data	4. Sistem menyimpan data						
Aksi Admin	Reaksi														
1. Admin memilih menu tambah <i>user</i>															
2. Admin melakukan manipulasi data	2. Sistem menampilkan data <i>user</i> yang harus diisi														
3. Admin melakukan manipulasi data	4. Sistem menyimpan data														
Sub flow	<table border="0" style="width: 100%;"> <tbody> <tr> <td>4.1 Admin menambah <i>user</i> baru</td> <td>4.1.1 Sistem melakukan pengecekan terhadap data yang dimasukan</td> </tr> <tr> <td></td> <td>4.1.2 Jika data benar system akan menyimpan data di database</td> </tr> <tr> <td></td> <td>4.1.3 Kembali ke halaman menu</td> </tr> <tr> <td>4.2 Melihat <i>user</i></td> <td>4.2.1 Sistem akan menampilkan nama <i>user</i></td> </tr> <tr> <td>4.3 Memilih <i>user</i></td> <td>4.3.1 Sistem akan menampilkan detail <i>user</i></td> </tr> <tr> <td>4.4 <i>Update user</i></td> <td>4.4.1 Sistem akan menampilkan data <i>user</i> yang akan diedit</td> </tr> <tr> <td>4.4.2 Admin memasukan data baru <i>user</i> <i>commit to user</i></td> <td></td> </tr> </tbody> </table>	4.1 Admin menambah <i>user</i> baru	4.1.1 Sistem melakukan pengecekan terhadap data yang dimasukan		4.1.2 Jika data benar system akan menyimpan data di database		4.1.3 Kembali ke halaman menu	4.2 Melihat <i>user</i>	4.2.1 Sistem akan menampilkan nama <i>user</i>	4.3 Memilih <i>user</i>	4.3.1 Sistem akan menampilkan detail <i>user</i>	4.4 <i>Update user</i>	4.4.1 Sistem akan menampilkan data <i>user</i> yang akan diedit	4.4.2 Admin memasukan data baru <i>user</i> <i>commit to user</i>	
4.1 Admin menambah <i>user</i> baru	4.1.1 Sistem melakukan pengecekan terhadap data yang dimasukan														
	4.1.2 Jika data benar system akan menyimpan data di database														
	4.1.3 Kembali ke halaman menu														
4.2 Melihat <i>user</i>	4.2.1 Sistem akan menampilkan nama <i>user</i>														
4.3 Memilih <i>user</i>	4.3.1 Sistem akan menampilkan detail <i>user</i>														
4.4 <i>Update user</i>	4.4.1 Sistem akan menampilkan data <i>user</i> yang akan diedit														
4.4.2 Admin memasukan data baru <i>user</i> <i>commit to user</i>															

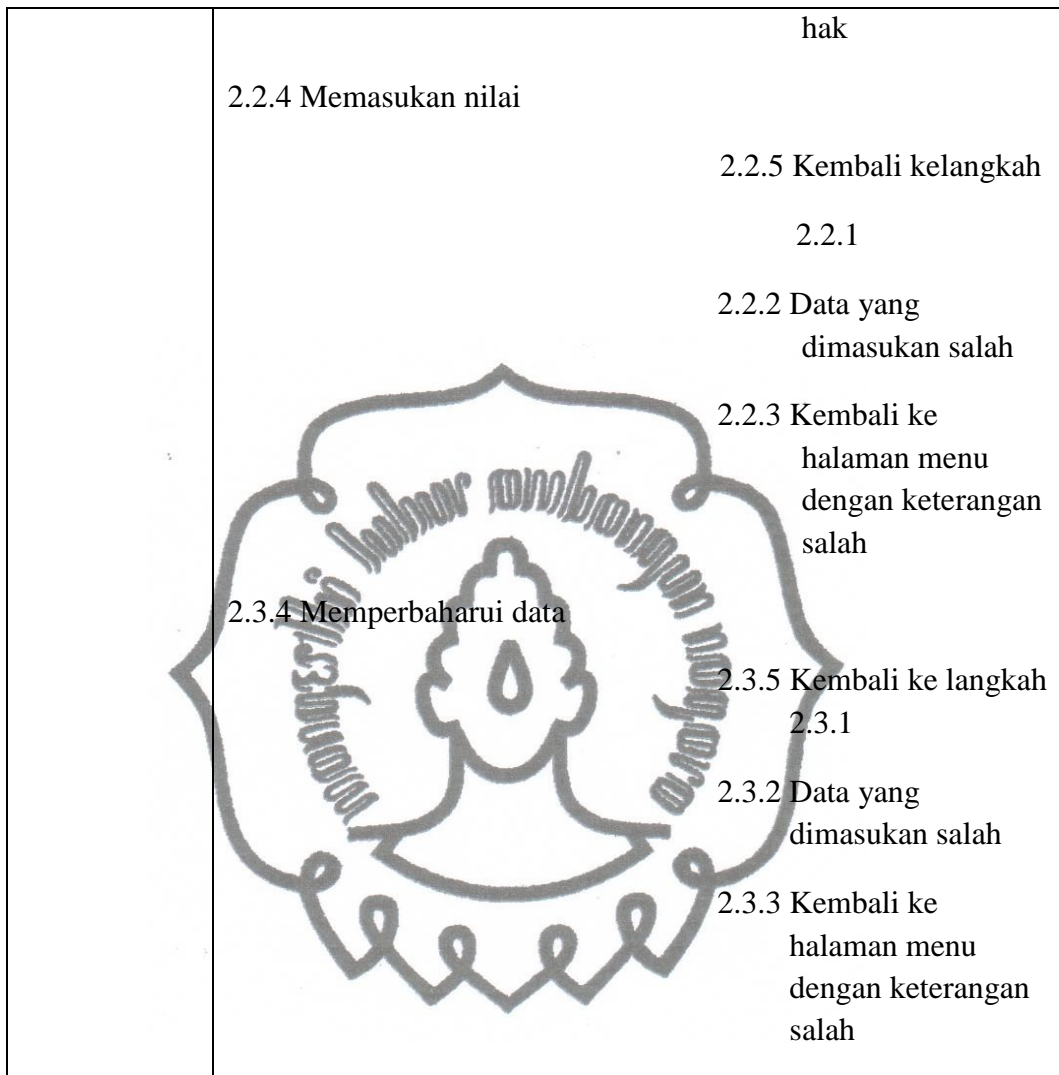
		<p>4.4.3 Sistem melakukan pengecekan terhadap data yang dimasukkan</p> <p>4.4.4 Jika data benar, system akan melakukan <i>update</i> data <i>user</i> di database</p> <p>4.4.5 Kembali ke halaman menu</p>
<i>Alternate/ Exceptional flow</i>	 <p>4.4.6 Memperbaharui data</p>	<p>4.1.2 Data yang dimasukan salah</p> <p>4.1.3 Kembali ke halaman menu dengan peringatan salah</p> <p>4.4.7 Kembali ke langkah 4.4.3</p> <p>4.4.4 Data yang dimasukan salah</p> <p>4.4.5 Kembali ke halaman menu dengan peringatan salah</p>

Tabel 3.5 *Use case Naratif Add Data Nilai Matapelajaran*

<i>Use case Name</i>	Add Data Nilai Matapelajaran
<i>ID</i>	3
<i>Impotance Level</i>	High

Primary Actor	Guru Matapelajaran				
Stakeholders and interest	Guru Matapelajaran : memasukan data nilai matapelajaran dan meng- <i>update</i> data nilai matapelajaran.				
Brief Description	<i>Use case</i> yang dapat diakses oleh guru matapelajaran untuk melakukan pengolahan data nilai matapelajaran, meliputi add, dan <i>update</i> data nilai matapelajaran.				
Trigger	Guru matapelajaran memasukan dan meng- <i>update</i> informasi data nilai matapelajaran yang akan diolah oleh sistem.				
Type	<i>Internal</i>				
Relationship	Assosiation : Guru Matapelajaran Include : login Extend : add data nilai matapelajaran Generalitaton :				
Normal flow event	<table border="0" style="width: 100%;"> <tr> <td style="width: 50%;">Aksi Guru Matapelajaran</td> <td style="width: 50%;">Reaksi</td> </tr> <tr> <td>1. Memilih menu add data nilai mata pelajaran.</td> <td>2. Menampilkan menu.</td> </tr> </table>	Aksi Guru Matapelajaran	Reaksi	1. Memilih menu add data nilai mata pelajaran.	2. Menampilkan menu.
Aksi Guru Matapelajaran	Reaksi				
1. Memilih menu add data nilai mata pelajaran.	2. Menampilkan menu.				
Sub flow	<p>2.1 Memilih menu untuk masukan nilai matapelajarannya.</p> <p>2.1.1 Sistem melakukan pengecekan terhadap hak Guru Matapelajaran.</p> <p>2.1.2 Jika benar Guru Matapelajaran mempunyai hak mengisi nilai, maka akan ditampilkan form nilai yang akan</p> <p style="text-align: center;"><i>commit to user</i></p>				

	<p>diisi.</p> <p>2.1.3 Kembali ke menu awal</p> <p>2.2 Memasukan data nilai mata pelajaran</p> <p>2.2.1 Sistem melakukan pengecekan terhadap data yang dimasukan</p> <p>2.2.2 Jika data benar sistem akan menyimpan data di database</p> <p>2.2.3 Kembali ke halaman memasukan data nilai matapelajaran</p> <p>2.3 Memasukan data baru nilai matapelajaran</p> <p>2.3.1 Sistem melakukan pengecekan terhadap data yang dimasukkan</p> <p>2.3.2 Jika data benar, sistem akan melakukan <i>update</i> data nilai matapelajaran di database</p> <p>2.3.3 Kembali ke halaman meng-<i>update</i> nilai matapelajaran.</p>
<p><i>Alternate/ Exceptional flow</i></p>	<p>2.1.2 Guru matapelajaran tidak mempunyai hak mengisi nilai</p> <p>2.1.3 Kembali ke menu awal, dengan keterangan bahwa tidak mempunyai</p> <p style="text-align: center;"><i>commit to user</i></p>



Tabel 3.6 Use case Naratif View Data Nilai Akhir

Use case Name	View Data Nilai Akhir
ID	4
Impotance Level	High
Primary Actor	Guru Matapelajaran, dan Sistem BK
Stakeholders	Guru Matapelajaran dan wali kelas : melakukan view dan print

and interest	data nilai matapelajaran. Sistem BK : melakukan <i>view</i> dan <i>print</i> data nilai matapelajaran, dimana sistem ini merupakan sistem lain yang mengakses database kusus data nilai akhir matapelajaran dan tidak akan dijelaskan fungsionalnya disini.
Brief Description	<i>Use case</i> yang dapat diakses oleh guru matapelajaran, wali kelas dan Sistem BK untuk melakukan pengolahan data nilai matapelajaran, meliputi <i>view</i> dan <i>print</i> data nilai matapelajaran.
Trigger	Guru Matapelajaran memiliki informasi nilai yang akan diolah, setelah itu hasilnya akan disampaikan pada wali kelas dan Sistem BK juga dapat melihat nilai matapelajaran beserta nilai akhirnya.
Type	<i>Internal</i>
Relationship	Assosiation : Guru Matapelajaran, Sistem BK Include : login, <i>print</i> data nilai akhir Extend : Generalitation :
Normal flow event	Aksi Guru Matapelajaran, Sistem BK Reaksi 1. Memilih kelas dan jurusan yang ingin di <i>view</i> data nilai akhir matapelajarannya. 2. Menampilkan data hasil nilai serta nilai akhir matapelajaran dari kelas dan jurusan yang telah dipilih.
Sub flow	2.1.1 Memilih menu <i>print</i> 2.1.2 Sistem akan mencetak nilai akhir matapelajaran dari <i>commit to user</i>

	kelas dan jurusan yang telah dipilih.
Alternate/ Exceptional flow	2.1.2 Data yang di <i>print</i> tidak ditemukan 2.1.3 Kembali ke proses 1

Tabel 3.7 Use case *Naratif Print Raport*

Use case Name	<i>Print Raport</i>						
ID	5						
Impotance Level	Medium						
Primary Actor	Wali Kelas						
Stakeholders and interest	wali kelas: melakukan pengolahan raport, meliputi <i>view</i> dan <i>print</i> raport						
Brief Description	<i>Use case</i> yang hanya dapat diakses oleh wali kelas untuk melakukan pengolahan raport, meliputi <i>view</i> dan <i>print</i> raport.						
Trigger	Wali kelas mendapatkan hasil raport dari siswa yang diampunya						
Type	<i>Internal</i>						
Relationship	Assosiation : Wali Kelas Include : login, <i>view</i> raport Extend : Generalititation :						
Normal flow event	<table border="0" style="width: 100%;"> <tr> <td style="text-align: center;">Aksi Wali Kelas</td> <td style="text-align: right;">Reaksi</td> </tr> <tr> <td style="text-align: center;"><i>commit to user</i></td> <td></td> </tr> <tr> <td>1. Memilih <i>view</i> raport pada siswa</td> <td></td> </tr> </table>	Aksi Wali Kelas	Reaksi	<i>commit to user</i>		1. Memilih <i>view</i> raport pada siswa	
Aksi Wali Kelas	Reaksi						
<i>commit to user</i>							
1. Memilih <i>view</i> raport pada siswa							

	<p>yang ingin dilihat raportnya.</p> <p>2. Menampilkan data nilai akhir matapelajaran yang telah disusun menjadi raport yang sesuai dengan siswa yang dipilih.</p>
Sub flow	<p>2.1 Memilih menu <i>print</i>.</p> <p>2.1.1 Sistem akan mengecek hak dari wali kelas, jika benar wali mengampu sesuai kelas yang dipilihnya maka sistem akan mencetak raport sesuai dengan siswa yang dipilih.</p>
Alternate/ Exceptional flow	<p>2.1.1 Wali kelas tidak mempunyai hak melihat raport</p> <p>2.1.2 Kembali ke proses 1</p>

Tabel 3.8 Use case Naratif Mengelola Profil

Use case Name	Mengelola Profil
ID	5
Impotance Level	Medium
Primary Actor	Guru Matapelajaran, Wali Kelas
Stakeholders	Guru Matapelajaran dan Wali Kelas : melakukan pengolahan

and interest	profil dirinya sendiri, meliputi		
Brief Description	<i>Use case</i> yang digunakan oleh <i>user</i> untuk mengelola profil mereka, meliputi mengganti <i>password</i> dan data diri.		
Trigger	Guru Matapelajaran, Wali Kelas mengelola profil mereka masing-masing		
Type	<i>Internal</i>		
Relationship	Assosiation : Guru Matapelajaran, Wali Kelas Include : login Extend : Generalitation : Mengelola profil		
Normal flow event	<table style="width: 100%; border: none;"> <tr> <td style="width: 50%; vertical-align: top;"> Aksi Guru Matapelajaran, Wali Kelas 1. Memilih profil 3. Melakukan pengolahan profil </td> <td style="width: 50%; vertical-align: top;"> Reaksi 2. Sistem menampilkan data <i>user</i> 4. Sistem menampilkan perubahan data </td> </tr> </table>	Aksi Guru Matapelajaran, Wali Kelas 1. Memilih profil 3. Melakukan pengolahan profil	Reaksi 2. Sistem menampilkan data <i>user</i> 4. Sistem menampilkan perubahan data
Aksi Guru Matapelajaran, Wali Kelas 1. Memilih profil 3. Melakukan pengolahan profil	Reaksi 2. Sistem menampilkan data <i>user</i> 4. Sistem menampilkan perubahan data		
Sub flow	<table style="width: 100%; border: none;"> <tr> <td style="width: 50%; vertical-align: top;"> 4.1 Mengelola Profil 4.1.2 Memasukan data baru untuk profil </td> <td style="width: 50%; vertical-align: top;"> 4.1.1 Sistem akan menampilkan data <i>user</i> lama yang akan diedit 4.1.3 Sistem melakukan pengecekan terhadap data yang dimasukkan 4.1.4 Jika data benar, 32system akan melakukan <i>update</i> data di database <i>commit to user</i> 4.1.5 Kembali ke </td> </tr> </table>	4.1 Mengelola Profil 4.1.2 Memasukan data baru untuk profil	4.1.1 Sistem akan menampilkan data <i>user</i> lama yang akan diedit 4.1.3 Sistem melakukan pengecekan terhadap data yang dimasukkan 4.1.4 Jika data benar, 32system akan melakukan <i>update</i> data di database <i>commit to user</i> 4.1.5 Kembali ke
4.1 Mengelola Profil 4.1.2 Memasukan data baru untuk profil	4.1.1 Sistem akan menampilkan data <i>user</i> lama yang akan diedit 4.1.3 Sistem melakukan pengecekan terhadap data yang dimasukkan 4.1.4 Jika data benar, 32system akan melakukan <i>update</i> data di database <i>commit to user</i> 4.1.5 Kembali ke		

	halaman mengelola profil
Alternate/ Exceptional flow	<p>4.1.6 Memperbaharui data</p> <p>4.1.7 Kembali ke langkah 4.1.3</p> <p>4.1.4 Data yang dimasukkan salah</p> <p>4.1.5 Kembali ke halaman mengelola profil dengan penandaan di data yang salah</p>

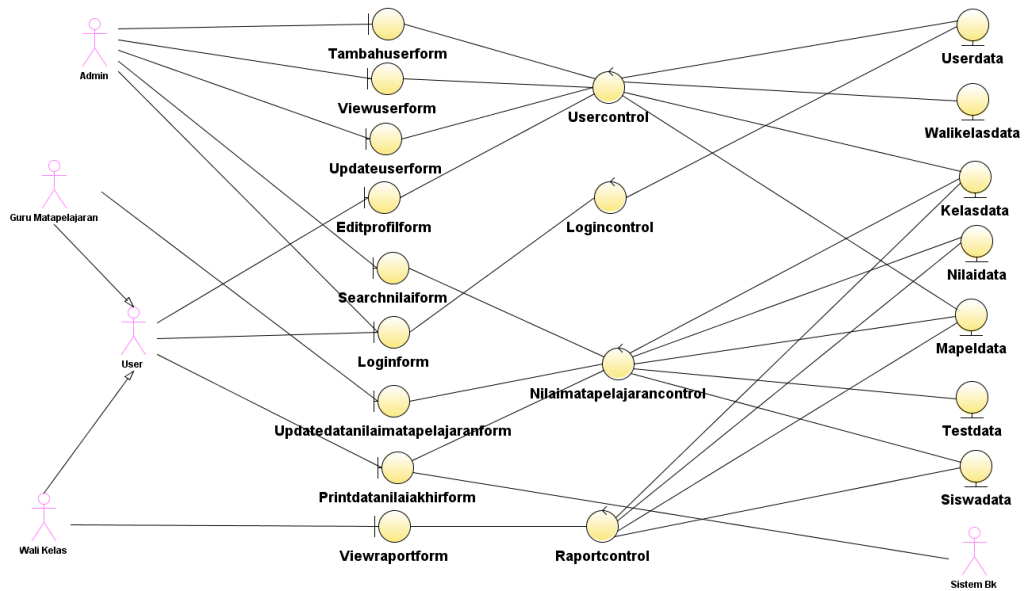
Tabel 3.9 Use case Naratif Search Nilai

Use case Name	Search Nilai
ID	6
Impotance Level	Medium
Primary Actor	Admin
Stakeholders and interest	Admin : melakukan <i>search</i> nilai.
Brief Description	<i>Use case</i> yang dapat diakses oleh admin untuk melakukan <i>search</i> data nilai matapelajaran.
Trigger	Admin dapat melakukan <i>search</i> nilai pada semua nilai siswa untuk mencari data nilai akhir dari siswa.
Type	<i>Internal</i>
Relationship	<p>Assosiation : Admin</p> <p>Include : <i>view</i> data nilai akhir</p>

	Extend : Generalitation :
<i>Normal flow event</i>	<p>Aksi Guru Matapelajaran, Wali Kelas</p> <p>1. Memilih menu <i>view</i> nilai.</p> <p>Reaksi</p> <p>2. Menampilkan menu untuk memilih nilai.</p>
<i>Sub flow</i>	<p>2.1 Admin melakukan <i>search</i> nilai</p> <p>2.1.1 Sistem akan menampilkan semua nilai siswa yang dipilih kategorinya</p>
<i>Alternate/ Exceptional flow</i>	<p>4. Data tidak ada, <i>search</i> gagal</p> <p>5. Kembali ke langkah 1</p>

4. Class Diagram Tahap Analisis

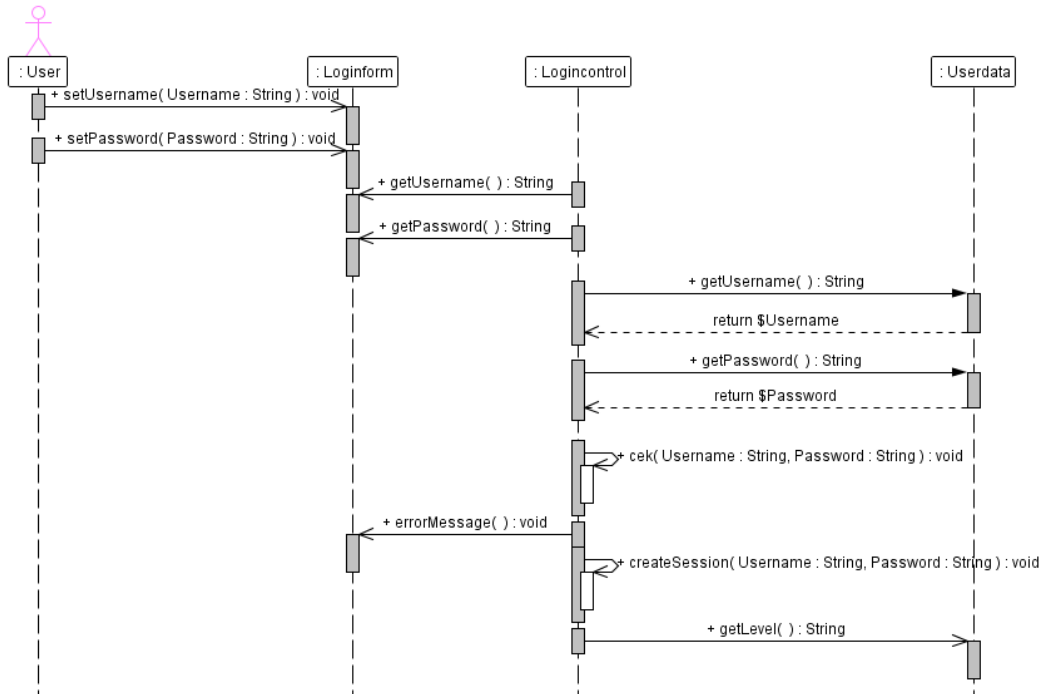
Class Diagram tahap *Analysis* merupakan diagram yang menjelaskan hubungan antar kelas secara sederhana. Misalnya seperti contoh diagram berikut :



Gambar 3.2 Class Diagram Tahap Analisis

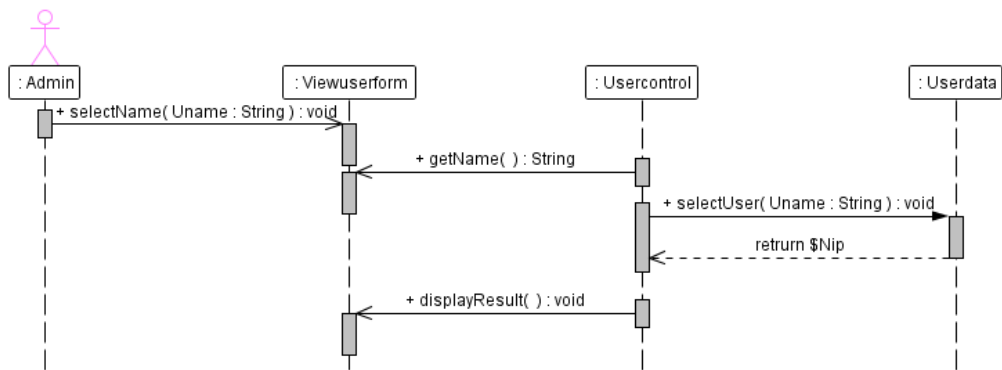
5. Sequence Diagram

a. Login



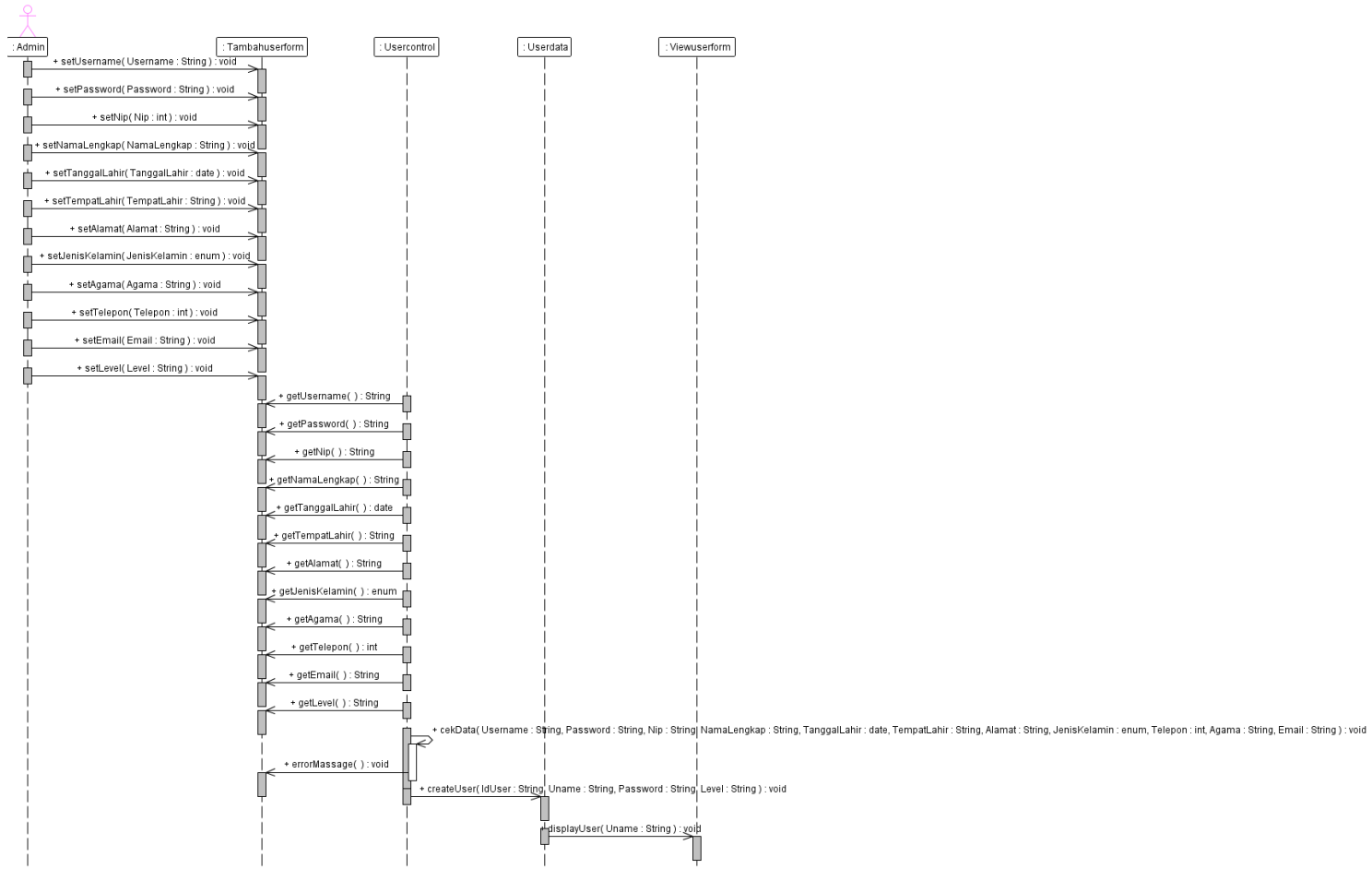
Gambar 3.3 Sequence Login

b. View User

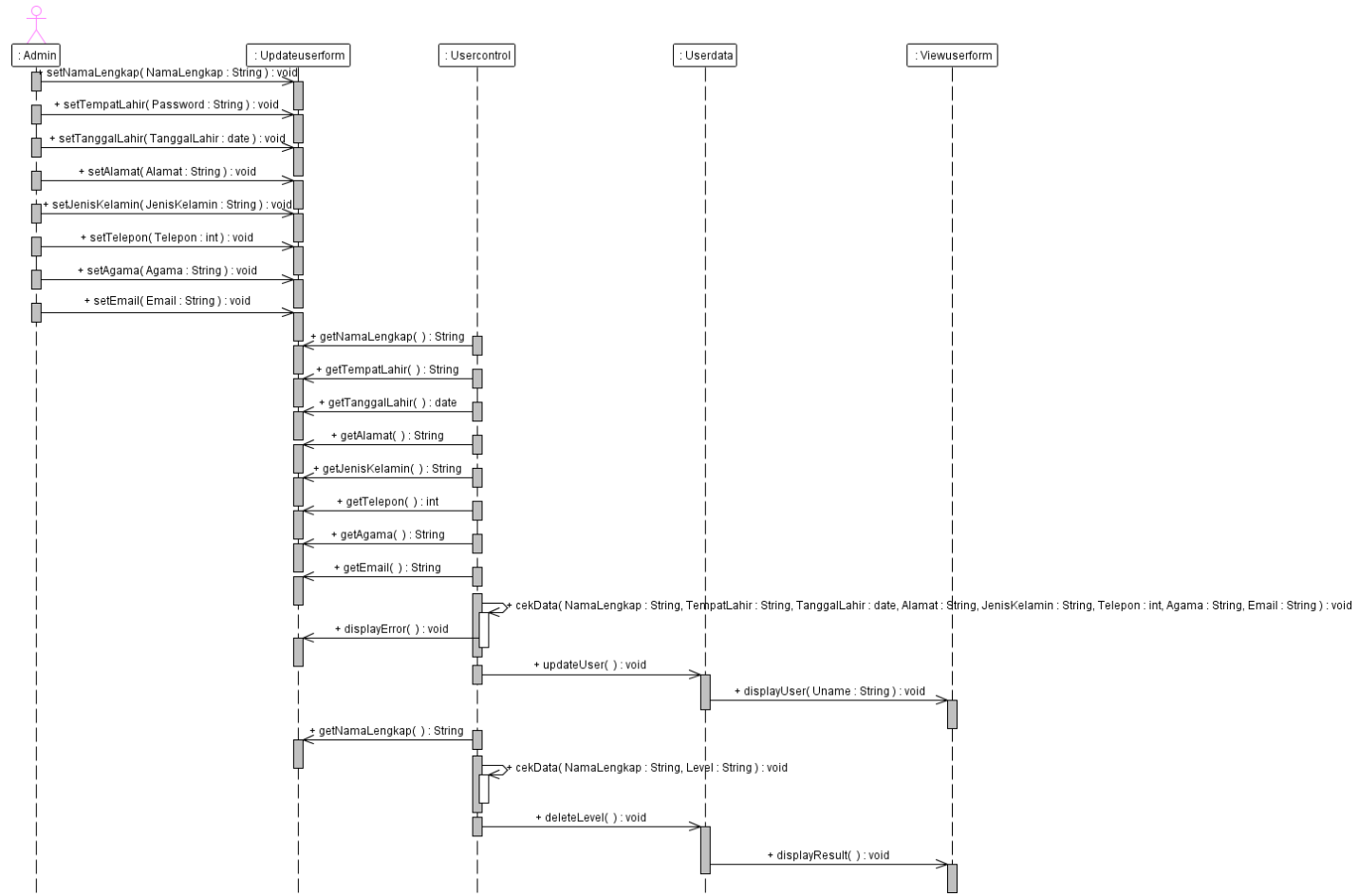


Gambar 3.4 Sequence View User

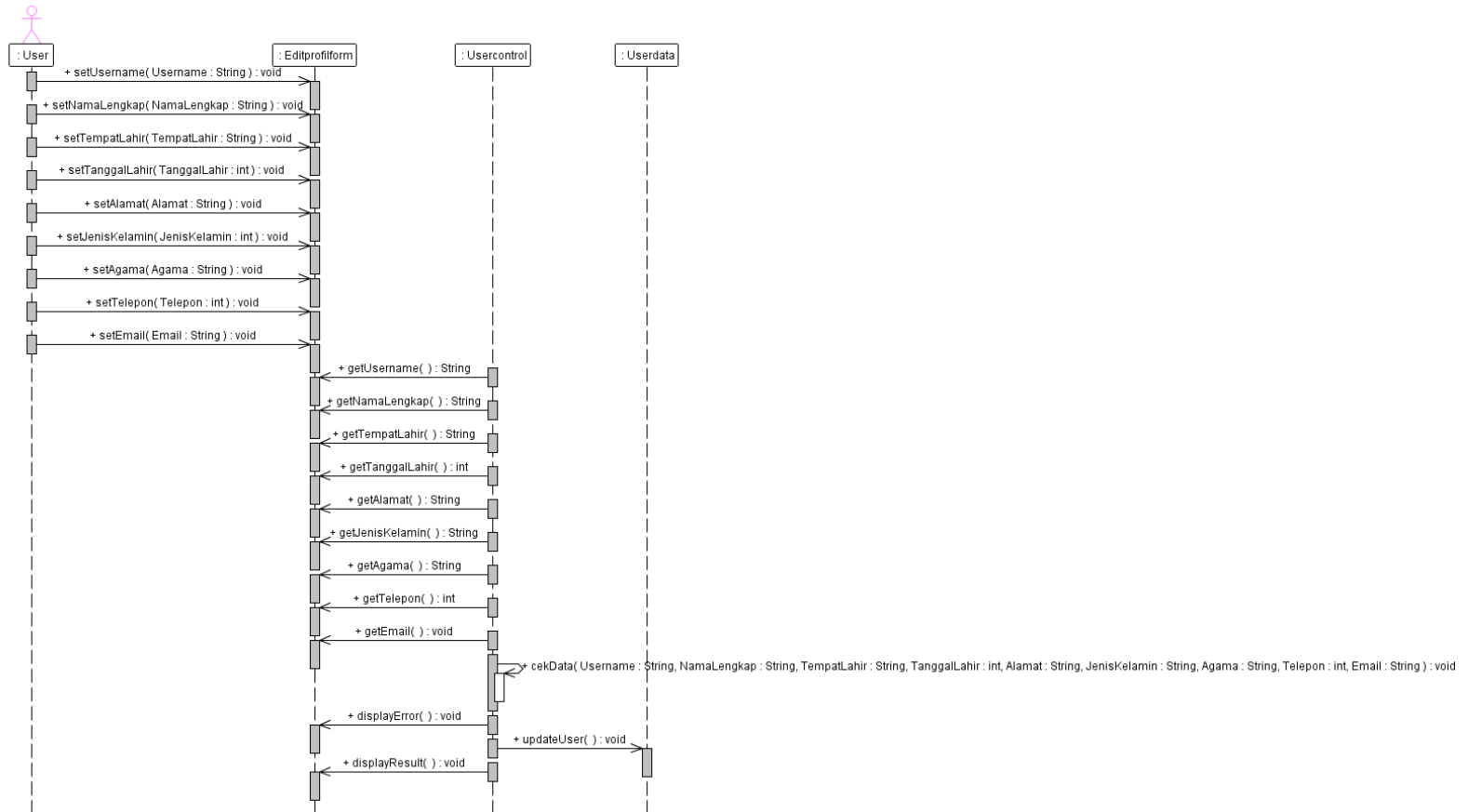
c. Tambah User



Gambar 3.5 Squence Tambah User

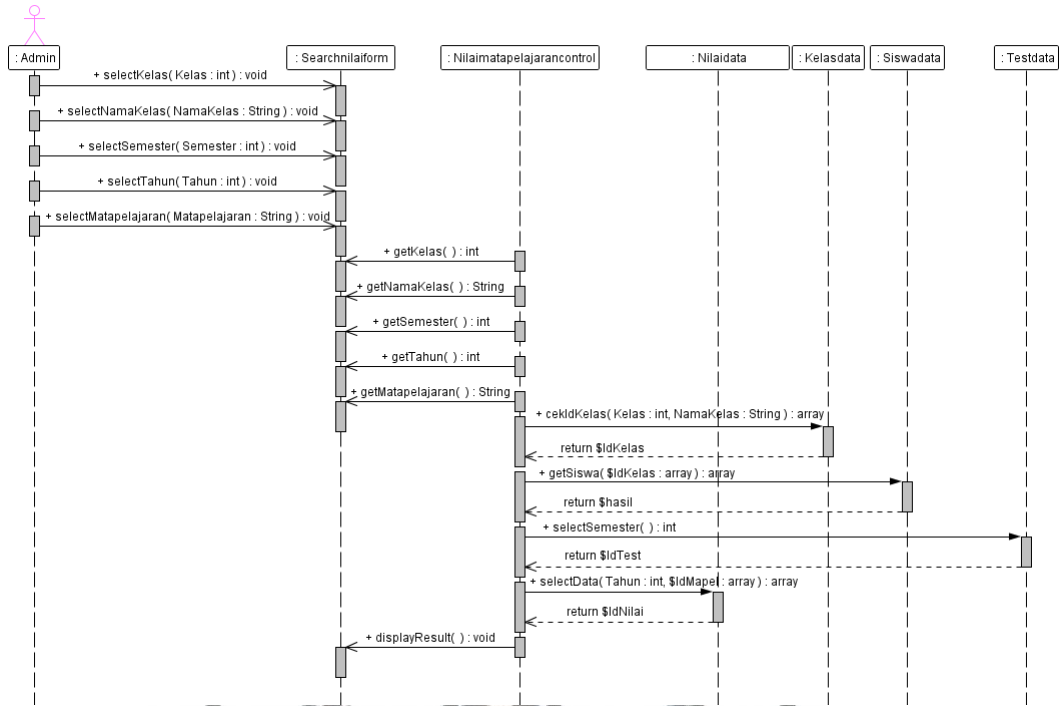
d. *Update User*Gambar 3.6 Sequence *Update User*

e. Mengelola Profil



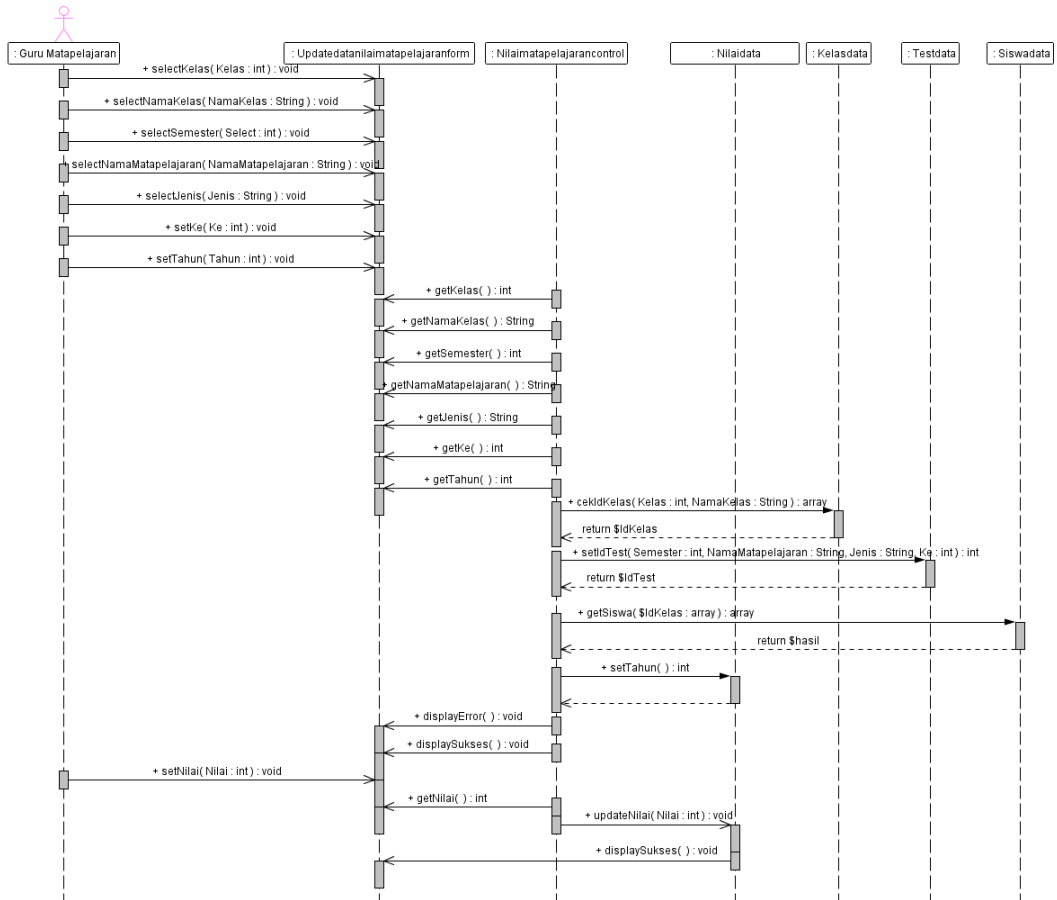
Gambar 3.7 Squence Mengelola Profil

f. Search Nilai



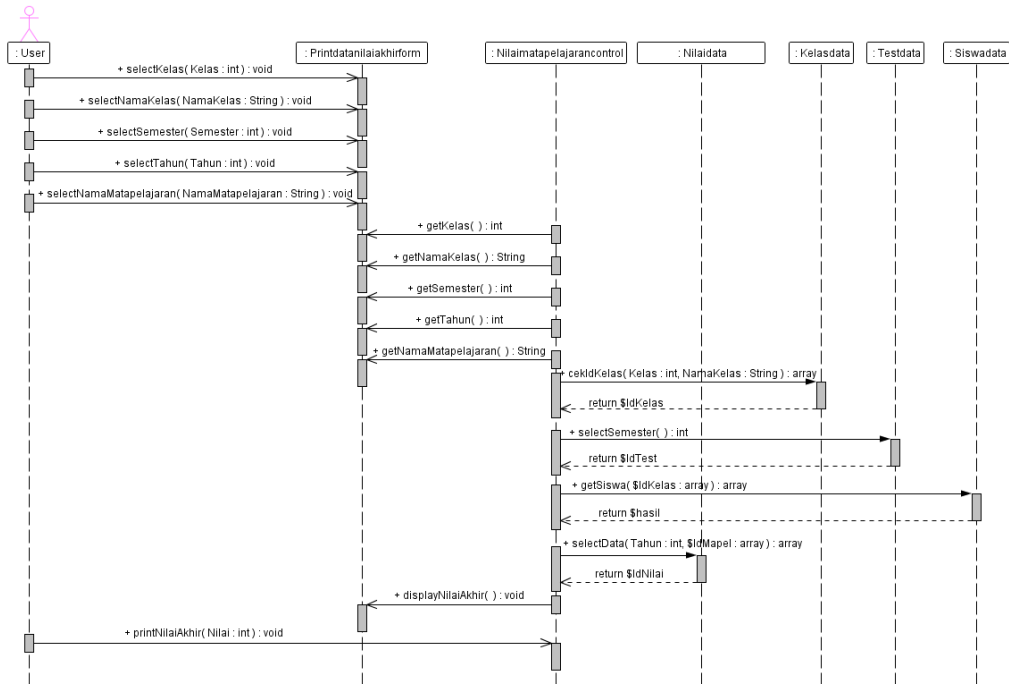
Gambar 3.8 Search Nilai

g. Update Nilai



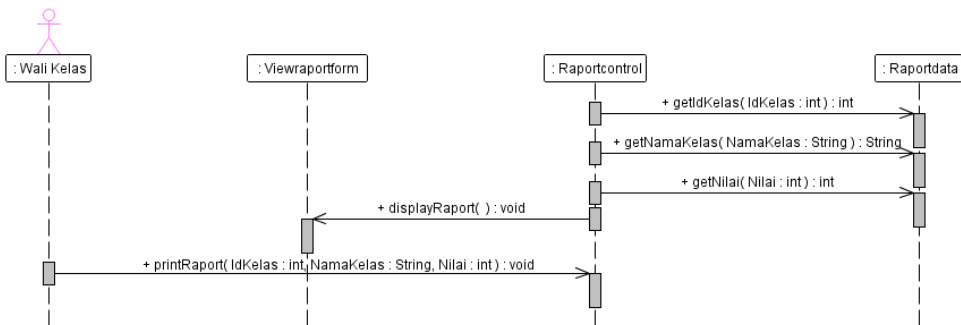
Gambar 3.9 Sequence Update Nilai

h. *Print Data Nilai Akhir*



Gambar 3.10 *Sequence Print Data Nilai Akhir*

i. *Print Raport*



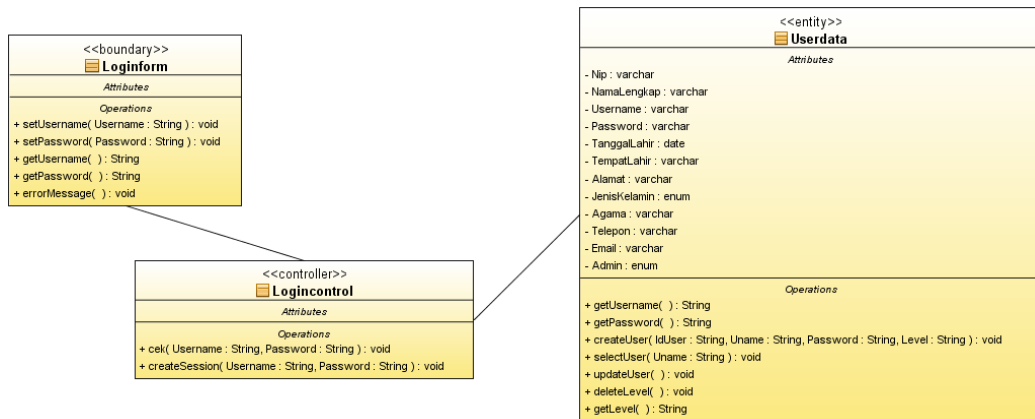
Gambar 3.11 *Sequence Print Raport*

6. *Class Element Diagram*

Class Element Diagram adalah diagram yang menjelaskan hubungan antar kelas secara lebih detail. Diagram ini juga menampilkan atribut dan operasi yang ada pada sistem. Sebagai contoh perhatikan diagram berikut :

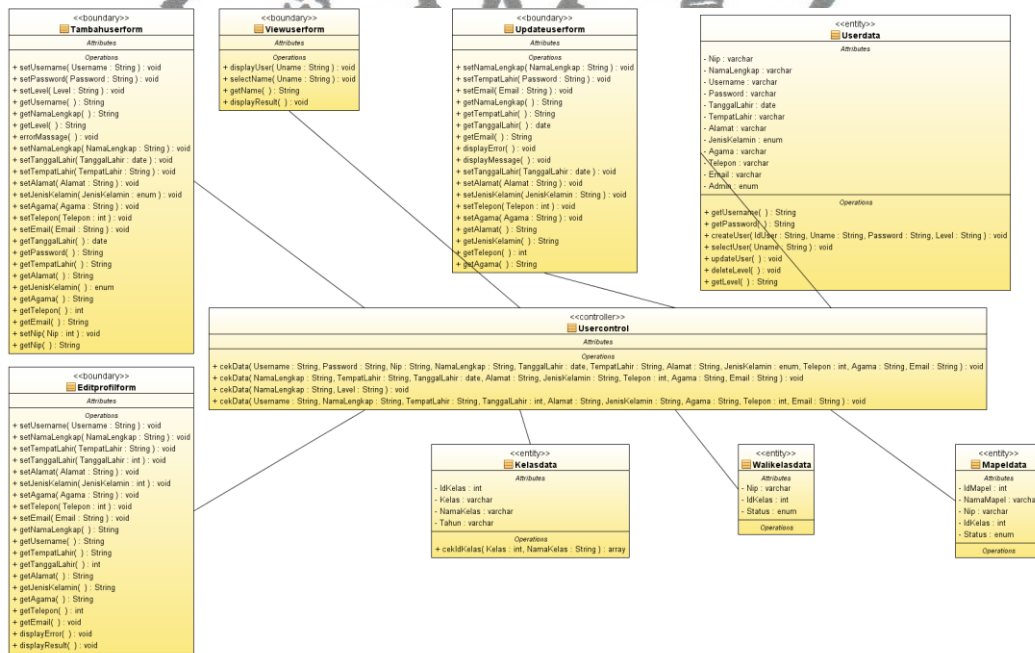
commit to user

1. Class Element Login



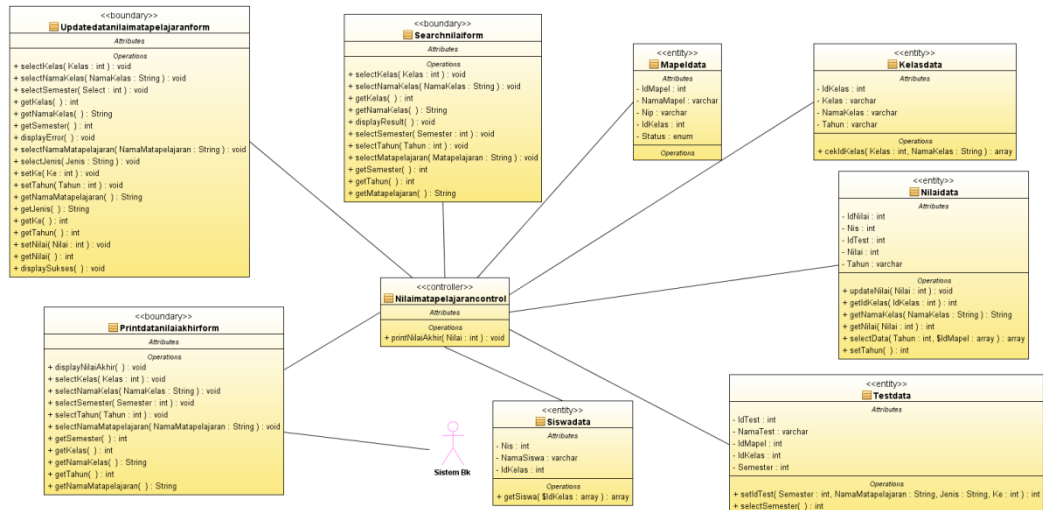
Gambar 3.12 Class Element Login

2. Class Element User



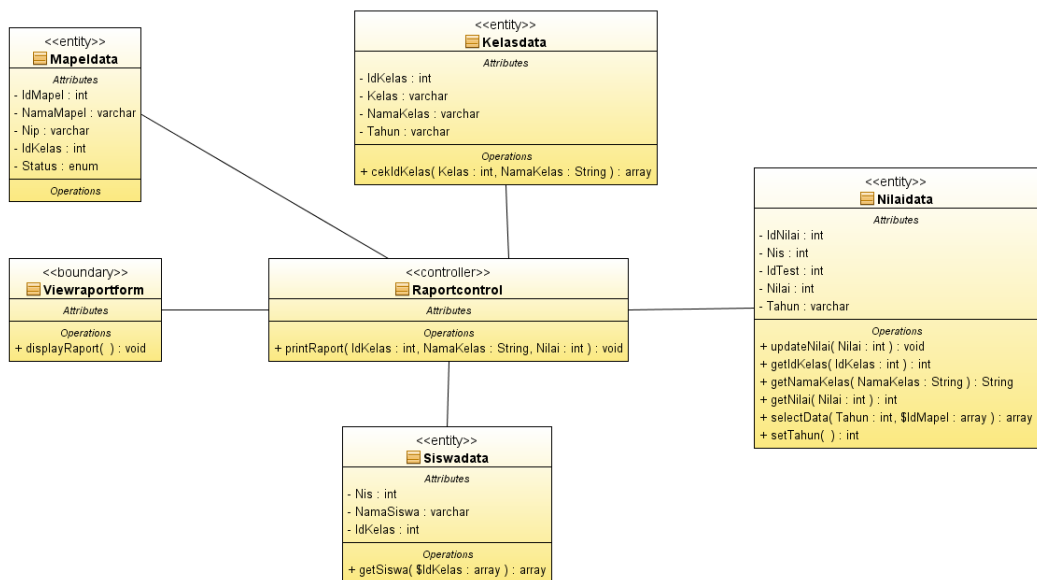
Gambar 3.13 Class Element User

3. Class Element Nilai Matapelajaran



Gambar 3.14 Class Element Nilai Matapelajaran

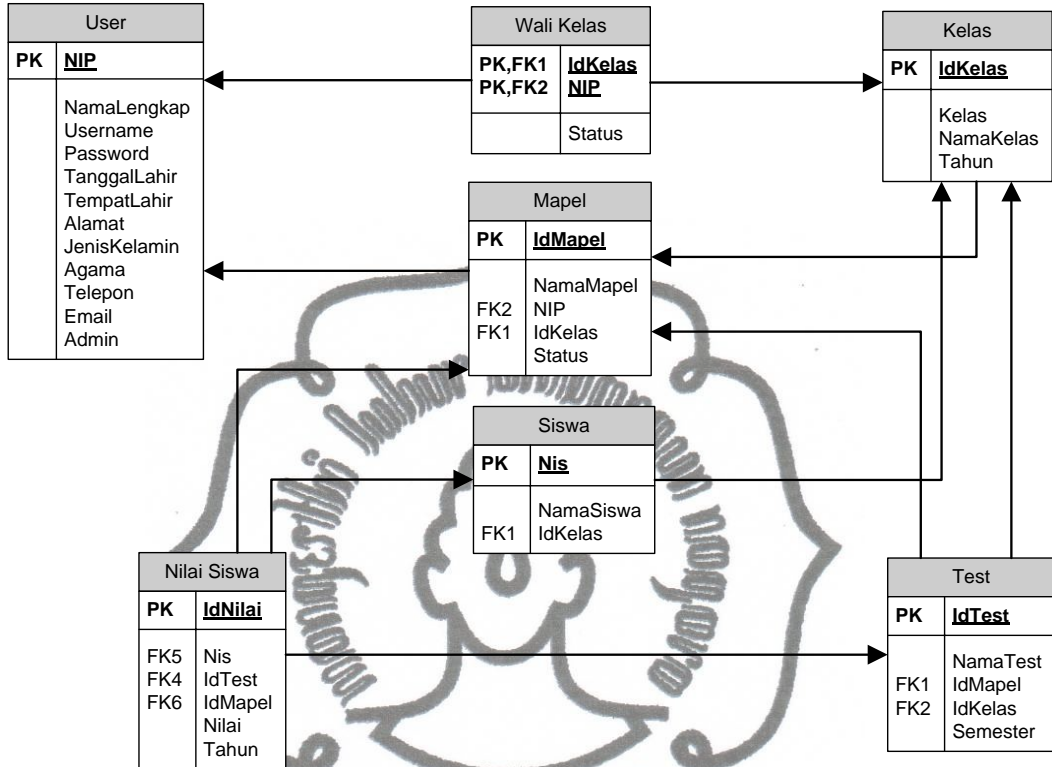
4. Class Element Report



Gambar 3.15 Class Element Report

C. Perancangan Database

1. Skema Diagram



Gambar 3.16 Skema Diagram

D. Perancangan Antarmuka (Interface)

1. Rancangan Form Login

Username

Password

Gambar 3.17 Rancangan Form Login

2. Rancangan Form Guru Matapelajaran

HEADER	Menu
	Profil
	Masukan Nilai
	Lihat Nilai
Wilujeng Rawuh :)	Logout

Gambar 3.18 Rancangan Form Guru Matapelajaran

3. Rancangan Form Wali Kelas

HEADER	Welcome Wali Kelas Team Penilaian
	Profil
	Masukan Nilai
	Lihat Nilai
	Lihat Raport
Wilujeng Rawuh :)	Logout

Gambar 3.19 Rancangan Form Wali Kelas

Wali kelas juga merupakan gurumatapelajaran juga, sehingga dia juga memiliki menu guru matapelajaran.

4. Rancangan Form Admin

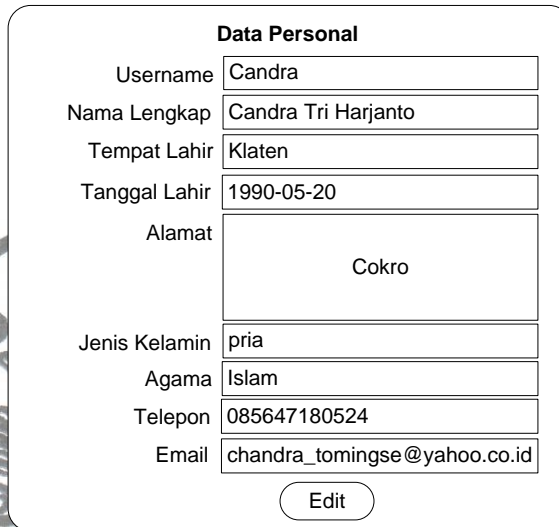
Gambar 3.20 Rancangan Form Admin

Admin merupakan gurumatapelajaran juga, sehingga admin juga memiliki menu guru matapelajaran, tapi admin juga berkemungkinan menjadi wali suatu kelas, jika itu terjadi maka otomatis menu admin akan tambah dengan menu seperti wali kelas.

5. Rancangan Form Profil

Gambar 3.21 Rancangan Form Profil

Semua *user* jika memilih menu profil maka akan ada 2 menu tambahan lagi, yaitu Edit Profil dan Ganti *Password*. Bila di pilih edit *user* maka akan muncul tampilan seperti di bawah ini :



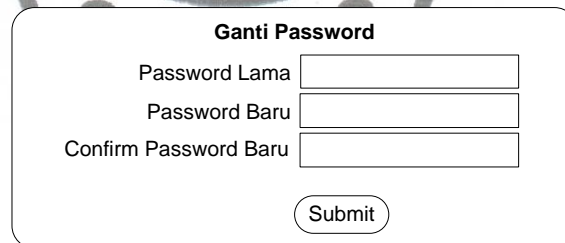
The image shows a web form titled "Data Personal" with the following fields and values:

Username	Candra
Nama Lengkap	Candra Tri Harjanto
Tempat Lahir	Klaten
Tanggal Lahir	1990-05-20
Alamat	Cokro
Jenis Kelamin	pria
Agama	Islam
Telepon	085647180524
Email	chandra_tomingse@yahoo.co.id

Below the form is an "Edit" button.

Gambar 3.22 Rancangan Form Edit Profil

Kalau di pilih ganti *password* maka akan muncul tampilan seperti dibawah ini :



The image shows a web form titled "Ganti Password" with the following fields:

Password Lama	<input type="text"/>
Password Baru	<input type="text"/>
Confirm Password Baru	<input type="text"/>

Below the form is a "Submit" button.

Gambar 3.23 Rancangan Form Ganti *Password*

6. Rancangan Form Input Nilai

The wireframe shows a user interface for entering grades. It features a header section, a main form area with several dropdown menus for selection (Semester, Kelas, Nama Kelas, Nama Mapel, Jenis) and two text input fields (Ke, Tahun Pelajaran), and a submit button labeled 'Input'. A sidebar on the right contains a welcome message and navigation buttons: 'Profil', 'Masukan Nilai', 'Lihat Nilai', and 'Logout'.

Gambar 3.24 Rancangan Form Select Input Nilai

Form diatas digunakan untuk memilih kelas yang diampunya, maka bila benar pilihanya guru matapelajaran tersebut dengan kelas yang duampunya akan muncul form seperti :

This wireframe illustrates the next step in the process. The main form area now displays a list of names, each with a corresponding checkbox for selection. The names listed are: Andi Irawan, Budi Hartanto, Bobby Kingdom, Carly Sitohang, Cika Uya, Dodik Item, Diki Raharjo, Dikut Wijaya, Endah Partawi, Farida Pasha, and Fuad Ardi. A 'Submit' button is located at the bottom of the form. The sidebar remains the same as in the previous wireframe.

commit to user
Gambar 3.25 Rancangan Form Input Nilai

7. Rancangan Form Lihat Nilai

Gambar 3.26 Rancangan Form Select Lihat Nilai

Form diatas digunakan untuk memilih kelas yang diampunya, maka bila benar pilihanya guru matapelajaran tersebut dengan kelas yang duampunya akan muncul form seperti :

Nilai Mata Pelajaran					
Nis	KD	Nilai	Edit	Nilai Akhir	Semester
100000001	1	90	Edit	90	1
100000002	1	80	Edit	80	1
100000003	1	85	Edit	90	1
100000004	1	95	Edit	85	1
100000005	1	85	Edit	80	1
100000006	1	75	Edit	75	1
100000007	1	80	Edit	70	1
100000008	1	90	Edit	90	1
100000009	1	70	Edit	80	1
100000010	1	80	Edit	85	1
100000011	1	90	Edit	95	1
100000012	1	100	Edit	90	1

<< 1 | 2 | 3 >> Print

Gambar 3.27 Rancangan Form Lihat Nilai

commit to user

8. Lihat Raport

HEADER

Welcome
Guru Matapelajaran
Team Penilaian

Raport Kelas _____

Nis : 100000001

Nama : Andi Irawan

Nama Matapelajaran	Nilai
Fisika	90
Biologi	85
Kimia	80
Matematika	75
Agama Islam	80
Bahasa Indonesia	90
Bahasa Inggris	95
Sejarah	80
Bahasa Jawa	70
Rata – Rata	82,78

Print

<< 1 | 2 | 3 >>

Profil

Masukan Nilai

Lihat Nilai

Logout

Gambar 3.28 Rancangan Form Lihat Raport

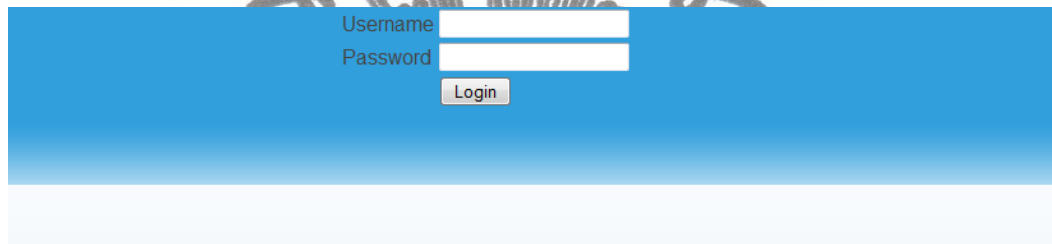
BAB IV

IMPLEMENTASI DAN HASIL PENGUJIAN

A. Implementasi Antar Muka

Struktur dan desain perancangan sistem informasi pengolahan nilai siswa pada sekolah menengah adalah sebagai berikut :

1. Form Login

A screenshot of a login form. The form has a blue gradient background. It contains three input fields: 'Username', 'Password', and a 'Login' button. The 'Username' and 'Password' fields are white with a blue border. The 'Login' button is a small, light blue rectangle with the text 'Login' in black.

Gambar 4.1 Form Login

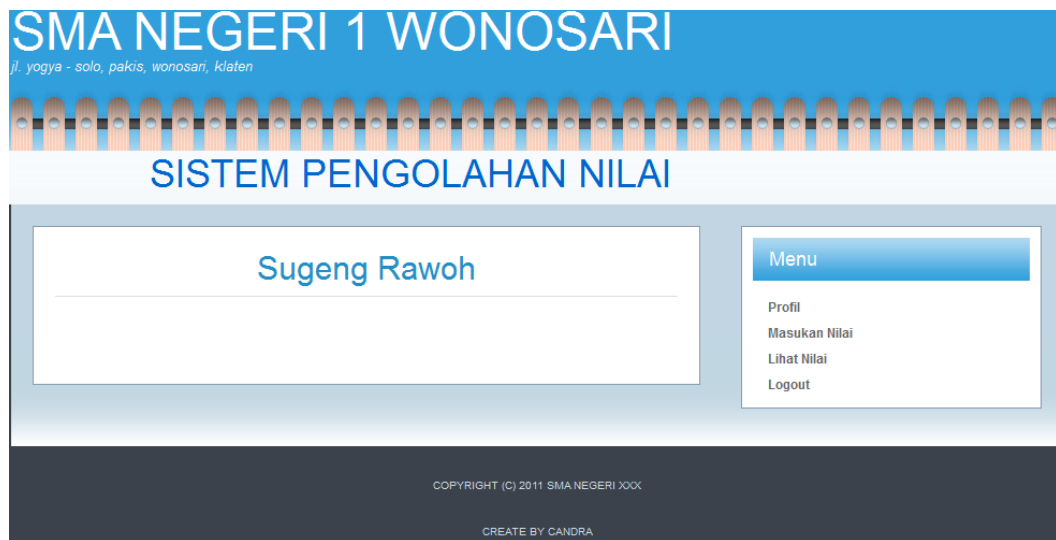
Form login di atas berfungsi untuk autentifikasi user atau pengguna program supaya user tidak dapat mengakses data tanpa mengetahui username dan password yang digunakan.

commit to user

2. Form Menu

Setelah user atau pengguna berhasil login, maka setiap user akan di berikan menu sesuai level masing – masing.

a. Form Menu User



Gambar 4.2 Form Menu User

Dimana setiap user disini adalah guru mata pelajaran.

b. Form Menu Wali Kelas



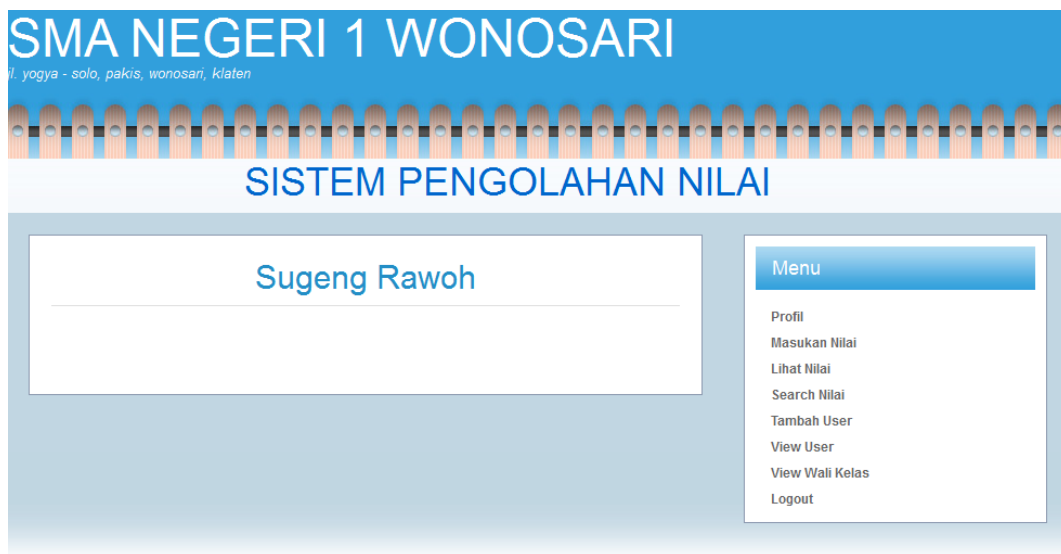
Gambar 4.3 Form Menu Wali Kelas

Menu wali kelas sama seperti menu user, yang berbeda adalah dimana wali kelas mendapatkan hak untuk melihat Raport, sedangkan user/guru

commit to user

matapelajaran tidak mempunyai, ini dikarenakan wali kelas juga merupakan salah satu user/matapelajaran.

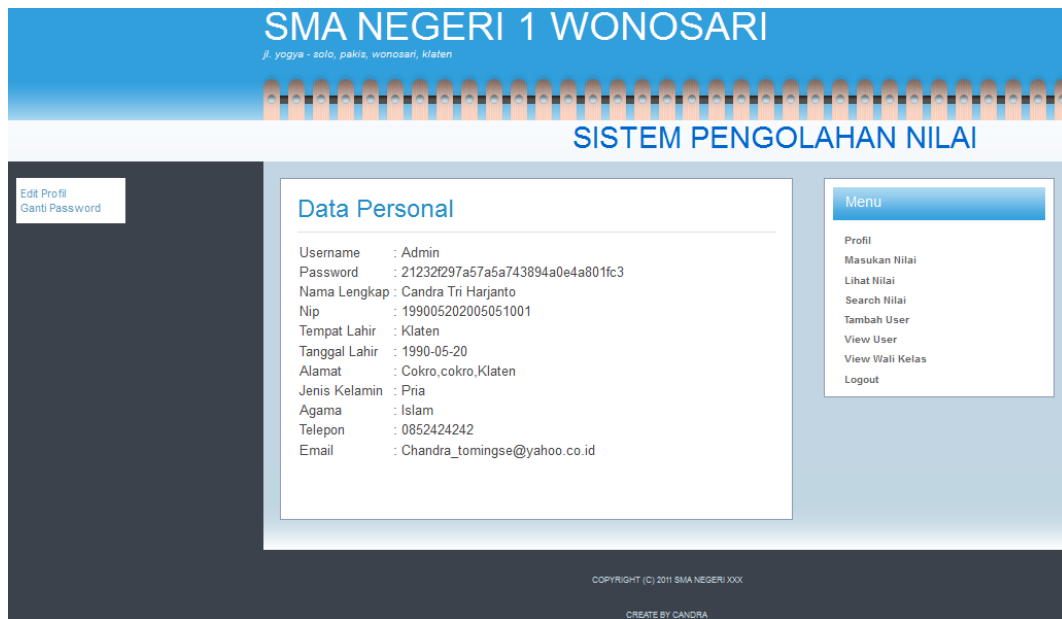
c. Form Menu Admin



Gambar 4.4 Form Menu Admin

Menu admin mirip dengan menu user, yang berbeda disini adalah admin memiliki menu tambahan kusus admin, dimana admin sendiri berasal dari user/guru mata pelajaran, sehingga admin juga memiliki menu user, tapi dalam kasus lain admin juga dapat menjadi wali kelas juga, tapi dalam hal ini tidak.

3. Form Profil



Gambar 4.5 Form Profil

Semua user jika memilih menu profil maka akan ada 2 menu tambahan lagi, yaitu Edit Profil dan Ganti Password. Bili di pilih edit user maka akan muncul tampilan seperti di bawah ini :



Gambar 4.6 Form Edit Profil

commit to user

Kalau di pilih ganti password maka akan muncul tampilan seperti dibawah ini :

The screenshot shows the 'Ganti Password' form within the 'SISTEM PENGOLAHAN NILAI' interface. The form includes three input fields: 'Password Lama', 'Password Baru', and 'Confirm Password Baru', each followed by a colon and a text box. Below these fields is a 'Ganti Password' button. To the right is a 'Menu' sidebar with options: Profil, Masukan Nilai, Lihat Nilai, Search Nilai, Tambah User, View User, View Wali Kelas, and Logout. The footer contains 'COPYRIGHT (C) 2011 SMA NEGERI 1 WONOSARI' and 'CREATE BY CANDRA'.

Gambar 4.7 Form Ganti Profil

4. Form Input Nilai

The screenshot shows the 'Tambah Nilai' form within the 'Sistem Pengolahan Nilai' interface. The form includes several input fields: 'Semester' (dropdown), 'Kelas' (dropdown), 'Nama Kelas' (dropdown), 'Nama Mapel' (dropdown), 'Jenis' (dropdown), 'Ke' (text), and 'Tahun Pelajaran' (text). Below these fields is an 'Input' button. To the right is a 'Menu' sidebar with options: Profil, Masukan Nilai, Lihat Nilai, Search Nilai, Tambah User, View User, View Wali Kelas, and Logout. The footer contains 'COPYRIGHT (C) 2011 SMA NEGERI 1 WONOSARI' and 'CREATE BY CANDRA'.

Gambar 4.8 Form Select Input Nilai

Form diatas digunakan untuk memilih kelas yang diampunya, maka bila benar pilihanya guru matapelajaran tersebut dengan kelas yang diampunya akan muncul form seperti dibawah ini:

NIS	Nama	Nilai
10001	Anton Budiman	<input type="text"/>
10002	Arie Sahri	<input type="text"/>
10003	Raharjo	<input type="text"/>
10004	Bambang Sulisty	<input type="text"/>
10005	Didik Winardi	<input type="text"/>
10006	Siti Aminah	<input type="text"/>
10007	Kartika Putri	<input type="text"/>
10008	Andi Setiawan	<input type="text"/>
10009	Wahyu Ningsih	<input type="text"/>
10010	Ambar Sulistyowati	<input type="text"/>
10011	Siti Aminah	<input type="text"/>
10012	Moh. Aji	<input type="text"/>
10013	Enggar Susanti	<input type="text"/>
10014	Dafit Wojayanto	<input type="text"/>
10015	Aris Munandar	<input type="text"/>
10016	Kristian Diki	<input type="text"/>
10017	Gangsar Apriyanto	<input type="text"/>
10018	Eko Prasetyo	<input type="text"/>
10019	Amien Zakarsieh	<input type="text"/>

Menu

- Profil
- Masukan Nilai
- Lihat Nilai
- Search Nilai
- Tambah User
- View User
- View Wali Kelas
- Logout

Gambar 4.9 Form Input Nilai

5. Form Lihat Nilai

SMA NEGERI 1 WONOSARI
jl. yogya - solo, pakis, wonosari, klaten

Sistem Pengolah

View Nilai

Kelas :

Nama Kelas :

Tahun :

Nama Mapel :

Menu

- Profil
- Masukan Nilai
- Lihat Nilai
- Search Nilai
- Tambah User
- View User
- View Wali Kelas
- Logout

COPYRIGHT (C) 2011 SMA NEGERI 1 WONOSARI
CREATE BY CANDRA

Gambar 4.10 Form Select Lihat Nilai

Form diatas digunakan untuk memilih kelas yang diampunya, maka bila benar pilihanya guru matapelajaran tersebut dengan kelas yang duampunya akan muncul form seperti dibawah ini :

SMA NEGERI 1 WONOSARI
J. Yogyakarta - Kab. Pekalongan, Wonosari (Akan)

Sistem Pengolahan Nilai

Nilai Matapelajaran

NIS	KD	Nilai	Edit	Nilai Akhir
10001	UL1.90	Edik	90.	
10002	UL1.80	Edik	80.	
10003	UL1.70	Edik	70.	
10004	UL1.90	Edik	90.	
10005	UL1.80	Edik	80.	
10006	UL1.70	Edik	70.	
10007	UL1.90	Edik	90.	
10008	UL1.80	Edik	80.	
10009	UL1.70	Edik	70.	
10010	UL1.90	Edik	90.	
10011	UL1.80	Edik	80.	
10012	UL1.70	Edik	70.	
10013	UL1.90	Edik	90.	
10014	UL1.80	Edik	80.	
10015	UL1.70	Edik	70.	
10016	UL1.90	Edik	90.	
10017	UL1.80	Edik	80.	
10018	UL1.70	Edik	70.	
10019	UL1.90	Edik	90.	
10020	UL1.80	Edik	80.	
10021	UL1.70	Edik	70.	
10022	UL1.90	Edik	90.	
10023	UL1.80	Edik	80.	
10024	UL1.70	Edik	70.	
10025	UL1.90	Edik	90.	
10026	UL1.80	Edik	80.	
10027	UL1.90	Edik	90.	
10028	UL1.80	Edik	80.	
10029	UL1.70	Edik	70.	
10030	UL1.90	Edik	90.	
10031	UL1.80	Edik	80.	
10032	UL1.90	Edik	90.	
10033	UL1.80	Edik	80.	
10034	UL1.70	Edik	70.	
10035	UL1.90	Edik	90.	
10036	UL1.80	Edik	80.	
10037	UL1.70	Edik	70.	
10038	UL1.90	Edik	90.	
10039	UL1.80	Edik	80.	
10040	UL1.70	Edik	70.	

Menu

- Profil
- Masukan Nilai
- Lihat Nilai
- Search Nilai
- Tambah User
- View User
- View Wall Kaba
- Logout

COPYRIGHT (C) 2011 SMA NEGERI 1 WONOSARI
CREATE BY CANDRIA

Gambar 4.11 Form Lihat Nilai

commit to user

6. Form Lihat Raport

SMA NEGERI 1 WONOSARI
Jl. yogya - solo, pakis, wonosari, klaten

SISTEM PENGOLAHAN NILAI

Pilih Semester

Semester :

Menu

- Profil
- Masukan Nilai
- Lihat Nilai
- Lihat Raport
- Logout

COPYRIGHT (C) 2011 SMA NEGERI 1 WONOSARI
CREATE BY CANDRA

Gambar 4.12 Form Select Lihat Raport

Form diatas digunakan untuk memilih semester, jika wali kelas telah memilih semester muncul form seperti dibawah ini :

SMA NEGERI 1 WONOSARI

Jl. Yogyakarta - Solo, Pakis, Wonosari, Klaten

SISTEM PENGOLAHAN NILAI

Raport

Raport XI - IPA1

Semester	= 1
NIS	= 10001
Nama	= Anton Budiman
Matematika	= 85
Bahasa Indonesia	= 60
Biologi	= 90
Fisika	= 60
Kimia	= 65
Pendidikan Jasmani	= 80
Pendidikan Agama	= 90
Bahasa Inggris	= 80
Bahasa Jawa	= 70
Pendidikan Kewarganegaraan	= 80
Seni Musik	= 80
Rata - Rata	= 76.36

No	Aspek Yang Dinilai	Keterangan
1	Kedisiplinan	
2	Kebersihan	
3	Kesehatan	
4	Tanggungjawab	
5	Sopan Santun	
6	Percaya Diri	
7	Kompetitif	
8	Hubungan Sosial	
9	Kejujuran	
10	Pelaksanaan ibadah Ritual	

[Print](#)

1 2 3 > Last >

Menu

- [Profil](#)
- [Masukan Nilai](#)
- [Lihat Nilai](#)
- [Lihat Raport](#)
- [Logout](#)

Gambar 4.13 Form Lihat Raport

B. Evaluasi Program

Tabel 4.1 Evaluasi Program

<i>Use Case</i>	Input	Proses	Output yang diharapkan	Hasil
Login	Username dan Password	Sistem mengecek Username dan Password yang dimasukkan dengan database.	<ol style="list-style-type: none"> 1. Muncul pesan kesalahan jika Username dan Password tidak sesuai dengan database. 2. Masuk ke dalam form user sesuai dengan level jika Username dan Password sesuai dengan database. 	<p>benar</p> <p>benar</p>
Mengelola User	Data User	Sistem melakukan pengecekan terhadap Data User yang dimasukkan.	<ol style="list-style-type: none"> 1. Muncul pesan kesalahan jika Data User ada yang belum diisi/sesuai dengan sistem. 2. Data user telah tersimpan di database jika Data User terisi semua/sesuai dengan sistem. 	<p>benar</p> <p>benar</p>
Add Data Nilai Matapelajaran	Semester, Kelas, Nama Kelas, Nama Matapelajaran, Jenis, Ke, Tahun Pelajaran, Nilai	Sistem melakukan pengecekan terhadap data yang diinputkan.	<ol style="list-style-type: none"> 1. Muncul pesan kesalahan jika data yang diinputkan salah. 2. Nilai disimpan di database jika data yang diinputkan benar. 	<p>benar</p> <p>benar</p>

View Data Nilai Akhir	Semester, Kelas, Nama Kelas, Nama Matapelajaran, Tahun	Sistem melakukan pengecekan terhadap data yang diinputkan.	<ol style="list-style-type: none"> 1. Muncul pesan kesalahan jika data yang diinputkan salah. 2. Nilai akan ditampilkan jika data yang diinputkan benar. 	benar benar
Print Raport	Semester	Sistem melakukan pengecekan terhadap data yang diinputkan.	<ol style="list-style-type: none"> 1. Muncul pesan kesalahan jika data yang diinputkan salah. 2. Raport akan ditampilkan jika data yang diinputkan benar. 	benar benar
Mengelola Profil	Data baru user	Sistem melakukan pengecekan terhadap data yang diinputkan.	<ol style="list-style-type: none"> 1. Muncul pesan kesalahan jika Data User ada yang belum diisi/sesuai dengan sistem. 2. Data user telah tersimpan di database jika Data User terisi semua/sesuai dengan sistem. 	benar benar
Search Nilai	Semester, Kelas, Nama Kelas, Nama Matapelajaran, Tahun	Sistem melakukan pengecekan terhadap data yang diinputkan.	<ol style="list-style-type: none"> 1. Muncul pesan kesalahan jika data yang diinputkan salah. 2. Nilai akan ditampilkan jika data yang diinputkan benar. 	benar benar

Deskripsi hasil pengujian :

1. Berhasil di implementasikan

- Login
- Mengelola user
- Add data nilai matapelajaran
- View data nilai akhir
- Print raport
- Mengelola profil
- Search nilai



BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan implementasi dan evaluasi sistem yang telah dibahas sebelumnya, maka dapat diambil kesimpulan bahwa :

Sistem telah berhasil dibangun dan telah mampu digunakan untuk mengelola dan menyampaikan informasi hasil pengolahan sistem yang berupa nilai akhir.

5.2. Saran

Saran yang dapat diberikan untuk pengembangan yang mungkin dilakukan pada sistem diantaranya sebagai berikut :

1. Penyempurnaan sistem raport yang mana dalam sistem ini sudah tersedia akan tetapi pada format penulisan belum sesuai dengan format penulisan raport yang asli.
2. Penambahan fasilitas untuk mencari nilai rata-rata raport terbesar sehingga dimana setiap tahun akan dapat diketahui siswa dengan nilai terbesar pada tiap kelas.
3. Menambahkan sistem instalasi untuk sistem ini, disertai dengan *setup*-nya
4. Penggunaan database yang lebih efisien untuk sistem jaringan, seperti postgresQL atau database lain yang lebih cepat proses aliran data yang dilakukan.

DAFTAR PUSTAKA

- Al Fatta, Hanif. 2007. *Analisis Dan Perancangan Sistem Informasi*. Andi: Yogyakarta.
- Julius, H. 2004. *Analisa Desain dan Pemrograman Berorientasi Obyek dengan UML dan Visual Basic.NET*. ANDI. Yogyakarta.
- Kristanto, H. 2004. *Konsep dan Perancangan Database*. ANDI: Yogyakarta
- Nugroho, B. 2004. *Pemrograman Web dengan PHP dan MySQL*. Gramedia : Yogyakarta
- Wiswakarma, Komang. 2010. *9 Langkah Menjadi Master Framework Codeigniter*. Loko Media: Yogyakarta
- Ruyanto. 2005. *Migrasi Microsoft SQL Sever dengan PostgreSQL*. PT Elex Media Komputindo: Jakarta
- Irwan, S. 2008. *Desain Web dengan CSS*.
- Rosihan, A.Y. 2008. *Dasar-dasar Pemrograman PHP*.
- Bima, S. 2001. *PHP dan MySQL untuk Web*. ANDI. Yogyakarta.
- Abdul, K. 1999. *Konsep dan Tuntunan Praktis Basis Data*. ANDI. Yogyakarta.
- Flerar, U. 2002. *Mengelola Database Server MySQL di Linux dan Windows*. ANDI. Yogyakarta.
- Pender, T.A. 2002. *UML Weekend Crash Course*. Wiley Publishing Inc.