LAPORAN TUGAS AKHIR

SISTEM INFORMASI UJIAN ONLINE BERBASIS WEB
(Studi Kasus: SMA Negeri 1 Gombong)

WEB BASED ONLINE EXAM INFORMATION SYSTEM
(Case Study: SMA Negeri 1 Gombong)

Disusun oleh:

DIANA ARI RAHMAWATI

14/369960/SV/07467

PROGRAM STUDI D3 KOMPUTER DAN SISTEM INFORMASI

DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA

SEKOLAH VOKASI

UNIVERSITAS GADJAH MADA

YOGYAKARTA

2017

LAPORAN TUGAS AKHIR

SISTEM INFORMASI UJIAN ONLINE BERBASIS WEB
(Studi Kasus: SMA Negeri 1 Gombong)

WEB BASED ONLINE EXAM INFORMATION SYSTEM
(Case Study: SMA Negeri 1 Gombong)

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Ahli Madya Komputer dan Sistem Informasi

Disusun oleh:

DIANA ARI RAHMAWATI

14/369960/SV/07467

PROGRAM STUDI D3 KOMPUTER DAN SISTEM INFORMASI

DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA

SEKOLAH VOKASI

UNIVERSITAS GADJAH MADA

YOGYAKARTA

2017

HALAMAN PENGESAHAN
LAPORAN TUGAS AKHIR

SISTEM INFORMASI UJIAN ONLINE BERBASIS WEB

disusun oleh:

DIANA ARI RAHMAWATI
14/369960/SV/07467

Diajukan untuk memenuhi salah satu syarat memperoleh gelar Ahli Madya
Pada Program Studi Diploma III Komputer Dan Sistem Informasi,
Departemen Teknik Elektro dan Informatika,
Sekolah Vokasi,
Universitas Gadjah Mada

Diterima dan disetujui oleh,

Ketua Penguji,	Sekretaris Penguji,
Drs. Janoe Hendarto, M.IKom	Ronald Adrian, S.T., M.Eng
NIP. 196301091988031001	NIU. 1120160178

Anggota Penguji/Dosen Pembimbing

Divi Galih Prasetyo P., S.Kom., M.Kom
NIU. 1120160032

Mengetahui,

Ketua Departemen Teknik Elektro dan Informatika	Ketua Program Studi Diploma III Komputer Dan Sistem Informasi
Ir. Lukman Subekti, M.T.	Sigit Priyanta, S.Si., M.Kom
NIP. 196210301993031002	NIP. 197704012002121002

PERNYATAAN

Dengan ini saya menyatakan bahwa laporan tugas akhir ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar Ahli Madya di suatu Perguruan Tinggi dan sepanjang pengetahuan saya, juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 7 Juni 2017

Diana Ari Rahmawati

PRAKATA

Alhamdulillah, segala puji bagi Allah yang telah melimpahkan rahmat dan hidayah-Nya sehingga tugas akhir dengan judul “Sistem Informasi Ujian Online Berbasis Web” dapat terlaksana dengan baik. Shalawat serta salam senantiasa kita panjatkan kepada junjungan Nabi Agung kita Nabi Muhammad shallallahualaihi wa sallam, para keluarganya, para sahabatnya, serta orang-orang yang mengukutinya hingga akhir zaman.

Selesainya penulisan tugas akhir ini tidak lepas dari bantuan dan dukungan dari berbagai pihak sehingga penulis dapat menyelesaikan tugas akhir ini :

1) Kedua orang tua penulis yang telah memberikan dukungan materil maupun non materil serta semangat dan doa yang tak pernah putus selama ini.
2) Saudara kandung penulis, Dwi Ristanti Rahmi yang selama ini memberikan semangat dan doa.
3) Ibu Divi Galih, M.Kom. selaku dosen pembimbing tugas akhir yang telah bersedia membimbing, memberikan arahan, dukungan dan motovasi kepada penulis.
4) Seluruh keluarga besar guru pengajar SMA Negeri 1 Gombong yang telah bersedia meluangkan waktu untuk menjadi narasumber penulis dalam rangka penelitian.
5) Teman-teman komsi, Ipul, Al, Ucup, Pras, Gavin yang sudah banyak membantu dalam proses pembuatan sistem tugas akhir.
6) Teman-teman komsi C2 yang sudah menjadi keluarga di masa perkuliahan.
7) Teman-teman Himakomsi yang sudah menjadi keluarga besar selama 3 tahun ini.
8) Teman-teman komsi 2014 yang tidak bisa disebutkan satu persatu.
9) Teman-teman Kontrakan Juweh yang sudah menjadi keluarga selama 1 tahun ini.
10) Segenap staf pengajar dan civitas akademik lingkungan Sekolah Vokasi dan Direktorat Kemahasiswaan Universitas Gadjah Mada.

11) Dan seluruh yang terlibat dalam pembuatan tugas akhir penulis.

Akhir kata penulis berharap tugas akhir ini dapat memberikan manfaat bagi kita semua, terutama bagi perkembangan ilmu pengetahuan serta perkembangan Ilmu Komputer dan Sistem Informasi.

Yogyakarta, 7 Juni 2017

Diana Ari Rahmawati

DAFTAR ISI

LAPORAN TUGAS AKHIR i
LAPORAN TUGAS AKHIR ii
HALAMAN PENGESAHAN iii
PERNYATAAN iv
PRAKATA v
DAFTAR ISI xii
DAFTAR GAMBAR xi
DAFTAR TABEL xvii
INTISARI xviii
ABSTRACT xix
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Batasan Masalah 3
1.4 Tujuan Penelitian 3
1.5 Manfaat Penelitian 3
1.6 Metodologi Penelitian 4
1.7 Sistematika Penulisan 5
BAB II TINJAUAN PUSTAKA 8
BAB III LANDASAN TEORI 10
3.1. Konsep Dasar Sistem 10
3.1.1. Karakteristik Sistem 10
3.1.2. Klasifikasi Sistem 12
3.2. Konsep Dasar Informasi 13
3.3.2. Siklus Informasi 13
3.3.2. Kualitas Informasi 13
3.3.2. Nilai Informasi 14
3.3. Konsep dasar Sistem Informasi 15
3.3.1. Komponen Sistem Informasi 16
3.3.2. Klasifikasi Sistem Infromasi 18

3.4. Siklus Hidup Sistem 18
3.5. Basis Data 20
3.5.1. Konsep Basis Data 21
3.5.2. Perancangan Basis Data 21
3.5.3. Model Basis Data 22
3.5.4. Relasi 24
3.5.5. Derajat Relasi (Kardinalitas) 24
3.5.6. Kunci 25
3.5.7. Query 25
3.6. UML (*Unified Modeling Language)* 26
3.6.1. *Use Case* Diagram 27
3.6.2. *Activity* Diagram 29
3.6.3. *Class* Diagram 31
3.7. Internet 34
3.8. Web 35
3.8.1. *Web Browser* 35
3.3.2. *Web Server* 35
3.9. *Software Framework* 35
3.10. Konsep MVC (Model, View, Controler) 36
3.11. Bahasa Pemrograman 37
3.11.1. HTML (Hyper Text Markup Language) 37
3.11.2. PHP (Hypertext Preprocessor) 38
3.11.3. CSS (*Cascading Style Sheet)* 39
3.11.4. Javascript 39
3.12. Perangkat Lunak Sistem 40
3.12.1. MySQL 40
3.12.3. Jquery 42
3.12.4. Bootstrap 43
3.3.2. XAMPP 43
3.12.6. Sublime Text 44
3.12.7. StarUML 44
3.12.8. Balsamiq Mockups 44

3.13 Pengertian Ujian 44
3.14 Pengertian Sistem Ujian Online 45
3.15 Sekilas Tentang SMA Negeri 1 Gombong 45
3.15.1 Sejarah SMA Negeri 1 Gombong 45
3.15.2 Visi dan Misi SMA Negeri 1 Gombong 46
BAB IV ANALISIS DAN PERANCANGAN SISTEM 48
4.1 Analisis Sistem 48
4.1.1 Analisis Sistem Yang Ada 48
4.1.2 Identifikasi Masalah 48
4.1.3 Analisis Kebutuhan Fungsional 49
4.1.4 Analisis Kebutuhan Non-fungsional 50
4.2 Perancangan Sistem 51
4.2.1 Perancangan Basis Data 51
4.2.2 Perancangan Proses 58
4.2.3 Perancangan Struktur Menu 71
4.2.4 Perancangan Antarmuka Pengguna 73
BAB V IMPLEMENTASI SISTEM 97
5.1 Implementasi Perangkat Lunak Pembangun 97
5.2 Implementasi Perangkat Keras Pembangun 97
5.3 Implementasi Basis Data 98
5.3.1 Pembuatan Basis Data 98
5.3.2 Pembuatan Tabel 98
5.4 Implementasi Antarmuka 105
5.4.1 Halaman *Login* 105
5.4.2 Halaman Utama (Dashboard) Admin 109
5.4.3 Halaman Utama (Dashboard) Guru 110
5.4.4 Halaman Utama (Dashboard) Siswa 111
5.4.5 Halaman Tahun Ajaran 113
5.4.6 Halaman Data Kelas 115
5.4.7 Halaman Data Guru 117
5.4.8 Halaman Data Siswa untuk Admin 121
5.4.9 Halaman Data Siswa untuk Guru 126

5.4.10 Halaman Data Mapel untuk Admin 127
5.4.11 Halaman Data Mapel Untuk Guru 131
5.4.12 Halaman Data Ujian untuk Admin..... 132
5.4.13 Halaman Data Ujian untuk Guru 134
5.4.14 Halaman Data Soal..... 135
5.4.15 Halaman Hasil Ujian..... 141
5.4.16 Halaman Soal Ujian Siswa..... 142
5.4.17 Halaman Riwayat Ujian Siswa..... 144
BAB VI PENGUJIAN DAN PEMBAHASAN SISTEM 145
5.1 Skenario Pengujian Sistem 145
5.2 Hasil Pengujian Sistem..... 146
5.2.1 Proses Login..... 146
5.2.2 Pengolahan Data Tahun Ajaran 148
5.2.3 Pengolahan Data Kelas 150
5.2.4 Pengolahan Data Guru 152
5.2.5 Pengolahan Data Siswa..... 155
5.2.6 Pengolahan Data Mapel 158
5.2.7 Pengolahan Data Ujian..... 162
5.2.8 Pengolahan Data Soal 164
5.2.9 Pengolahan Data Hasil Ujian 170
5.2.10 Pengelolaan Soal Ujian Siswa..... 171
5.2.11 Pengelolaan Data Riwayat Ujian Siswa..... 173
5.2.12 Pengelolaan Cetak Soal..... 173
5.2.13 Pengelolaan Cetak Hasil Ujian..... 174
BAB VII KESIMPULAN DAN SARAN 175
7.1 Kesimpulan..... 175
7.2 Saran..... 175
DAFTAR PUSTAKA..... 176

DAFTAR GAMBAR

Gambar 3.1 Siklus sistem informasi (Jogiyanto, 2005) 15
Gambar 3.2 Daur Hidup Sistem 20
Gambar 3.3 Konsep MVC 37
Gambar 3.4 Struktur dasar HTML 38
Gambar 3.5 Contoh script CSS 39
Gambar 3.6 Contoh script Javascript 40
Gambar 3.7 Ilustrasi sistem kerja framework CodeIgniter 41
Gambar 3.8 Contoh script Jquery..... 43
Gambar 4.1 Entity Relationship Diagram (ERD) 51
Gambar 4.2 Relasi Antar Tabel..... 58
Gambar 4.3 *Usecase Diagram* 59
Gambar 4.4 Activity Diagram Admin Mengelola Data Tahun Ajaran 60
Gambar 4.5 Activity Diagram Admin Mengelola Data Kelas 61
Gambar 4.6 Activity Diagram Admin Mengelola Data Guru 62
Gambar 4.7 Activity Diagram Admin Mengelola Data Siswa 63
Gambar 4.8 Activity Diagram Admin Mengelola Data Mapel 64
Gambar 4.9 Activity Diagram Admin Mengelola Data Ujian 65
Gambar 4.10 Activity Diagram Admin Meengelola Data Soal Ujian 66
Gambar 4.11 Activity Diagram Admin Mengelola Data Hasil Ujian..... 67
Gambar 4.12 Activity Diagram Guru Mengelola Data Soal Ujian..... 68
Gambar 4.13 Activity Diagram Guru Mengelola Data Hasil Ujian..... 69
Gambar 4.14 Activity Diagram Siswa Melakukan Ujian 70
Gambar 4.15 Activity Diagram Siswa melihat Riwayat Ujian 70
Gambar 4.17 Class Diagram 71
Gambar 4.18 Rancangan struktur menu admin..... 72
Gambar 4.19 Rancangan struktur menu guru 72
Gambar 4.20 Rancangan struktur menu siswa..... 73
Gambar 4.21 Rancangan Antarmuka Halaman Login 74
Gambar 4.22 Rancangan antarmuka Halaman Utama Admin 74
Gambar 4.23 Rancangan Antarmuka Halaman Utama Guru..... 75
Gambar 4.24 Rancangan Antarmuka Halaman Utama Siswa 75
Gambar 4.25 Rancangan Antarmuka Halaman Data Tahun Ajaran 76

Gambar 4.26 Rancangan Antarmuka Halaman Tambah Tahun Ajaran 77
Gambar 4.27 Rancangan Antarmuka Halaman Ubah Tahun Ajaran 77
Gambar 4.28 Rancangan Antarmuka Halaman Data Kelas 78
Gambar 4.29 Rancangan Antarmuka Halaman Tambah Kelas 78
Gambar 4.30 Rancangan Antarmuka Halaman Data Guru 79
Gambar 4.31 Rancangan Antarmuka Halaman Tambah Guru 80
Gambar 4.32 Rancangan Antarmuka Halaman Ubah Guru 80
Gambar 4.33 Rancangan Antarmuka Halaman Data Siswa oleh Admin..... 81
Gambar 4.34 Rancangan Antarmuka Halaman Data Siswa oleh Guru 82
Gambar 4.35 Rancangan Antarmuka Halaman Tambah Siswa 82
Gambar 4.36 Rancangan Antarmuka Halaman Ubah Siswa 83
Gambar 4.37 Rancangan Antarmuka Halaman Data Mapel untuk Admin..... 84
Gambar 4.38 Rancangan Antarmuka Halaman Data Mapel untuk Guru..... 84
Gambar 4.39 Rancangan Antarmuka Halaman Tambah Mapel 85
Gambar 4.40 Rancangan Antarmuka Halaman Ubah Mapel 85
Gambar 4.41 Rancangan Antarmuka Halaman Lihat Pengajar 86
Gambar 4.42 Rancangan Antarmuka Halaman Tambah Pengajar..... 86
Gambar 4.43 Rancangan Antarmuka Halaman Data Ujian untuk Admin 87
Gambar 4.44 Rancangan Antarmuka Halaman Data Ujian untuk Guru 88
Gambar 4.45 Rancangan Antarmuka Halaman Tambah Ujian..... 88
Gambar 4.46 Rancangan Antarmuka Halaman Ubah Ujian 89
Gambar 4.47 Rancangan Antarmuka Halaman Data Soal untuk Admin..... 89
Gambar 4.48 Rancangan Antarmuka Halaman Data Soal untuk Guru..... 90
Gambar 4.49 Rancangan Antarmuka Halaman Tambah Soal 90
Gambar 4.50 Rancangan Antarmuka Halaman Ubah Soal 91
Gambar 4.51 Rancangan Antarmuka Halaman Lihat Daftar Pertanyaan 92
Gambar 4.52 Rancangan Antarmuka Halaman Tambah Pertanyaan 92
Gambar 4.53 Rancangan Antarmuka Halaman Data Hasil Ujian..... 93
Gambar 4.54 Rancangan Antarmuka Halaman Riwayat Ujian Siswa 94
Gambar 4.55 Rancangan Antarmuka Halaman Soal Ujian Siswa 94
Gambar 4.56 Rancangan Antarmuka Halaman Nilai Hasil Ujian Siswa..... 95
Gambar 4.57 Rancangan Antarmuka Halaman Ubah Password..... 96
Gambar 5.1 Kode Program Pembuatan Basis Data 98
Gambar 5.2 Kode Program Pembuatan Tabel User 99

Gambar 5.3 Kode Proram Pembuatan Tabel Admin 99
Gambar 5.4 Kode Program Pembuatan Tabel Guru 100
Gambar 5.5 Kode Program Pembuatan Tabel Siswa 100
Gambar 5.6 Kode Program Pembuatan Tabel Kelas 100
Gambar 5.7 Kode Program Pembuatan Tabel Mapel 101
Gambar 5.8 Kode Program Pembuatan Tabel Tahun 101
Gambar 5.9 Kode Program Pembuatan Tabel Kelas Siswa 102
Gambar 5.10 Kode Program Pembuatan Tabel Guru Mapel 102
Gambar 5.11 Kode Program Pembuatan Tabel Ujian..... 103
Gambar 5.12 Kode Program Pembuatan Tabel Soal 103
Gambar 5.13 Kode Program Pembuatan Tabel Hasil Ujian 104
Gambar 5.14 Kode Program Pembuatan Tabel Pertanyaan..... 104
Gambar 5.15 Kode Program Pembuatan Tabel Opsi Jawaban 105
Gambar 5.16 Tampilan Halaman Utama Untuk *Login* 106
Gambar 5.17 Tampilan Halaman *Login* Pengelola 106
Gambar 5.18 Tampilan Halaman *Login* Siswa 107
Gambar 5.19 Kode Program Controller Halaman *Login* Pengelola 108
Gambar 5.20 Kode Program *Controller* Halaman *Login* Siswa 108
Gambar 5.21 Tampilan Halaman Utama Admin 110
Gambar 5.22 Kode Program Controller Halaman Utama Admin..... 110
Gambar 5.23 Tampilan Halaman Utama Guru 111
Gambar 5.24 Kode Program Controller Halaman Utama Guru..... 111
Gambar 5.25 Tampilan Halaman Utama Siswa 112
Gambar 5.26 Tampilan Halaman Utama Ujian Siswa 112
Gambar 5.27 Kode Program Controller Halaman Utama Ujian Siswa 113
Gambar 5.28 Tampilan Halaman Data Tahun Ajaran 113
Gambar 5.29 Tampilan Halaman Tambah Tahun Ajaran 114
Gambar 5.30 Kode Program Controller Halaman Tambah Tahun 114
Gambar 5.31 Tampilan Halaman Ubah Tahun Ajaran 115
Gambar 5.32 Kode Program Controller Halaman Ubah Tahun..... 115
Gambar 5.33 Tampilan Halaman Data Kelas 116
Gambar 5.34 Tampilan Halaman Tambah Data Kelas 116
Gambar 5.35 Kode Program Controller Halaman Tambah Data Kelas..... 117
Gambar 5.36 Tampilan Halaman Data Guru 117

Gambar 5.37 Tampilan Halaman Tambah Data Guru 118
Gambar 5.38 Kode Program *Controller* Halaman Tambah Data Guru 119
Gambar 5.39 Kode Program *Controller* Tambah Data Guru (Lanjutan)..... 119
Gambar 5.40 Tampilan Halaman Ubah Data Guru..... 120
Gambar 5.41 Kode Program *Controller* Halaman Ubah Data Guru..... 120
Gambar 5.42 Tampilan Halaman Lihat Detail Guru..... 121
Gambar 5.43 Kode Program *Controller* Halaman Lihat Detail Guru..... 121
Gambar 5.44 Tmapilan Halaman Data Siwa..... 122
Gambar 5.45 Kode Program *Controller* Halaman Tambah Siswa 123
Gambar 5.46 Kode Program *Controller* Halaman Tambah Siswa(Lanjutan).. 123
Gambar 5.47 Tampilan Halaman Tambah Data Siswa..... 124
Gambar 5.48 Tampilan Halaman Ubah Data Siswa 124
Gambar 5.49 Kode Program *Controller* Halaman Ubah Siswa..... 125
Gambar 5.50 Kode Program *Controller* Halaman Ubah Siswa(Lanjutan)..... 125
Gambar 5.51 Tampilan Halaman Lihat Detail Siswa 126
Gambar 5.52 Kode Program Controller Halaman Lihat Detail Siswa..... 126
Gambar 5.53 Tampilan Halaman Data Siswa Untuk Guru..... 127
Gambar 5.54 Tampilan Halaman Data Mapel 127
Gambar 5.55 Tampilan Halaman Tambah Data Mapel 128
Gambar 5.56 Kode Program Controller Halaman Tambah Mapel 129
Gambar 5.57 Kode Program Controller Halaman Ubah Data Mapel 129
Gambar 5.58 Kode Program Controller Halaman Lihat Pengajar Mapel..... 130
Gambar 5.59 Kode Program Controller Halaman Tambah Pengajar Mapel ... 131
Gambar 5.60 Tampilan Halaman Tambah Pengajar Mapel..... 131
Gambar 5.61 Tampilan Halaman Data Mapel untuk Guru..... 132
Gambar 5.62 Tampilan Halaman Data Ujian..... 132
Gambar 5.63 Kode Program *Controller* Halaman Data Ujian..... 133
Gambar 5.64 Kode Program *Controller* Halaman Data Ujian (Lanjutan)..... 133
Gambar 5.65 Kode Program *Controller* Halaman Ubah Data UJian..... 134
Gambar 5.66 Tampilan Halaman Data Ujian untuk Guru 135
Gambar 5.67 Tampilan Halaman Data Soal 135
Gambar 5.68 Kode Program *Controller* Halaman Tambah Data Soal 136
Gambar 5.69 Kode Program *Controller* Halaman Ubah Data Soal..... 137
Gambar 5.70 Kode Program *Controller* Halaman Daftar Pertanyaan 137

Gambar 5.71 Halaman Daftar Pertanyaan 138
Gambar 5.72 Kode Program *Controller* Halaman Tambah Pertanyaan 139
Gambar 5.73 Kode Program *Controller* Halaman Tambah Pertanyaan 139
Gambar 5.74 Halaman Ubah Pertanyaan 140
Gambar 5.75 Kode Program *Controller* Halaman Lihat Detail Soal 140
Gambar 5.76 Halaman Hasil Ujian 141
Gambar 5.77 Kode Program *Controller* Halaman Hasil Ujian 141
Gambar 5.78 Tampilan Halaman Soal Ujian Siswa 142
Gambar 5.79 Tampilan Halaman Soal Ujian Siswa (Lanjutan) 143
Gambar 5.80 Kode Program *Controller* Halaman Soal Ujian Siswa 143
Gambar 5.81 Tampilan Halaman Nilai Ujian 143
Gambar 5.82 Kode Program *Controller* Halaman Nilai Ujian 144
Gambar 5.83 Halaman Nilai Ujian 144
Gambar 6.1 Halaman Login 147
Gambar 6.2 Pemberitahuan *Login* Gagal 147
Gambar 6.3 Halaman Data Tahun Ajaran 148
Gambar 6.4 Halaman Tambah Tahun Ajaran 149
Gambar 6.5 Peringatan Tahun Ajaran Sudah Terdaftar 149
Gambar 6.6 Peringatan untuk Menonaktifkan Tahun Ajaran Sebelumnya 149
Gambar 6.7 Peringatan Konfirmasi Menonaktifkan Tahun Ajaran 150
Gambar 6.8 Peringatan Tahun Ajaran Tidak Dapat Dihapus 150
Gambar 6.9 Halaman Data Kelas 151
Gambar 6.10 Halaman Data Kelas 151
Gambar 6.11 Peringatan Kelas Tidak Dapat Dihapus 152
Gambar 6.12 Halaman Data Guru 152
Gambar 6.13 Pemberitahuan Nip Sudah Terdaftar Sebagai Guru 153
Gambar 6.14 Pemberitahuan Username Sudah Digunakan 153
Gambar 6.15 Pemberitahuan Kolom Masih Kosong 154
Gambar 6.16 Halaman Ubah Guru 154
Gambar 6.17 Halaman Detail Guru 154
Gambar 6.18 Peringatan guru tidak dapat dihapus 155
Gambar 6.19 Halaman Data Siswa 156
Gambar 6.20 Pemberitahuan Nis Sudah Terdaftar Sebagai Siswa 157
Gambar 6.21 Pemberitahuan Username Sudah Digunakan 157

Gambar 6.22 Pemberitahuan Kolom Masih Kosong 157
Gambar 6.23 Halaman Ubah Data Siswa..... 158
Gambar 6.24 Halaman Detail Siswa 158
Gambar 6.25 Halaman Data Mapel..... 159
Gambar 6.26 Pemberitahuan Id Mapel Sudah Terdaftar Sebagai Mapel..... 159
Gambar 6.27 Peringatan saat Hapus Mata Pelajaran 160
Gambar 6.28 Halaman Ubah Mapel..... 161
Gambar 6.29 Halaman Lihat Pengajar 161
Gambar 6.30 Halaman Tambah Pengajar 162
Gambar 6.31 Halaman Data Ujian 162
Gambar 6.32 Pemberitahuan Id Ujian Sudah Digunakan 163
Gambar 6.33 Halaman Ubah Ujian..... 163
Gambar 6.34 Peringatan Ujian Tidak Dapat Dihapus..... 164
Gambar 6.35 Halaman Data Soal..... 164
Gambar 6.36 Tampilan Aksi Input *Disable* Pada Halaman Data Soal 165
Gambar 6.37 Halaman Tambah Data Soal..... 165
Gambar 6.38 Halaman Ubah Data Soal 166
Gambar 6.39 Peringatan Hapus Data Soal..... 166
Gambar 6.40 Peringatan Tidak Dapat Menghapus Data Soal 167
Gambar 6.41 Halaman Daftar Pertanyaan 167
Gambar 6.42 Halaman Tambah Pertanyaan..... 168
Gambar 6.43 Halaman Tambah Pertanyaan (Lanjutan)..... 168
Gambar 6.44 Halaman Ubah Pertanyaan 169
Gambar 6.45 Halaman Lihat Detail Soal dan Pertanyaan..... 170
Gambar 6.46 Halaman Data Hasil Ujian..... 171
Gambar 6.47 Halaman Cetak Data Hasil Ujian 171
Gambar 6.48. Halaman Soal Ujian Siswa..... 172
Gambar 6.49. Halaman Soal Ujian Siswa (Lanjutan)..... 172
Gambar 6.50 Halaman Nilai Ujian Siswa 172
Gambar 6.51 Halaman Riwayat Ujian Siswa..... 173
Gambar 6.52 Tampilan Cetak Soal Ujian Dalam Print Window 174
Gambar 6.53 Tampilan Hasil Ujian Siswa Dalam Bentuk PDF 174

DAFTAR TABEL

Tabel 2.1 Perbandingan Tinjauan Pustaka 9
Tabel 3.1 Simbol Diagram E-R (Kadir, 1999) 23
Tabel 3.2 Simbol Diagram E-R (Kadir, 1999) (lanjutan) 24
Tabel 3.3 Simbol *use ase* diagram 28
Tabel 3.5 Simbol *activity* diagram 30
Tabel 3.6 Simbol *class* diagram 32
Tabel 3.7 Simbol *class* diagram (lanjutan) 33
Tabel 3.8 Simbol *class* diagram (lanjutan) 34
Tabel 4.1 Tabel User 52
Tabel 4.2 Tabel Admin 52
Tabel 4.3 Tabel Guru 52
Tabel 4.4 Tabel Siswa 53
Tabel 4.5 Tabel Ujian 53
Tabel 4.6 Tabel Tahun Ajaran 54
Tabel 4.7 Tabel Mapel 54
Tabel 4.8 Tabel Guru Mapel 54
Tabel 4.9 Tabel Kelas 55
Tabel 4.10 Tabel Kelas Siswa 55
Tabel 4.11 Tabel Hasil Ujian 56
Tabel 4.12 Tabel Soal 56
Tabel 4.13 Tabel Pertanyaan 57
Tabel 4.14 Tabel Opsi Jawaban 57
Tabel 6.1 Skenario Pengujian Sistem 145

INTISARI

SISTEM INFORMASI UJIAN ONLINE BERBASIS WEB

Oleh:

DIANA ARI RAHMAWATI

14/369960/SV/07467

SMA Negeri 1 Gombong merupakan Sekolah Menengah Atas di kota Kebumen. Pelaksanaan ujian di SMA Negeri 1 Gombong masih menggunakan metode konvesional yang memerlukan waktu, biaya dan tenaga yang cukup banyak. Proses ujian konvensional dan manual memungkinkan terjadinya kecurangan dan kebiasaan mencontek, lambatnya proses penilaian, serta pemborosan biaya kertas dalam penggandaan soal maupun lembar jawaban.

Berdasarkan permasalahan tersebut, perlu dibangun sebuah sistem yang mampu menangani pengelolaan ujian dari soal hingga nilai yang terkomputerisasi. Sistem Informasi Ujian Online ini dirancang menggunakan bahasa pemrograman PHP, *framework CodeIgniter, MySQL* untuk pengelolaan basis data dan *Javascript* dengan *framework Jquery*. Sistem pengelolaan ujian ini menyediakan beberapa fitur seperti pembuatan soal, cetak soal, proses ujian online, melihat hasil ujian, cetak hasil ujian.

Pengujian sistem dilakukan dengan menggunakan *blackbox*. Berdasarkan hasil pengujian dapat disimpulkan bahwa sistem yang telah dibangun dapat membantu guru dan siswa dalam proses pelaksanaan ujian di SMA Negeri 1 Gombong.

Kata Kunci : Sistem Informasi, Ujian Online, Codeigniter, PHP, MySQl

ABSTRACT

WEB BASED ONLINE EXAM INFORMATION SYSTEM

By:

DIANA ARI RAHMAWATI

14/369960/SV/07467

SMA Negeri 1 Gombong is a Senior High School in Kebumen City. Implementation of the exam in SMA Negeri 1 Gombong still use conventional methods that require time, cost and considerable energy. The conventional and manual test process allows cheating and cheating habits, slow assessment process, and waste of paper costs in duplication of questions and answer sheets.

Based on these problems, it is necessary to build a system capable of handling test management from the problem to the computerized value. Information Systems Online Exam is designed using PHP programming language, CodeIgniter framework, MySQL for database management and Javascript with Jquery framework. This exam management system provides several features such as problem-making, problem-solving, online exam processing, exam results, exam results.

System testing is done using blackbox. Based on the test results can be concluded that the system has been built to help teachers and students in the implementation of exams in SMA Negeri 1 Gombong.

Keyword : Informations Systems, Online Exam, Codeigniter, PHP, MySQl

BAB I
PENDAHULUAN

1.1 Latar Belakang

Teknologi informasi merupakan teknologi yang menghubungkan antara komputasi dan komunikasi untuk melakukan tugas-tugas informasi sehingga arus informasi dapat berjalan dengan baik. Teknologi informasi berkembang pesat di berbagai aspek kehidupan mulai dari personal hingga instansi. Dalam instansi negeri maupun swasta, teknologi informasi sangat dibutuhkan dalam optimalisasi segala proses yang berkaitan dengan pembangunan dan perbaikan sistem. Salah satu bentuk pengoptimalan tersebut adalah penerapan sistem informasi. Kriteria dalam sistem informasi antara lain adalah fleksibel, efektif dan efisien. SMA Negeri 1 Gombong adalah salah satu instansi di bidang pendidikan yang telah menggunakan teknologi informasi dalam berbagai aspek kegiatan dan ingin mengembangkan proses ujian menjadi lebih efisien dengan menggunakan sistem informasi.

Ujian Tengah Semester maupun Ujian Akhir Semester adalah kegiatan yang dilakukan oleh satuan pendidikan untuk mengukur pencapaian kompetensi peserta didik sebagai pengakuan prestasi belajar selama setengah atau satu semester di sekolah. Dalam melakukan ujian membutuhkan kertas dalam pelaksanaan ujian, serta melakukan audit penilaian secara manual yang biasanya dilakukan oleh guru, sehingga standart pengaplikasian ujian bisa berproses lama hingga dan memakan biaya.

Pelaksanaan ujian di SMA Negeri 1 Gombong masih menggunakan metode konvesional yang memerlukan waktu, biaya dan tenaga yang cukup banyak. Kegiatan ujian tengah semester atau ujian akhir semester biasanya diawali dari pembuatan soal dari guru bidang studi, soal yang dibuat oleh guru akan digandakan pihak akademis. Kemudian untuk mengetahui hasilnya, jawaban siswa akan dikumpulkan dan dikoreksi oleh guru bidang studi. Siswa tidak bisa langsung mengetahui hasil atau nilainya. Semua itu memerlukan waktu yang

cukup lama apalagi jumlah siswa di SMA Negeri 1 Gombong cukup banyak yaitu sekitar 27 kelas dengan jumlah seluruh siswa kurang lebih 1000 siswa. Selain itu, SMA Negeri 1 Gombong juga memiliki dua jurusan yaitu IPA(Ilmu Pengetauan Alam) dan IPS(Ilmu Pengetahuan Sosial) dengan jumlah guru sebanyak 54 orang. Sistem ujian konvensional yang sering dipakai dalam kegiatan akademik sering kali mempunyai banyak kendala. Faktor kecurangan dan kebiasaan mencontek merupakan kendala yang paling besar. Selain itu faktor lambatnya proses penilaian dalam ujian konvensional/tertulis dikarenakan pesertanya yang banyak sehingga dapat menghambat guru dalam memasukkan nilai kedalam raport. Pemborosan biaya kertas dan tinta dalam pembuatan soal dan penggandaan soal termasuk kedalam faktor negatif dalam ujian konvesional.

Namun seiring dengan berkembangnya teknologi informasi, sistem ujian konvensional bergeser menjadi sistem ujian online. Suatu isu yang menarik adalah format ujian online. Ujian online merupakan ujian yang dilakukan dengan alat bantu komputer. Ujian dikerjakan di depan komputer dengan format ujian yang biasanya pilihan ganda. Ketika ujian selesai dikerjakan, biasanya hasil dari ujian langsung dikeluarkan. Berdasarkan pelaksanaan ujian tersebut timbulah sebuah pemikiran, bagaimana caranya membuat proses pelaksanaan ujian online lebih mudah dan efisien, bermanfaat bagi guru dan siswa dengan memanfaatkan kemajuan teknologi komputer di SMA Negeri 1 Gombong. SMA Negeri 1 Gombong sudah menggunakan aplikasi Moodle yang digunakan untuk pelaksanaan ujian online mulai dari Ulangan Harian dan Tes Uji Coba Ujian Nasional. Tetapi sistem tersebut dirasa masih kurang *user friendly* dan tidak bisa menampung data ujian yang digunakan langsung seluruh siswa untuk Ujian Akhir Semester.

Mengacu pada hal tersebut, untuk itulah penulis merancang dan mengembangkan sebuah Sistem Informasi Ujian Online Berbasis Web (Studi Kasus: SMA Negeri 1 Gombong) yang diharapkan dapat membantu siswa dan guru dalam melaksanakan Ujian Akhir Semester maupun Ujian Tengah Semester

dengan efektif dan efisien serta mempermudah guru dalam mengoreksi hasil ujian siswa.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, dapat diperoleh sebuah rumusan masalah, yaitu bagaimana membangun sebuah sistem ujian online yang mampu mengatasi masalah dalam pelaksanaan ujian manual yang dapat diikuti oleh seluruh siswa di SMA Negeri 1 Gombong.

1.3 Batasan Masalah

Sistem informasi ujian online memiliki beberapa batasan dalam proses perancangannya yaitu:

1. Sistem Informasi Ujian Online hanya dapat digunakan oleh guru dan siswa dengan menggunakan akun yang telah diaktifkan oleh admin.
2. Sistem ini hanya untuk menangani sistem ulangan yang terjadwal (UTS dan UAS), tidak menangani ulangan remidial. Bentuk soal ulangan yang digunakan adalah *multiple choices single answer*.
3. Sistem yang dibangun tidak mendukung tampilan pada perangkat bergerak.

1.4 Tujuan Penelitian

1. Merancang dan membangun sistem informasi ujian online berbasis web pada sekolah SMA Negeri 1 Gombong.
2. Membangun sistem yang mampu mengelola informasi khususnya pelaksanaan ujian secara online.
3. Mengetahui tentang proses pelaksanaan dan penilaian ujian di SMA Negeri 1 Gombong.

1.5 Manfaat Penelitian

Manfaat Penelitian pada Sistem Informasi Ujian Online Berbasis Web (Studi Kasus: SMA Negeri 1 Gombong) telah dijelaskan sebagai berikut

1. Memudahkan dan memanfaatkan sumber daya manusia secara efektif dan efisien.

2. Pemanfaatan waktu secara cepat dan akurat.
3. Membantu guru dalam mengoreksi hasil ujian dengan penyajian hasil ujian secara langsung setelah siswa melaksanakan ujian.

1.6 Metodologi Penelitian

Metodologi penelitian ini dibagi menjadi empat bagian utama yaitu observasi, wawancara, studi literatur, dan metode pengembangan dan perancangan sistem.

1. Wawancara

   Wawancara dilakukan dengan narasumber guru dan kepala sekolah di lingkungan sekolah SMA Negeri 1 Gombong. Wawancara ini dilakukan dengan tujuan untuk mendapatkan informasi mengenai kebutuhan fungsional maupun nonfungsional dalam Sistem Informasi Ujian Online Berbasis Web.
2. Observasi

   Observasi dilakukan langsung dengan mengamati langsung bagaimana proses pelaksanaan ujian atau ulangan siswa melalui media-media terkait yang berhubungan dengan pembuatan soal, koreksi jawaban dan pelaksanaan ujian. Dalam hal ini media yang digunakan berupa observasi terhadap moodle, aplikasi yang sebelumnya pernah digunakan untuk ujian online siswa.
3. Studi Literatur

   Studi literatur dilakukan dengan mengumpulkan data dan mempelajari informasi yang berkaitan dengan Sistem Informasi Ujian Online di program studi Komputer dan Sistem Informasi, termasuk dalam perancangan, analisa data, dan implementasi sistem.
4. Metode Perancangan dan Pengembangan Sistem

a. Analisis Sistem

   Analisis Sistem berisi tentang permasalahan yang ada dalam proses pelaksanaan ujian secara manual pada SMA Negeri 1 Gombong. Hasil dari analisis permasalahan ini kemudian dituliskan dalam analisis kebutuhan

sistem, kebutuhan data, kebutuhan fungsional dan juga kebutuhan non fungsional sistem yang akan dibangun.

b. Perancangan Sistem

Perancangan sistem berisi tentang perancangan proses yang terdiri dari perancangan *usecase diagram*, *activity diagram*, dan *class diagram* berdasarkan analisis yang telah dilakukan. Kemudian perancangan selanjutnya adalah perancangan basis data, perancangan menu, dan perancangan user interface.

c. Implementasi Sistem

Implementasi Sistem berisi tentang sistem yang akan diimplementasikan dari perancangan yang telah dilakukan, Sistem Informasi Ujian Online Berbasis Web menggunakan bahasa pemograman Hypertext Prepocessor (PHP), framework CodeIgniter, basis data MySQL dan server Apache.

d. Pengujian Sistem

Pengujian Sistem berisi tentang pembahasan dalam pengujian Sistem Informasi Ujian Online yang telah dibuat. Tujuannya adalah menguji kesesuaian sistem yang dibuat dengan perancangan yang disusun sebelumnya, mengurangi kesalahpahaman pada sistem serta memastikan sistem ini dapat digunakan dengan baik.

e. Dokumentasi dan Pelaporan

Dokumentasi dan Pelaporan berisi tentang dokumentasi dan pelaporan yang merupakan bahasan tentang pembuatan laporan dan dokumentasi dari Sistem Informasi Ujian Online Berbasis Web yang telah dibuat.

1.7 Sistematika Penulisan

Sistematika penulisan tugas akhir ini merupakan Gambaran secara menyeluruh mengenai masalah yang akan dibahas dalam penelitian, yaitu terdiri dari :

BAB I PENDAHULUAN

Membahas tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan

sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Membahas tentang tinjauan secara umum sebagai dasar penelitian. Pada bagian ini menjelaskan informasi-informasi yang digunakan sebagai bahan acuan penulis yang bersumber Informasi tersebut dapat berupa penelitian-penelitian yang sudah dilakukan sebelumnya.

BAB III LANDASAN TEORI

Menjelaskan mengenai teori-teori yang menjadi landasan dalam tahap analisis dan perancangan sistem. Selain itu, pada bab ini juga berisi penjelasan tentang komponen-komponen yang digunakan dalam membangun sistem.

BAB IV ANALISIS DAN PERANCANGAN SISTEM

Membahas tentang analisis dan perancangan sistem pada penelitian ini. Bagian analisis meliputi analisis permasalahan, kebutuhan fungsional dan nonfungsional sistem, Gambaran umum sistem, analisis kebutuhan data, analisis basis data, dan analisis model sistem. Perancangan sistem meliputi perancangan struktur menu sistem dan perancangan antarmuka pengguna yang digunakan dalam penelitian ini.

BAB V IMPLEMENTASI SISTEM

Membahas tentang implementasi sistem beserta hal-hal yang mendukung dalam pembangunan sistem. Pada bab ini menjelaskan beberapa kode program yang digunakan dalam membangun sistem dan menampilkan tampilan antarmuka yang digunakan dalam sistem.

BAB VI PENGUJIAN DAN PEMBAHASAN SISTEM

Membahas tentang pengujian dan pembahasan sistem yang menguji kesesuaian sistem dengan perancangan yang sudah dibuat. Pengujian sistem bertujuan untuk mengevaluasi kinerja sistem, mengurangi adanya kesalahan serta memastikan sistem dapat digunakan dengan baik.

BAB VII KESIMPULAN DAN SARAN

Memuat tentang kesimpulan dari laporan proyek dan saran terhadap seluruh kegiatan proyek yang telah dilakukan. Saran diperlukan sebagai upaya pengembangan sistem selanjutnya.

DAFTAR PUSTAKA

Memuat tentang sumber referensi literatur yang digunakan dalam penyusunan laporan proyek ini.

BAB II
TINJAUAN PUSTAKA

Dalam pembuatan tugas akhir ini, dilakukan tinjauan pada beberapa sumber pustaka dan berikut adalah beberapa sumber pustaka yang berkaitan dengan Sistem Informasi Ujian Online yang digunakan sebagai acuan oleh penulis:

Darmawan (2000) melakukan penelitian yang berjudul "Aplikasi PHP Pada Website Online Examination", dengan tujuan sebagai fasilitas dan sarana bagi pengguna untuk mengasah kemampuannya dalam suatu bidang tertentu sehingga dalam pelaksanaan ujian tersebut tidak memerlukan bantuan dari orang lain untuk mengerjakannya. Situs ujian online ini diharapkan mampu melakukan otomatisasi sistem ujian baik dalam menampilkan soal-soal ujian maupun dalam penilaian sehingga mempermudah pengguna untuk menggunakannya serta bagi administrator untuk mengelola soal-soal yang akan ditampilkan.

Kusworo (2010) melakukan penelitian yang berjudul "Pembuatan Sistem Ujian Online", sistem ujian yang masih dilakukan secara manual termasuk dengan sistem koreksi ujiannya, membuat dosen biasanya harus menambah jam kerja untuk membuat dan menilai soal ujian para mahasiswa secara manual. Dalam kemajuan teknologi informasi yang semakin pesat, ujian manual tidak lagi dapat dijadikan sebagai kegiatan rutin. Untuk itu dibutuhkan sebuah sistem ujian online sebagai solusi untuk mengoptimalkan sistem ujian manual.

Sistem ujian online bermanfaat untuk mengurangi tingkat kecurangan mahasiswa karena soal yang disajikan akan berbeda satu dengan yang lain serta dapat menghemat waktu yang biasanya digunakan untuk mengoreksi ujian sesuai dengan kriteria sistem itu sendiri. Sistem ujian online juga memilki manfaat finansial yaitu dapat menghemat kertas ujian karena menggunakan piranti yang tidak sekali pakai. Penerapan sistem ujian online diharapkan akan mampu meningkatkan mutu pendidikan dan pengajaran di masa mendatang.

Ihsan (2012) mengembangkan "Sistem Informasi Ujian Online dan Penilaian Siswa Berbasis Web Pada SMK Pasundan 4 Bandung" dengan tujuan

untuk memperbaiki sistem informasi yang ada di SMK Pasundan 4 Bandung dengan mengganti sistem ujian yang manual dengan yang terkomputerisasi. Sistem ini telah dirancang dengan bahasa pemrograman PHP, Javascript, dan database MySQL.

Untuk melihat perbandingan pada masing-masing penelitian yang telah dilakukan oleh penilti diatas, dapat dilihat pada Tabel 2.1.

Tabel 2.1 Perbandingan Tinjauan Pustaka

Penulis	Judul	Teknologi	Deskripsi
Darmawan(2000)	Aplikasi PHP Pada Website Online Examination	PHP, MySQL	Otomatisasi sistem dalam mengelola soal-soal dan penilaian ujian
Kusworo (2010)	Pembuatan Sistem Ujian Online	PHP, MySQL	Mengelola data soal ujian, proses evaluasi hasil ujian, serta menangani ujian yang bersifat teoritis dan soal yang disajikan berupa soal dengan tipe *multiple choice.*
Ihsan(2012)	Sistem Informasi Ujian Online dan Penilaian Siswa Berbasis Web Pada SMK Pasundan 4 Bandung	PHP, Javascript, dan database MySQL.	Mengolah soal-soal ujian siswa, penjadwalan, dan penilaian ujian.
Diana Ari Rahmawati (2017)	Sistem Informasi Ujian Online Berbasis Web (Strudi Kasus: SMA Negeri 1 Gombong	HTML, PHP, *Javascript, framework* CodeIgniter, Bootstrap, DBMS MySQL, XAMPP	Mengelola soal ujian dan penilaian hasil ujian siswa. Sehingga siswa dapat melihat nilai hasil ujian secara langsung dan pengelola dapat mencetak hasil ujian.

BAB III
LANDASAN TEORI

3.1.Konsep Dasar Sistem

Suatu sistem pada dasarnya adalah sekelompok unsur yang berhubungan satu sama lain untuk mencapai suatu tujuan tertentu. Secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu. Menurut Warren D. Stallings, Jr dkk. (Jogiyanto, 2005) Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu.

Sistem terdiri dari sekelompok elemen yang terdiri dari orang, prosedur, perangkat keras, perangkat lunak, basis data, jaringan komputer dan komunikasi data. Semua elemen ini merupakan komponen fisik dari sebuah sistem.

3.1.1. Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu mempunyai komponen-komponen *(components)*, batas sistem *(boundary)*, lingkungan luar sistem *(environments)*, penghubung *(interface)*, masukan *(input)*, keluaran *(output)*, pengolah *(process)* dan sasaran *(objective)* atau tujuan *(goal)* (Jogiyanto, 2005).

1. Memiliki komponen

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerjasama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem tidak peduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem.

2. Batas sistem *(Boundary)*

   Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini

memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batas suatu sistem menunjukkan ruang lingkup *(scope)* dari sistem tersebut.

3. Lingkungan luar sistem *(enviroment)*

   Lingkungan luar sistem adalah segala sesuatu di luar sistem yang dapat mempengaruhi sistem tersebut. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan agar tidak mengganggu kelangsungan hidup dari sistem.

4. Penghubung sistem *(Interface)*

   Penghubung sistem merupakan media perantara antar subsistem. Melalui penghubung ini memungkinkan sumber daya mengalir dari satu subsistem ke subsistem lainnya. *Output* dari satu subsistem akan menjadi input untuk subsistem yang lainnya. Dengan penghubung tersebut, satu subsistem dapat berinteraksi dengan subsistem yang lainnya membentuk satu kesatuan.

5. Masukan sistem *(Input)*

   Masukan sistem adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa *maintenance input* dan sinyal *input. Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Sinyal *input* adalah energi yang diproses untuk mendapatkan keluaran.

6. Keluaran sistem *(Output)*

   Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran merupakan masukan untuk subsistem yang lain.

7. Pengelolaan sistem *(Process)*

   Suatu sistem memiliki bagian pengolahan sistem atau sistem tersebut memiliki kemampuan untuk mengolah sistemnya sendiri. Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan-bahan yang lain menjadi keluaran berupa barang jadi.

8. Sasaran sistem

   Suatu sistem mempunyai tujuan atau sasaran. Jika suatu sistem tidak

mempunyai sasaran, maka operasi sistem tidak memiliki kegunaan. Sasaran dari sistem sangat menentukan masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

3.1.2. Klasifikasi Sistem

Menurut Jogiyanto (2005), Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya adalah sebagai berikut. Sistem diklasifikasikan sebagai sistem abstrak *(abstrack sistem)* dan sistem fisik *(physical sistem).* Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak nampak secara fisik. Sistem fisik adalah sistem yang ada secara fisik. Sistem diklasifikasikan sebagai sistem alamiah *(natural sistem)* dan sistem buatan manusia *(human made sistem).*

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia. Sistem buatan manusia adalah sistem yang dirancang oleh manusia. Sistem buatan manusia melibatkan interaksi antara manusia dengan mesin disebut dengan *human-machine sistem*, karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

Sistem diklasifikasikan sebagai sistem tertentu *(deterministic sistem)* dan sistem tak tentu *(probabilistic sistem).* Sistem tertentu merupakan sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi, interaksi diantara bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diprediksi. Sistem tak tentu merupakan sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas. Sistem diklasifikasikan sebagai sistem tertutup *(closed sistem)* dan sistem terbuka *(open sistem).* Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luar. Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan atau subsistem yang lainnya.

3.2.Konsep Dasar Informasi

Secara umum informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam bentuk yang lebih berguna dan lebih berarti bagi penerimanya. Informasi menggambarkan suatu kejadian yang nyata yang digunakan untuk pengambilan keputusan. (Sutabri, 2004) Informasi adalah data yang telah diklasifikasikan atau diolah yang digunakan dalam proses pengambilankeputusan. Informasi memiliki kegunaan untuk mengurangi ketidakpastian dalam proses pengambilan keputusan tentang suatu keadaan. Dalam prosesnya, suatu informasi tidak hanya digunakan oleh satu pihak saja melainkan oleh beberapa pihak.

3.3.2. Siklus Informasi

Data merupakan bentuk yang masih mentah sehingga perlu diolah lebih lanjut. Data diolah untuk menghasilkan informasi menggunakan suatu model proses tertentu. Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan suatu tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai masukan, untuk kemudian diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus tertentu. Siklus ini disebut sebagai siklus informasi *(information cycle)* dan siklus pengolahan data *(data processing cycle).* (John & Grudnitski, 1986).

3.3.2. Kualitas Informasi

Kualitas dari suatu informasi *(quality of information)* tergantung dari hal, yaitu informasi harus akurat *(accurate),* tepat pada waktunya *(timeliness*) dan relevan *(relevance).* Kualitas dari sebuah informasi dapat digambarkan dengan bentuk bangunan yang ditunjang oleh tiga buah pilar (John & Grudnitski, 1986).

1. Akurat

   Akurat dapat diartikan informasi harus bebas dari kesalahan-kesalahan dan tidak bisa atau menyesatkan. Akurat juga berarti informasi yang disampaikan harus jelas mencerminkan maksud yang ingin disampaikan. Informasi harus akurat karena dari sumber informasi sampai ke penerima kemungkinan informasi mengalami gangguan *(noise),* yang mengakibatkan

informasi menjadi berubah atau informasi tersebut menjadi rusak.

2. Tepat Pada Waktunya

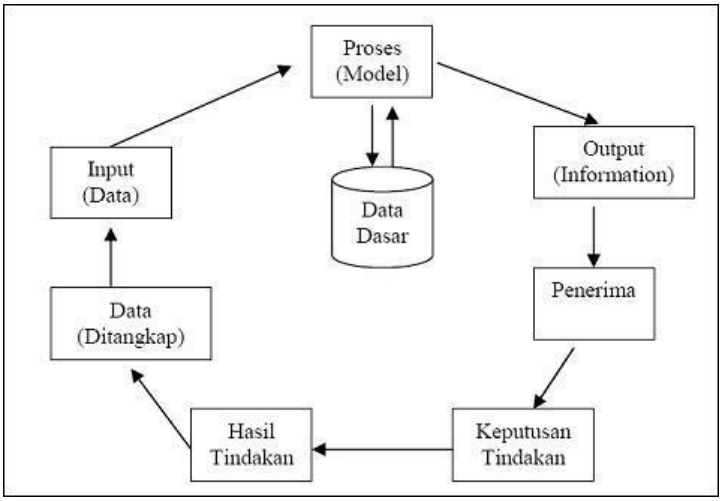
   Tepat pada waktunya berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang akan mengakibatkan informasi menjadi tidak bernilai lagi. Karena informasi merupakan landasan di dalam pengambilan keputusan. Keterlambatan dalam pengambilan keputusan akan berdampak fatal untuk organisasi.

3. Relevan

   Relevan berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi suatu informasi bagi masing-masing individu tentunya berbeda oleh karean itu relevansi bersifat relatif *(relative).* Sebagai contoh informasi mengenai kerusakan mesin produksi kepada seorang akuntan perusahaan adalah kurang relevan dan akan menjadi lebih relevan apabila diitujukan kepada ahli teknik perusahaan.

3.3.2. Nilai Informasi

Nilai dari informasi *(value of information)* ditentukan oleh dua hal, yaitu manfaat dan biaya untuk mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Akan tetapi perlu diperhatikan bahwa informasi yang digunakan pada suatu sistem informasi umumnya digunakan untuk beberapa kegunaan. Sehingga tidak memungkinkan dan sulit untuk menghubungkan suatu bagian informasi pada suatu masalah yang tertentu dengan biaya untuk memperolehnya, karena sebagian besar informasi dinikmati tidak hanya oleh satu pihak didalam suatu perusahaan saja. Informasi dapat dijelaskan dalam sebuah siklus, yaitu siklus informasi seperti terlihat pada Gambar 3.1



Gambar 3.1 Siklus sistem informasi (Jogiyanto, 2005)

Data merupakan bentuk mentah yang perlu diolah lebih lanjut. Data ditangkap sebagai input, diproses melalui suatu model membentuk informasi. Pemakai kemudian menerima informasi sebagai landasan untuk membuat sebuah keputusan dan melakukan tindak operasional yang akan membuat sejumlah data baru. Data baru tersebut selanjutnya menjadi *input* pada proses berikutnya, begitu seterusnya sehingga membentuk sebuah siklus informasi (Sutabri,2004)

3.3. Konsep dasar Sistem Informasi

Sistem Informasi merupakan sistem yang mempunyai kemampuan untuk mengumpulkan informasi dari semua sumber dan menggunakan berbagai media untuk menampilkan informasi (Mcleod, 2001). Menurut Alter (1992), sistem informasi adalah kombinasi antara prosedur kerja, informasi, orang dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi. Berdasarkan dua pengertian tersebut, dapat disimpulkan bahwa sistem informasi adalah komponen atau kelompok elemen yang mempunyai kemampuan untuk mengumpulkan informasi dari berbagai sumber untuk mencapai suatu tujuan tertentu. Menurut Turban dkk. (2006) sistem informasi terdiri dari beberapa komponen. Diantaranya adalah sebagai berikut :

1. *Hardware* (Perangkat Keras)

   Perangkat keras bagi suatu sistem informasi terdiri atas komputer (pusat pengolah, unit masukan/keluaran), peralatan penyimpanan data, dan

terminal masukan/keluaran. Contoh perangkat keras adalah Monitor, CPU, *Keyboard*, *Mouse*, dll.

2. *Software* (Perangkat Lunak)

   Perangkat lunak dapat dibagi dalam 3 jenis utama :

a. Sistem perangkat lunak umum, seperti sistem pengoperasian dan sistem manajemen data yang memungkinkan pengoperasian sistem komputer.
b. Aplikasi perangkat lunak umum, seperti model analisis dan keputusan.
c. Aplikasi perangkat lunak yang terdiri atas program yang secara spesifik dibuat untuk setiap aplikasi.

3. Basis Data

   Basis data adalah sekumpulan file, tabel relasi dan lain-lain yang saling berkaitan dan menyimpan data serta berbagai hubungan diantaranya.

4. Jaringan

   Jaringan adalah informasi dan data bergerak melalui kabel-kabel atau tanpa kabel sehingga memungkinkan pengguna jaringan komputer dapat saling bertukar dokumen dan data.

5. Prosedur

   Prosedur adalah serangkaian instruksi mengenai bagaimana menggabungkan beberapa komponen diatas agar dapat memproses informasi dan menciptakan sistem yang diinginkan.

6. Orang

   Orang adalah individu yang bekerja, berinteraksi atau menggunakan sistem informasi secara langsung.

3.3.1. Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan atau *building block* (Jogiyanto, 2005). Terdiri dari komponen *input*, komponen model, komponen *output*, komponen teknologi, komponen *hardware*, komponen *software*, komponen basis data, dan komponen kontrol. Semua komponen tersebut saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran.

1. Komponen *Input*

   Input mewakili data yang masuk kedalam sistem informasi. *Input* disini

termasuk metode dan media untuk menangkap data yang dimasukkan, yang dapat berupa dokumen dasar.

2. Komponen Model

   Komponen ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Komponen *Output*

   Hasil dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua pemakai sistem.

4. Komponen Teknologi

   Teknologi merupakan *"tool*dari*box"*sebuahsistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan, dan mengakses data, menghasilkan, dan mengirimkan keluaran, dan membantu pengendalian dari sistem secara keseluruhan.

5. Komponen *Hardware*

   Hardware atau perangkat keras sebagai suatu media penting bagi sistem informasi. Berfungsi sebagai tempat untuk menampung database atau lebih mudah dikatakan sebagai sumber data dan informasi untuk memperlancar dan mempermudah kerja sistem informasi.

6. Komponen *Software*

   Software atau perangkat lunak digunakan sebagai tempat mengolah, menampilkan dan memanipulasi data yang diperoleh dari *hardware* untuk menciptakan sebuah informasi.

7. Komponen Basis Data

   Kumpulan data yang tersimpan dan saling berhubungan satu sama lain yang diolah perangkat lunak serta disimpan oleh perangkat keras.

8. Komponen Kontrol

   Banyak hal yang akan merusak sebuah sistem informasi seperti bencana alam, kegagalan kegagalan sistem itu sendiri, ketidakefisienan, dan lain lain. Beberapa pengendalian perlu

dirancang agar meyakinkan bahwa hal hal yang akan merusak sistem dapat dicegah.

3.3.2. Klasifikasi Sistem Infromasi

Sistem informasi dalam organisasi mencatat/merekam file yang permanen dan mengelola data untuk menghasilkan informasi berguna yang mendukung sebuah organisasi. Sistem informasi datang dalam sebuah bentuk dan ukuran, yang diklasifikasikan berdasarkan fungsi yang dimiliki. Bentuk klasifikasi sistem informasi berupa (Supriyanto, 2005) :

3. Sistem pemrosesan transaksi, yaitu yaitu memproses transaksi bisnis seperti pemesanan, pembayaran, reservasi, dan sebagainya.
4. Sistem informasi manajemen, yaitu menghasilkan informasi untuk kebutuhan manajer.
5. Sistem pendukung keputusan, yaitu digunakan untuk membantu pengambilan keputusan atau mengidentifikasi atau memilih antara pilihan dan keputusan.
6. Sistem informasi eksekutif, yaitu untuk kebutuhan eksekutif dalam merencanakan bisnis dan menilai performa terhadap rancangan tersebut.
7. Sistem pakar, meng-*capture* dan menghasilkan kembali pengetahuan pemecahan masalah ahli atau pengambilan keputusan kemudian menyimulasikan “pemikiran” ahli tersebut.
8. Sistem komunikasi dan kolaborasi, yaitu untuk meningkatkan komunikasi dan kolaborasi antara orang-orang dalam dan luar organisasi.
9. Sistem otomatisasi kantor, yaitu membantu karyawan untuk membuat dan berbagi dokumen yang mendukung aktivitas kantor sehari-hari.

3.4. Siklus Hidup Sistem

Siklus hidup sistem (*system life cycle*) adalah proses evolusioner yang diikuti dalam menerapkan sistem atau subsistem informasi berbasis computer. Siklus hidup sistem terdiri dari serangkaian tugas yang erat mengikuti langkah-langkah pendekatan sistem karena tugas-tugastersebut mengikuti pola yang teratur dan dilakukan secara *top down.* Sikulus hidup sistem sering disebut sebagai pendekatan air terjun (*waterfall approach*) bagi pembangunan dan pengembangan sistem (Tata Sutabri, 2003).

Pembangunan sistem hanyalah salah satu dari rangkaian daur hidup suatu sistem. Meskipun demikian, proses ini merupakan aspek yang sangat penting. Kita akan lihat beberapa fase/tahapan dari daur hidup suatu sistem.

Mengenali adanya kebutuhan

Sebelum segala sesuatu terjadi, timbul suatu kebutuhan atau problema yang harus dapat dikenali sebagaimana adanya. Kebutuhan dapat terjadi sebagai hasil perkembangan dari organisasi dan volume yang meningkat melebihi kapasitas dari sistem yang ada. Semua kebutuhan ini harus dapat didefinisikan dengan jelas. Tanpa adanya kejelasan dan kebutuhan yang ada, pembangunan sistem akan kehilangan arah dan efektifitasnya.

Pembangunan Sistem

Suatu proses atau seperangkat prosedur yang harus diikuti untuk menganalisis kebutuhan yang timbul dan membangun suatu sistem untuk dapat memenuhi kebutuhan tersebut.

Pemasangan Sistem

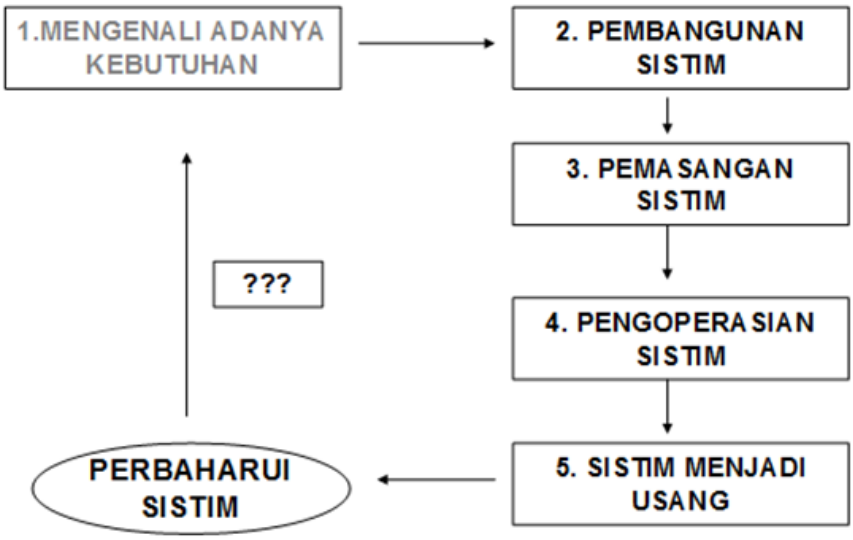
Setelah tahap pembangunan sistem selesai. Sistem kemudian akan dioperasikan. Pemasangan sistem merupakan tahap yang penting pula dalam daur hidup sistem. Peralihan dari tahap pembangunan menuju tahap operasional terjadi pemasangan sistem yang sebenarnya, yang merupakan langkah akhir dari suatu pembangunan sistem.

Pengoperasian Sistem

Program-program komputer dan prosedur-prosedur pengoperasian yang membentuk suatu sistem informasi semuanya bersifat statis, sedangkan organisasi ditunjang oleh sistem informasi tadi. Ia selalu mengalami perubahan-perubahan itu karena pertumbuhan kegiatan bisnis, perubahan peraturan, dan kebijaksanaan ataupun kemajuan teknologi. Untuk mengatasi perubahan-perubahan tersebut, sistem harus diperbaiki atau diperbaharui.

Sistem Menjadi Usang

Kadang perubahan yang terjadi begitu drastis sehingga tidak dapat diatasi hanya dengan melakukan perbaikan-perbaikan pada sistem yang berjalan. Tibalah saatnya secara ekonomis dan teknis sistem yang ada sudah tidak layak lagi untuk dioperasikan dan sistem yang baru perlu dibangun untuk menggantikannya. Sistem informasi kemudian akan melanjutkan daur hidupnya. Sistem dibangun untuk memenuhi kebutuhan yang muncul. Sistem berdaptasi terhadap perubahan-perubahan lingkungan yang dinamis. Sampailah pada kondisi dimana sistem tersebut tidak dapat lagi beradaptasi dengan perubahan-perubahan yang ada ataupun secara ekonomis tidak layak lagi untuk dioperasikan. Sistem yang baru kemudian dibangun untuk menggantikannya. Urutan dari daur hidup sistem dapat dilihat pada Gambar 3.2.



Gambar 3.2 Daur Hidup Sistem

3.5. Basis Data

Basis data merupakan sekumpulan dari data yang saling terkait satu sama lain secara logis dan menggambarkan hubungan antara suatu tabel dengan tabel lainnya yang dirancang untuk memenuhi kebutuhan informasi dari suatu organisasi. (Connolly & Carolyn, 2002).

Beberapa alasan penggunaan basis data ke dalam tiga hal. (Date, 2000). Padat, artinya tidak perlu membuat arsip kertas dalam ukuran yang besar.

Kemudian basis data memiliki kecepatan, artinya sistem dapat mendapatkan kembali dan mengubah data jauh lebih cepat dari pada yang dapat dilakukan oleh manusia. Aktual, artinya informasi yang terbaru selalu akurat dan tersedia setiap waktu pada saat yang dibutuhkan.

Terdapat delapan keuntungan menggunakan basis data, yaitu mengurangi redudansi, konsisten, data dapat dibagikan, sesuai dengan bentuk standar, membatasi keamanan, integritas terjaga, keperluan yang bertentangan dapat diseimbangkan dan tersedianya dukungan untuk transaksi. (Date, 2000)

3.5.1. Konsep Basis Data

Menurut Sutanta (2004), Basis Data adalah kumpulan dari data-data yang saling terhubung dan tersimpan secara bersama-sama pada suatu media. Disimpan dengan cara-cara tertentu sehingga mudah untuk digunakan atau ditampilkan kembali. James Martin dalam bukunya mengatakan ada 2 kelompok yang menjadi tujuan dari dibentuknya sebuah basis data yaitu tujuan primer dan tujuan sekunder (Sutanta, 2004). Tujuan primer adalah tujuan utama yang ingin dicapai dalam perancangan dan pengembangan basis data sedangkan tujuan sekunder adalah tujuan tambahan yang dimaksud untuk mencapai tujuan primer.

3.5.2. Perancangan Basis Data

Dalam proses perancangan basis data, awalnya dibuat berdasarkan Diagram Alir Data atau DAD. DAD adalah salah satu alat yang paling mudah dan cepat dalam menggambarkan sistem. Terdapat beberapa macam diagram yang digunakan dalam merancang sebuah sistem yang sedang dianalisa atau sistem yang sedang dibuat diantaranya Diagram Konteks dan Diagram Alir Data.

Diagram Konteks digunakan untuk menjelaskan prosedur secara garis besar dan mendetail berdasarkan entitas dan atau prosesnya. Diagram Konteks juga memberikan pandangan sistem yang bersifat perangkat lunak secara keseluruhan baik *input* maupun *output* yang dihasilkan oleh sistem. Sebuah DAD adalah representasi grafis dari aliran data melalui sistem informasi dan dapat juga digunakan untuk visualisasi dari pengolahan data. Sebuah DAD tidak memberikan informasi operasi secara berurutan dan paralel, oleh karena itu sangat berbeda dari suatu *flowchart.* Terdapat komponen komponen pembentuk Diagram

Alir Data ada 4 yaitu :

1. *Terminator*

   Digambarkan dengan bentuk kotak yang mewakili entitas eksternal yang berkomunikasi dengan sistem yang sedang dikembangkan.

2. Proses

   Digambarkan dengan bentuk lingkaran yang menggambarkan bagian dari sistem yang mentransformasikan *input* menjadi *output* dan menjelaskan proses/kegiatan yang sedang dilaksanakan.

3. Data Store

   Digambarkan dengan 2 bentuk garis sejajar, biasanya berkaitan dengan penyimpanan-penyimpanan, seperti file atau *database* yang berkaitan dengan penyimpanan secara komputerisasi.

4. Alur Data

   Digambarkan dengan tanda anak panah yang menunjukkan arah menuju ke suatu proses dan arah keluar dari suatu proses. Fungsinya untuk menerangkan perpindahan data.

3.5.3. Model Basis Data

Menurut Kadir (1999), model basis data menyatakan hubungan antar rekaman yang tersimpan dalam basis data. Berikut ini adalah macam-macam model basis data:

1. Model Hirarkis

   Model hirarkis biasa disebut model pohon. Model ini menggunakan pola hubungan orang tua–anak *(parent-child).* Simpul yang terhubung ke simpul pada level dibawahnya disebut orang tua *(parent).* Setiap orang tua bisa memiliki lebih dari satu anak, tetapi anak hanya memiliki satu orang tua. Simpul yang dibawah oleh simpul orang tua disebut anak *(child).* Simpul orang tua yang tidak memiliki orang tua disebut akar *(root).* Simpul yang tidak memiliki anak disebut daun. Adapun hubungan antara orang tua dan anak disebut cabang.

2. Model Jaringan *(Network Model)*

   Model ini menyerupai model hirarkis, dengan perbedaan suatu simpul anak

bisa memiliki lebih dari satu orang tua.

3. Model Keterhubungan Entitas *(Entity-Relationship Model)*

   Model ini berisi komponen-komponen himpunan entitas dan relasi yang masing-masing dilengkapi dengan atribut-atribut, dan dapat digambarkan dengan menggunakan Diagram *Entity-Relationship* (Diagram E-R). Perancangan basis data seringkali diasosiasikan pembuatan model *Entity-Relationship* (model E-R), dimana kelompok-kelompok data dan relasi antar kelompok tersebut diwujudkan dalam bentuk diagram. Diagram E-R merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan atau relasi antara objek-objek tersebut. Diagram E-R digunakan untuk memodelkan struktur data dan hubungan antara data. Diagram E-R menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data. Terdapat beberapa simbol yang digunakan dalam pembuatan sebuah Diagram E-R ditunjukan pada Tabel 3.1 dan Tabel 3.2.

Tabel 3.1 Simbol Diagram E-R (Kadir, 1999)

Komponen	Penjelasan	Simbol
Himpunan Entitas	Objek yang dapat dibedakan dengan yang lain dapat berupa objek secara fisik seperti : orang, rumah , kendaraan atau objek secara konsep seperti : pekerjaan, perusahaan, dan lain lain.	
Atribut	Karakteristik dari entitiy atau relationship, yang menyediakan penjelasan detail tentang entity atau relationship tersebut.	
Atribut (Kunci Primer)	Atribut yang membedakan antara objek yang satu dengan objekyang lainnya.	Kunci Primer

Tabel 3.2 Simbol Diagram E-R (Kadir, 1999) (lanjutan)

Komponen	Penjelasan	Simbol
Garis Konektor	Garis yang menghubungkan antara entitas dengan atribut atau entitas dengan relasi.	
Relasi	Hubungan yang terjadi antara satu entitas dengan entitas lainnya.	

4. Model Relasional

   Model ini menggunakan sekumpulan tabel berdimensi dua (yang disebut relasi atau tabel). Berikut model data relasional menurut (Kadir, 1999) :

a. Struktur data, data-data diorganisasikan dalam bentuk tabel dengan barisbaris dan kolom-kolom.

b. Manipulasi data, operasi yang sangat berdaya-guna (menggunakan *Structured Query Languange*) digunakan untuk memanipulasi data-data yang disimpan direlasi-relasi. Integritas data, fasilitas-fasilitas untuk menspesifikasi aturan bisnis yang memelihara integritas data saat mereka dimanipulasi.

3.5.4. Relasi

Relasi adalah tabel data dua dimensi, dengan ketentuan-ketentuan sebagai berikut (Kadir,1999) :

1. Setiap baris data *(record)* memiliki beberapa atribut *(fields)*
2. Setiap tipe *record* membentuk tabel dan relasi
3. Derajat relasi suatu tabel dinyatakan dengan jumlah atribut yang terdapat di dalam tabel yang bersangkutan. Suatu tabel yang hanya memiliki satu atribut disebut memiliki relasi *binary* dan suatu tabel yang memiliki dua atribut disebut tabel dengan relasi *binary,* sedangkan tabel dengan sejumlah n-atribut disebut tabel relasin-nya.

3.5.5. Derajat Relasi (Kardinalitas)

Kardinalitas merupakan spesifikasi dari sejumlah peristiwa dari satu objek

yang dapat dihubungkan ke sejumlah peristiwa dari objek yang lain. Kardinalitas menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Entitas merupakan individu yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain (Pressman, 2001). Menurut Pressman (2001), kardinalitas dapat terjadi antara dua himpunan entitas berupa :

1. Satu ke satu *(one to one)*, terjadi apabila entitas himpunan pertama berelasi hanya dengan satu entitas di himpunan kedua dan sebaliknya.
2. Satu ke banyak *(one to many)*, terjadi apabila entitas himpunan pertama berelasi dengan lebih dari satu entitas di himpunan kedua, tetapi entitas di himpunan kedua berelasi hanya satu entitas di himpunan pertama.
3. Banyak ke satu *(many to one)*, terjadi apabila entitas himpunan pertama berelasi hanya dengan satu entitas di himpunan kedua, tetapi entitas di himpunan kedua berelasi dengan lebih dari satu entitas di himpunan pertama.
4. Banyak ke banyak *(many to many)*, terjadi apabila entitas himpunan pertama berelasi dengan lebih dari satu entitas di himpunan kedua dan sebaliknya.

3.5.6. Kunci

Ada beberapa kunci yang dapat diterapkan pada tabel dalam model basis data relasional (Kadir, 1999), yaitu :

1. *Superkey*, satu atau lebih atribut yang dapat membedakan setiap baris data dalam sebuah tabel secara unik.
2. *Candidate-key*, kumpulan atribut minimal yang dapat membedakan setiap baris data sebuah tabel secara unik.
3. *Primary-key*, kunci yang diambil dari beberapa *candidate-key* yang unik.
4. *Foreign-key*, kunci tamu atau kunci asing dari suatu tabel dimana kunci ini juga terdapat pada tabel lain sebagai *primary-key*.

3.5.7. Query

Query adalah bahasa yang dipergunakan untuk melakukan proses permintaan yang diberikan oleh user atau pengguna untuk mengambil informasi yang terdapat pada *database* dengan memberikan suatu kriteria tertentu untuk penyaringan data. Secara umum *query* terbagi menjadi 3 komponen, yaitu :

1. *Data Definion Language* (DDL)

   DDL digunakan untuk mendefinisikan, mengubah, serta menghapus basis data dan objek-objek yang diperlukan dalam basis data. DDL yang sering digunakan adalah *create* yang digunakan untuk membuat objek baru, *alter* yang digunakan untuk mengubah objek yang sudah ada, *use* digunakan untuk menggunakan objek tersebut, dan *drop* yang digunakan untuk menghapus objek tersebut. DDL digunakan untuk pembuatan sebuah aplikasi basis data.

2. *Data Manipulation Language* (DML)

   DML digunakan untuk memanipulasi data yang sudah ada dalam tabel.DDL yang digunakan secara umum yaitu :

a. *Select* : Untuk menampilkan data yang sudah ada.
b. *Insert* : Untuk memasukkan data yang baru.
c. *Update* : Untuk mengubah data yang sudah ada.
d. *Delete* : Untuk menghapus data yang sudah ada.

3. *Data Control Language* (DCL)

   DCL digunakan untuk pemberian otoritas, konfirmasi penyimpanan *database*, merubah perubahan dan menghapus otoritas yang telah diberikan. Namun, tidak semua database dapat melakukan pengotrolan data seperti pembuatan otoritas.

3.6. UML (*Unified Modeling Language)*

Menurut Nugroho (2010:6), UML (*Unified Modeling Language*) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek. Pemodelan *(modeling)* sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

UML berorientasi objek menerapkan banyak level abstraksi, tidak bergantung proses pengembangan, tidak bergantung bahasa dan teknologi. Pemaduan beberapa notasi di beragam metodologi usaha bersama dari banyak pihak, di dukung oleh kakas- kakas yang di integrasikan lewat XML.

3.6.1. *Use Case* Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankanstem,dan bukanadalah “a “bagaimana”. Use case merepresentasikan.*Use case* menggambarkan pekerjaan tertentu, misalnya *login* ke sistem, membuat daftar belanja, dan sebagainya.

Aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case* diagram dapat sangat membantu ketika sedang menyusun analisis kebutuhan sebuah sistem, mengkomunikasikan rancangan dengan *client*, dan merancang *test case* untuk semua fitur yang ada pada sistem.

Use case dapat memasukkan *(include)* fungsionalitas use case lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang sejenis.

Use case juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar use case menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain. Terdapat beberapa simbol yang digunakan dalam pembuatan sebuah *Use case* diagram ditunjukan pada Tabel 3.3 dan Tabel 3.4.

Tabel 3.3 Simbol *use ase* diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
3		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
4		*Include*	Menspesifikasikan bahwa *use case* sumber secara *eksplisit*.
5		*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*System*	Menspesifikasikan ¡paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu ¡ hasil ¡ yang terukur bagi suatu aktor

Tabel 3.4 Simbol *use ase* diagram (lanjutan)

NO	GAMBAR	NAMA	KETERANGAN
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

3.6.2. *Activity* Diagram

Activity diagram menggambarkan berbagai alir kegiatan 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.

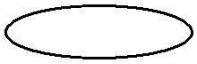
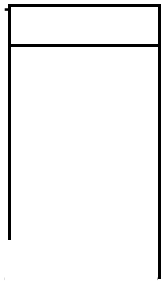
Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya *(internal processing).* Oleh karena itu *activity* diagram tidak menggambarkan *behavior internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur kegiatan dari level atas secara umum.

Sebuah kegiatan dapat direalisasikan oleh satu *use case* atau lebih. Kegiatan menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan kegiatan. Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan kegiatan. *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 objek *swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk kegiatan tertentu. Terdapat beberapa simbol yang digunakan dalam pembuatan sebuah *activity* diagram ditunjukan pada Tabel 3.5.

Tabel 3.5 Simbol *activity* diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*Actifity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Initial Node*	Bagaimana objek dibentuk atau diawali.
3		*Actifity Final Node*	Bagaimana objek dibentuk dan dihancurkan
4		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran. Syarat maju ke aktivitas berikutnya. Semua aktivitas harus sudah selesai
5		*Decision*	Suatu kondisi dimana ada kemungkinan perbedaan transisi. Salah satu kondisi masuk sudah cukup untuk masuk ke aktivitas berikutnya
6		*Swimlane*	Sebuah cara untuk mengelompokkan activity berdasarkan Aktor (mengelompokkan activity Swimlane dalam sebuah urutan yang sama)

3.6.3. *Class* Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class* diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. Terdapat beberapa simbol yang digunakan dalam pembuatan sebuah *class* diagram ditunjukan pada Tabel 3.6, Tabel 3.7, dan Tabel 3.8.

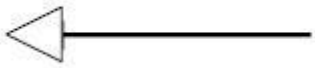
Tabel 3.6 Simbol *class* diagram

SIMBOL	NAMA	KETERANGAN
Site Config +sqlDNS:string +Adminemail:String	*Class*	*Class* adalah blok - blok pembangun pada pemrograman berorientasi objek. Sebuah *class* digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari *class*. Bagian tengah mendefinisikan *property*/atribut *class*. Bagian akhir mendefinisikan method method dari sebuah *class*.
1..n Owned by 1	*Assosiation*	Sebuah asosiasi merupakan sebuah *relationship* paling umum antara 2 *class*, dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 *class*. Garis ini bisa melambangkan tipe-tipe *relationship* dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah *relationship* (Contoh: *One-to-one*, *one-to-many*, *many-to-many*).

Tabel 3.7 Simbol *class* diagram (lanjutan)

SIMBOL	NAMA	KETERANGAN
	Composition	Jika sebuah *class* tidak bisa berdiri sendiri dan harus merupakan bagian dari *class* yang lain, maka *class* tersebut memiliki relasi *Composition* terhadap *class* tempat dia bergantung tersebut. Sebuah *relationship composition* digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/solid.
	Dependency	Kadangkala sebuah *class* menggunakan *class* yang lain. Hal ini disebut *dependency*. Umumnya penggunaan *dependency* digunakan untuk menunjukkan operasi pada suatu *class* yang menggunakan *class* yang lain. Sebuah *dependency* dilambangkan sebagai sebuah panah bertitik-titik.
	Aggregation	*Aggregation* mengindikasikan keseluruhan bagian *relationship* dan biasanya disebut sebagai relasi "mempunyai sebu dari". Sebuah*aggregation* digambarkan sebagai sebuah garis dengan sebuah jajaran genjang yang tidak berisi/tidak solid.

Tabel 3.8 Simbol *class* diagram (lanjutan)

SIMBOL	NAMA	KETERANGAN
	*Generalization*	Sebuah relasi *generalization* sepadan dengan sebuah relasi *inheritance* pada konsep berorientasiobyek. Sebuah *generalization* dilambangkan dengan sebuah panah dengan kepala panah yang tidak solid yang mengarah ke*parent*k-nya/induknya.

3.7. Internet

Internet *(inter-network)* adalah istilah untuk sekumpulan jaringan komputer yang mengembangkan berbagai situs. Internet menyediakan akses layanan telekomunikasi dan sumber daya informasi untuk seluruh pengguna di dunia. (Ramadhani, 2003). Layanan Internet meliputi komunikasi langsung melalui email, *chatting*, diskusi menggunakan *mailing list*, sumber daya informasi yang terdistribusi melalui situs *web*, *remote login*, kegiatan bertukar file dan dokumen menggunakan telnet maupun FTP *(file transfer protocol).*

Jaringan yang membentuk Internet bekerja berdasarkan suatu set protokol standar yang digunakan untuk menghubungkan jaringan komputer dan menyediakan alamat jaringan. (Ramadhani, 2003). Protokol digunakan untuk mengatur format data yang diizinkan melalui standar komunikasi dan lintas pesan. Protokol berjalan diberbagai jenis komputer dan *platform* serta bekerja dengan domain dan alamat Internet protokol yang biasa disebut dengan IP.

3.8. Web

Menurut (Ramadhani, 2003), *world wide web* atau yang lebih sering disebut dengan *web* merupakan aplikasi Internet yang paling popular. *Web* adalah sebuah sistem dimana terdapat informasi berupa teks, Gambar, suara atau multimedia yang tersimpan dalam *web server* dan diterjemahkan dalam bentuk *hypertext. Web* dapat diakses menggunakan *web browser*.

3.8.1. *Web Browser*

Menurut (Ramadhani, 2003), *web browser* adalah perangkat lunak dari sisi *client. Web browser* membaca halaman-halaman yang tersimpan dalam *web server* melalui protokol yang disebut dengan HTTP *(Hypertext Transfer Protocol).* Dengan memasukkan alamat *web* pada *web browser,* maka penggun dapat mendapatkan informasi dengan mudah. Saat ini ada beberapa jenis *web browser,* diantaranya adalah Microsoft Internet Explorer, Google Chrome, Mozilla Firefox, Opera dan Safari.

3.3.2. *Web Server*

Menurut (Pearson Higher Education, 1999), *web server* adalah alat yang menerima permintaan dari *client* kemudian merespon kembali dengan halaman *hypertext*. Tujuan utama *web server* adalah menyediakan dokumen *web* untuk *client. Web server* memungkinkan dokumen-dokumen *web* dihasilkan dengan cepat dan tidak perlu disimpan dalam disk pada computer *client. Web server* menggunakan protokol HTTP dan *web browser* sehingga *client* bisa mengaksesnya dengan mudah.

3.9. *Software Framework*

Software framework adalah sekumpulan kode program yang merupakan sebuah landasan dan bersifat universal yang digunakan oleh para *programmer* untuk membangun sebuah aplikasi. *Software framework* merupakan sebuah struktur konseptual dasar yang digunakan untuk memecahkan sebuah permasalahan atau isu-isu kompleks (Daqiqil, 2011). Keuntungan menggunakan *Software framework* (Daqiqil, 2011), diantaranya :

1. Menghemat waktu pengembangan karena *software framework* telah menyediakan *library* dan struktur sehingga hanya fokus kepada bisnis yang akan dikerjakan.
2. *Rouse of Code*, dengan menggunakan *software framework* maka pekerjaan yang dibuat akan memiliki struktur yang baru sehingga dapat diterapkan pada proyek-proyek lainnya.
3. Bantuan Komunitas, terdapat banyak komunitas pengguna *software framework* yang siap membantu bila ada kesulitan dalam pengerjaan proyek.
4. Kumpulan *best practice*, sebuah *framework* merupakan kumpulan *best practice* yang sudah teruji sehingga dapat meningkatkan kualitas kode yang akan digunakan.

3.10. Konsep MVC (Model, View, Controler)

MVC adalah konsep dasar yang harus diketahui sebelum menggunakan sebuah software framework. MVC adalah sebuah *patter*n / teknik pemograman yang memisahkan antara desain, data dan proses (Daqiqil, 2011). Adapun komponen-komponen MVC, diantaranya:

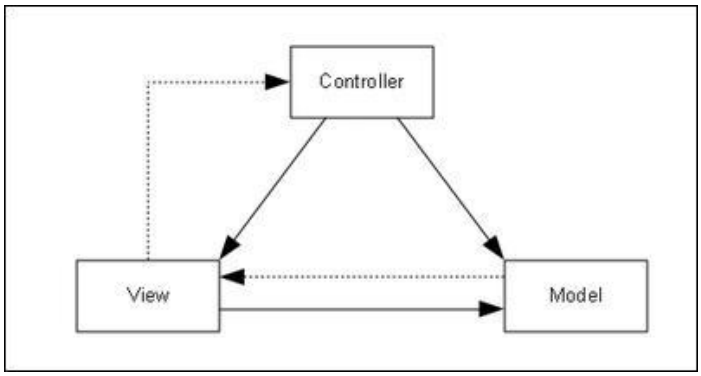
1. *Model*

   Di dalam *model* terdapat *class* dan fungsi untuk mengambil, melakukan *update*, dan menghapus data yang digunakan dalam pembuatan sebuah *website*. Sebuah aplikasi *website* biasanya menggunakan basis data dalam menyimpan data, maka pada bagian model biasanya akan berhubungan dengan perintah-perintah *query* SQL.
2. *View*

   View berhubungann dengan segala sesuatu yang akan ditampilkan kepengguna. V*iew* berisi data-data berupa HTML, CSS, dan Javascript. Bagian ini hanya menampilkan data data hasil dari *model* dan *controller*.
3. *Controller*

   Controller merupakan bagian penghubung antara *view* dan *model.* Terdapat *class* dan fungsi yang memproses permintaa dari *view* ke dalam

struktur data yang ada pada *model*. Tugas *Controller* adalah menyediakan variabel data yang akan ditampilkan di *view*, memanggil *model* untuk mengakses *database*, menyediakan penanganan untuk *error/*kesalahan, mengerjakan proses logika dan melakukan validasi/cek terhadap *input.* Konsep MVC dapat dilihat di Gambar 3.3.



Gambar 3.3 Konsep MVC

3.11. Bahasa Pemrograman

Bahasa pemrograman adalah perangkat lunak atau *software* yang dapat digunakan dalam proses pembuatan program yang melalui beberapa tahapan-tahapan penyelesaian masalah. (Indrajani (2007:22)). Bahasa pemrograman *desktop* salah satunya adalah yaitu Visual Basic (VB), C++ dan C sharp. Sedangkan pemrograman *web* dibagi 3 yaitu PHP, ASP dan HTML. Dan untuk pemrograman *mobile* adalah J2ME.

3.11.1. HTML (Hyper Text Markup Language)

HTML merupakan bahasa markup yang digunakan untuk membuat halaman *web* dan menampilkan berbagai informasi di dalam sebuah *browser* (Karim, 2012). HTML memiliki struktur dasar, adapun struktur dasar dari HTML dapat dilihat pada Gambar 3.

```
<html>
    <head>
        <title>JudulForm/caption</title>
    </head>

    <body>
         isi atau badan web
    </body>
</html>
```

Gambar 3.4 Struktur dasar HTML

Bahasa pemrograman *web* terus mengalami perkembangan. Versi yang terbaru adalah HTML5. Dalam HTML5 terdapat kelas-kelas di dalam *web* sehingga aplikasi yang dibuat menjadi terlihat lebih nyata. (Kessin, 2012). HTML5 didukung dengan javascript, API browser, dan AJAX sehingga memudahkan dalam membuat aplikasi. HTML5 membuat aplikasi *web* menjadi lebih interaktif pada *browser* dari sisi pengguna dengan adanya fitur-fitur multimedia dan grafis.

3.11.2. PHP (Hypertext Preprocessor)

PHP merupakan salah satu bahasa pemograman untuk *web* yang bekerja dari sisi *server*. Hampir semua *website* menggunakan bahasa pemograman PHP dan sangat populer karena menawarkan kemudahan dalam mengembangkan aplikasi Internet, terutama dengan dukungan luar biasa keberbagai *database server*. (Karim, 2012) Keunggulan bahasa pemograman PHP adalah :

1. Cepat, karena disisipkan diantara tag HTML maka proses pembuatannya bisa dilakukan dengan cepat dan mudah.
2. Mudah digunakan, kode program yang dimiliki oleh Bahasa pemograman PHP sangat sederhana sehingga mudah dipahami dan digunakan.
3. Serba guna, dapat dijalankan di semua sistem operasi.
4. Dukungan teknis yang luas.
5. Aman, selama *website* di desain dengan benar, pengguna tidak akan melihat kode PHP yang dibuat.
6. Bisa dimodifikasi sesuai dengan kebutuhan. Struktur dasar PHP adalah :

```
<?php
        .........
?>
```

3.11.3. CSS (*Cascading Style Sheet)*

CSS merupakan *style* yang digunakan untuk mengatur tampilan dari halaman *web*. Dengan menggunakan CSS, maka akan menghasilkan tampilan *web* yang indah dan menarik jauh lebih mudah dan ringkas dibandingkan atribut ditiap tag html. Selain itu bisa membuat tampilan *web* yang dinamis dan fleksibel baik tampilan di *web browser* ataupun ketika halaman *web* tersebut dicetak. (Karim, 2012) Struktur CSS terdiri dari 2 komponen yaitu :

1. *Selector* berfungsi memberi tahu *web browser* bahwa pada elemen mana css akan diterapkan. *Selector* dapat berupa nama tag, *selector class* atau *selector id.*
2. *Declaration* merupakan aturan css yang diterapkan yang mana terdiri dari *property* dan *value.*

   Contoh syntax CSS dapat dilihat pada Gambar 3.5.

```
#footer {

   clear:both;
   position:relative;
   background:#666;
   height:100px;
   margin-bottom:10px;

}
```

Gambar 3.5 Contoh script CSS

3.11.4. Javascript

Javascript merupakan bahasa *scripting* yang didesain untuk membuat halaman *web* menjadi lebih interaktif. Javascript bekerja secara *client-side* (disisi *client*) sehingga javascript bergantung kepada *browser* yang digunakan untuk menampilkan *web* yang mengandung javascript (Karim, 2012). Terdapat contoh script javascript pada Gambar 3.6.

```
1   <HTML>
2      <HEAD>
3         <TITLE>Tampilkan Pesan</TITLE>
4      </HEAD>
5
6      <BODY>
7         <SCRIPT LANGUAGE = "JavaScript">
8            <!--
9               window.alert("Selamat Belajar Javascript");
10           //-->
11        </SCRIPT>
12     </BODY>
13  </HTML>
```

Gambar 3.6 Contoh script Javascript

3.12. Perangkat Lunak Sistem

Pengembangan sistem aplikasi ini menggunakan beberapa perangkat lunak, di antaranya bahasa pemrograman HTML5, PHP, CSS3, javascript, MySQL dan f*ramework* CodeIgniter.

3.12.1. MySQL

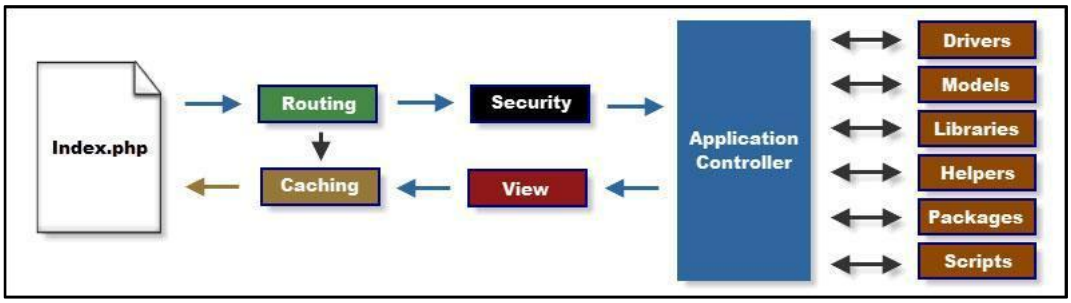
MySQL adalah basis data yang paling banyak digunakan dalam pemrograman *web* dengan bahasa pemrogrman PHP (Nixon, 2009). SQL pada MySQL adalah *Structured Query Language*, yaitu bahasa terstruktur yang menjelaskan permitaan untuk mengakses dan mengelola basis data. MySQL sangat *scalable*, yang berarti dapat berkembang sesuai aplikasi *web* yang dibuat. Tujuan MySQL dibuat adalah untuk memungkinkan permintaan sederhana dari basis data melalui perintah DDL maupun DML.

3.12.2 CodeIgniter

Sebuah *web application framework* yang bersifat *open source*, yang digunakan untuk membangun aplikasi PHP dinamis. Tujuan utama pengembangan CodeIgniter adalah untuk membantu *developer* untuk mengerjakan aplikasi lebih cepat daripada menulis semua *code* dari awal. CodeIgniter menyediakan berbagai macam *library* yang dapat mempermudah dalam pengembangan.

CodeIgniter diperkenalkan pada publik pada tanggal 28 Februari 2006. CodeIgniter dibangun menggunakan konsep MVC (*Model-View-Controller*) *development pattern*. (Basuki, 2010). Dalam CodeIgniter, *browser* berinteraksi melalui *controller* yang bertugas menerima dan membalas semua *request* dari

browser. Untuk data maka *controller* akan meminta ke model dan untuk bagian antarmuka akan meminta ke *view*. Ketika browser meminta sebuah halaman *web* maka *router* akan mencarikan *controller* mana yang harus menangani *request* tersebut. Setelah itu barulah *controller* menggunakan *model* untuk mengakses data dan *view* untuk menampilkan data tersebut. Hal tersebut dapat diilustrasikan seperti Gambar 3.7 (Basuki, 2010).



Gambar 3.7 Ilustrasi sistem kerja framework CodeIgniter

Ada beberapa kelebihan menggunakan CodeIgniter dibandingkan dengan *framework* PHP lainnya, antara lain :

1. Dokumentasi yang lengkap

   Setiap paket instalasi CodeIgniter sudah disertakan *user guide* yang sangat lengkap dan rinci untuk dijadikan sebagai referensi, bahasa yang digunakan pun mudah dipahami karena terdapat contoh penggunaanya.

2. Banyak Komunitas

   Dengan banyaknya komunitas yang berkembang tentang CodeIgniter maka akan memudahkan kita sebagai pengguna untuk berinteraksi satu sama lain.

3. Konfigurasi yang minim

   Untuk menyesuaikan dengan basis data dan keleluasaan *routing* tetap diizinkan melakukan konfigurasi dengan mengubah beberapa file seperti database.php atau autoload.php.

4. Performa yang cepat

   Salah satu alasan mengapa tidak menggunakan *framework* adalah karena eksekusi yang lebih lambat dari PHP *from the scratch*. Tetapi, dengan menggunakan *framework* CodeIgniter maka performa yang didapat akan lebih cepat, bahkan *framework* CodeIgniter merupakan *framework* yang paling cepat dibandingkan *framework* lainnya.

5. Gratis

   CodeIgniter merupakan *framework* yang berlisensi dibawah Apache/BSD *open source*, jadi *framework* ini dapat digunakan secara bebas.

6. Ringan

   Pada *framework* CodeIgniter dapat dikonfigurasi *library* apa saja yang perlu di-*load* sehingga dengan me-*load library* yang dibutuhkan saja akan membuat sistem dapat berjalan dengan ringan dan cepat.

7. Menggunakan konsep MVC

   Framework CodeIgniter menggunakan konsep MVC (*Model View Controller*) sehingga memungkinkan pembagian *layer application, logic* dan *presentation.*

8. Memiliki paket *library* yang lengkap

   Framework CodeIgniter memiliki *library* yang lengkap untuk mengerjakan operasi-operasi yang umum dibutuhkan oleh aplikasi berbasis *web.*

9. URL yang sederhana

   Secara *default*, URL yang dihasilkan CodeIgniter sangat bersih dan *search engine friendly.*

10. *Ekstensibel*

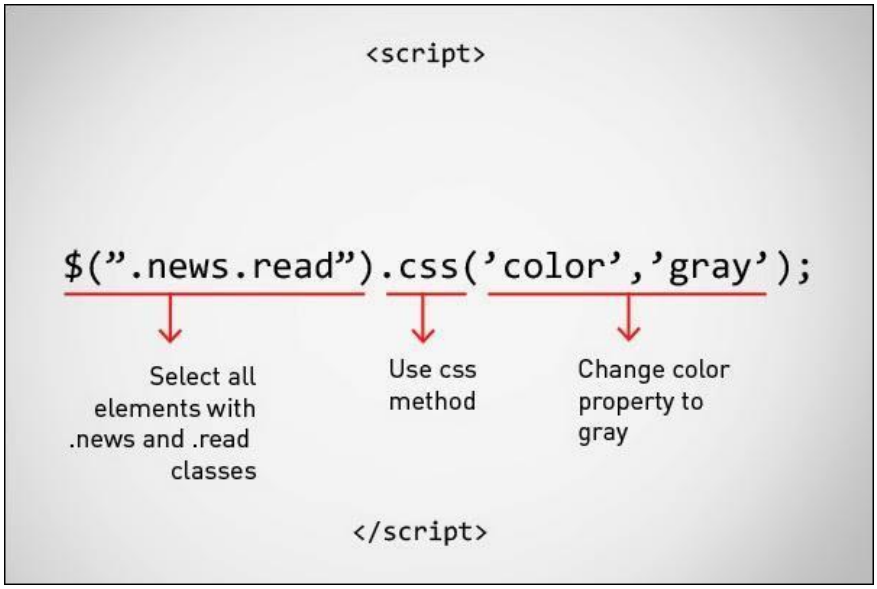
    Sistem dapat dikembangkan dengan mudah dengan menggunakan *plugins* dan *helper* atau menggunakan *hooks.*

11. Tidak memerlukan *template engine*

    Meskipun CodeIgniter dilengkapi dengan *template parser* sederhana yang dapat digunakan, tetapi tidak mengharuskan untuk menggunakannya.

3.12.3. Jquery

JQuery adalah javascript *library* yang cepat dan ringkas. JQuery menyederhanakan arus lintas dokumen HTML, penanganan *event*, animasi, dan interaksi Ajax untuk pengembangan *web* secara cepat. jQuery dirancang untuk mengubah cara menulis JavaScript (Hakim, 2013). Contoh script pada Jquery akan ditampilkan pada Gambar 3.8.



Gambar 3.8 Contoh script Jquery

3.12.4. Bootstrap

Bootstrap adalah sebuah *front-end framework* untuk membangun suatu *web* yang responsif. Framework ini digunakan dalam aplikasi, blog, atau aplikasi CMS lainnya. Bootstrap menggabungkan kombinasi antara HTML, CSS, dan javascript untuk membangun situs tanpa menambahlan banyak *kode.*

Bootstrap merupakan produk *open source* dari Mark Otto dan Jacob Thornton, keduanya merupakan karyawan di Twitter. Sejak bootstrap diluncurkan pada Agustus 2011, *framework* CSS ini mengalami perkembangan yang pesat dan menjadi sangat popular. Tujuan utamanya membuat *web* yang dihasilkan menjadi lebih responsif. Bootstrap sangat mudah digunakan oleh pemula, di dalam bootstrap terdapat tiga direktori utama yaitu css, js, dan img dengan dokumentasi yang lengkap di dalamnya. (Cochran, 2012).

3.3.2. XAMPP

XAMPP merupakan perangkat lunak yang mendukung banyak sistem operasi dan merupakan kombinasi dari berbagai program. XAMPP merupakan singkatan dari X, yaitu menunjukan empat sistem operasi (Windows, Linux, Mac OS dan Solaris), A yaitu menunjukan Apache, M yaitu menunjukan MySQL, P menunjukan PHP dan P menunjukan Perl. Fungsi dari XAMPP adalah server yang berdiri sendiri (*localhost*) yang terdiri atas HTTP *server,* basis data MySQL dan bahasa penerjemah menggunakan PHP. XAMPP dapat diunduh secara gratis pada

situs resmi Apache.

3.12.6. Sublime Text

Sublime text merupakan editor teks yang menggunakan *user interface customer toolkit* yang mengoptimalkan kecepatan. Sublime text merupakan editor teks yang canggih untuk mengelola prosa dan *kode* program. Antarmuka editor ini sangat responsif dengan kinerja yang baik. Keunggulan dari sublime text adalah memiliki beberapa fitur, yaitu *goto anything*, yaitu fitur yang memungkinkan seleksi pada teks, c*ommand pallete, distraction free mode, split editing, instant project switch, plugin API, customize anything* dan memungkinkan untuk lintas *platform.*

3.12.7. StarUML

StarUML adalah *software* pemodelan yang mendukung UML (*Unified Modeling Language).* StarUML merupakan proyek open source untuk pengembangan secara cepat, fleksibel, *extensible, featureful*, dan bebas-tersedia. Tujuan dari proyek StarUML adalah untuk membangun sebuah alat pemodelan perangkat lunak dan juga *platform* yang menarik. UML biasa digunakan untuk mempresentasikan sistem kepada orang-orang yang tidak mengerti bahasa pemograman. Hal ini dikarenakan UML memakai pengGambaran logika algoritma suatu program.

3.12.8. Balsamiq Mockups

Balsamiq Mockups adalah salah satu software yang digunakan dalam pembuatan desain atau *prototyping* dalam pembuatan tampilan user interface sebuah aplikasi. Dengan menggunakan Balsamiq Mockup akan dimudahkan dalam pembuatan user interface karena Balsamiq Mockup sudah menyediakan tools yang dapat memudahkan dalam membuat desain *prototyping* aplikasi yang akan dibuat. *Software* ini berfokus pada konten yang ingin di gambar dan fungsionalitas yang dibutuhkan oleh pengguna.

3.13 Pengertian Ujian

Dalam konteks akademis atau profesional, ujian adalah tes yang bertujuan untuk menentukan kemampuan seseorang. Biasanya ujian tes tertulis, walaupun

beberapa mungkin praktis atau komponen praktis, dan sangat bervariasi dalam struktur, isi dan kesulitan tergantung pada subjek, kelompok usia orang yang diuji dan profesi. Pemeriksaan kompetitif adalah ujian di mana pelamar bersaing untuk sejumlah posisi, sebagai lawan hanya harus mencapai tingkat tertentu untuk lulus.

”Ujian adalah kegiatan untuk mengetahui seberapa tinggi tingkat keberhasilan siswa.” (Suharsimi Arikunto, 2007)

3.14 Pengertian Sistem Ujian Online

Online adalah terhubung, terkoneksi, aktif dan siap untuk operasi, dapat berkomunikasi dengan atau dikontrol oleh komputer. Online ini juga bisa diartikan sebagai suatu keadaan dimana sebuah device (komputer) terhubung dengan device lain, biasanya melalui modem. Online adalah sedang menggunakan jaringan, terhubung dalam jaringan, satu perangkat dengan perangkat lainnya yang terhubung sehingga bisa saling berkomunikasi. Berdasarkan teori online diatas dapat disimpulkan bahwa online adalah suatu keadaan dimana sebuah device (komputer) terhubung dengan device lain dengan menggunakan perangkat modem, sehingga bisa saling berkomunikasi.

Sistem ujian online adalah sistem pengujian baru yang memungkinkan melakukan generate dan random soal secara dinamis. Sehingga pertanyaan pertanyaan dan jawaban yang ditampilkan akan berbeda dari peserta yang satu dengan peserta yang lain sesuai dengan bank soal. Hasil dari ujian online bisa langsung dilihat setelah melakukan ujian tanpa harus menunggu lama.

3.15 Sekilas Tentang SMA Negeri 1 Gombong

Sekolah Menengah Atas Negeri 1 Gombong atau dikenal juga dengan nama smansago yang berada di Gombong, Kebumen. SMA Negeri 1 Gombong menempati gedung di Jl. Sempor Lama No. 64 yang didirikan tahun 1963.

3.15.1 Sejarah SMA Negeri 1 Gombong

Pemrakarsa pendirian SMA Negeri 1 Gombong adalah Bp. S. Suripto, S.H. melalui kontak dan musyawarah dengan para pejabat dan tokoh masyarakat di wilayah Gombong. Pada periode tahun 1963-1965 SMA Negeri 1 Gombong masih bernama SMA dipersiapkan Negeri, dengan Kepala Sekolah Bp. S. Suripto,

S. H. Pada tahun pelajaran pertama jumlah murid sebanyak 65 siswa yang diasuh oleh 7 orang Guru. Selanjutnya jumlah murid berkembang menjadi 150 siswa dalam tahun 1964, dan berubah menjadi 300 siswa dalam tahun 1965. Lokasi SMA Dipersiapkan Negeri Gombong berada di Gedung bekas KWN Wonokriyo. Honorarium Guru dan Karyawan besarnya tidak menentu, bahkan sebagian honornya dibayar dengan beras. Guru-guru yang berstatus mahasiswa harus pulang pergi Yogyakarta – Gombong, mendapat bantuan transportasi gratis dari Perusahaan Bus PRAWARSA, EKA JAYA, MULYO, ESTU. Usaha untuk menegerikan SMA Dipersiapkan Negeri ini terus dilakukan baik oleh Kepala Sekolah maupun Panitia Pendiri. Setelah mengalami lika-liku perjuangan maka pada tanggal 16 Agustus 1965 SMA Dipersiapkan Negeri Gombong Resmi menjadi SMA Negeri Gombong.

Dewasa ini SMA Negeri 1 Gombong memiliki jumlah guru 54 orang, jumlah karyawan 18 orang, jumlah siswa 828 siswa. Sarana yang dimiliki diantaranya yaitu : Ruang kelas 24 kelas, ruang Laboratorium 5 buah terdiri dari: Laboratorium Fisika, Laboratorium Kimia, Laboratorium Biologi, Laboratorium Bahasa, Laboratorium Komputer, dan memiliki beberapa ruang lainnya yaitu: Ruang Perpustakaan, Ruang UKS, Ruang Ketrampilan, Ruang Guru, Ruang Waka Sek, Ruang BP, Ruang Kepala Sekolah, Ruang Tata Usaha, Musholla, Toko Koperasi, Kantin, Tempat Kamar Kecil, Lapangan Basket, Lapangan Volly. Adapun luas tanah yang dimiliki : 6.523 M2.

3.15.2 Visi dan Misi SMA Negeri 1 Gombong

1. Visi

SMA Negeri 1 Gombong dalam prosesnya memiliki visi yaitu :

“Mendidik Siswa Agar Bertakwa, Unggul Dalam Mutu, Berakhlak Mulia, Mandiri dan Berwawasan Global”

2. Misi

SMA Negeri 1 Gombong dalam menjalankan misi yaitu :

1. Melaksanakan pembelajaran dan bimbingan secara efektif dan efisien, dengan memanfaatkan dukungan dokumen kurikulum dan teknologi

pembelajaran seperti penggunaan IT dalam proses pembelajaran berbasis TIK.

2. Menumbuhkan semangat keunggulan kepada seluruh warga sekolah.
3. Membekali siswa dengan keunggulan kompetitif di bidang :
    a. Ilmu-ilmu dasar
    b. Penguasaan IMTAK dan IPTEK
    c. Kemandirian dan berwawasan global
4. Mendorong dan membantu siswa untuk mengenali dan mengembangkan potensi dirinya.
5. Menumbuhkan penghayatan terhadap ajaran agama serta budaya dan karakter bangsa sehingga terbentuk manusia dengan kepribadian dan perilaku luhur.
6. Menerapkan manajemen partisipatif dengan melibatkan seluruh warga sekolah dan kelompok kepentingan yang terkait.

BAB IV
ANALISIS DAN PERANCANGAN SISTEM

4.1 Analisis Sistem

Pembuatan sebuah sistem memerlukan sebuah proses pemahaman mengenai sistem yang akan dibangun. Proses pemahaman tersebut merupakan tahapan analisis pada Sistem Informasi Ujian Online Berbasis Web di SMA Negeri 1 Gombong. Sistem Informasi Ujian Online merupakan sebuah sistem informasi yang digunakan untuk memudahkan guru dan siswa dalam proses pelaksanaan ujian, memudahkan guru dalam mengoreksi hasil ujian siswa, melatih kejujuran siswa, serta efisiensi penggunaan kertas sehingga dapat menghemat biaya pelaksanaan ujian di SMA Negeri 1 Gombong yang membutuhkan biaya tidak sedikit. Pelaksanaan ujian online dapat diikuti oleh seluruh siswa dalam satu waktu. Siswa dapat melihat hasil penilaian ujian secara langsung dan guru dapat mengelola data soal dan hasil ujian. Berikut akan dibahas analisis Sistem Informasi Ujian Online yang meliputi analisis sistem yang sudah ada, identifikasi masalah, analisis kebutuhan fungsional dan analisis kebutuhan non fungsional.

4.1.1 Analisis Sistem Yang Ada

SMA Negeri 1 Gombong sudah menggunakan aplikasi Moodle yang digunakan untuk pelaksanaan ujian online Ulangan Harian. Aplikasi Moodle tersebut merupakan sistem e-Learning yang digunakan untuk memudahkan proses belajar mengajar di SMA Negeri 1 Gombong. Saat ini aplikasi ini masih digunakan untuk melaksanakan ulangan harian yang digunakan 1 sampai 3 kelas dalam satu waktu. Sistem tersebut dirasa masih kurang *user friendly* dan tidak bisa menampung data ujian yang digunakan langsung seluruh siswa untuk Ujian Akhir Semester.

4.1.2 Identifikasi Masalah

Dalam pelaksanaan ujian di SMA Negeri 1 Gombong terdapat berbagai permasalahan yang ada saat ini, yaitu:

1. Aplikasi yang ada saat ini tidak *user friendly*.

2. Tidak bisa melaksanakan ujian secara bersama-sama seluruh siswa, seperti Ujian Akhir Semester sehingga masih menggunakan cara manual yang membutuhkan biaya banyak dan membuang waktu saat persiapan maupun sesudah pelaksanaan ujian.

4.1.3 Analisis Kebutuhan Fungsional

Analisis Kebutuhan Fungsional merupakan kegiatan dalam mendeskripsikan fungsionalitas atau layanan yang diharapkan akan diberikan oleh sistem. Adapun kebutuhan fungsional dari sistem yang dikembangkan adalah sebagai berikut :

1. Sistem dapat menampilkan data paket soal.
2. Sistem dapat menampilkan nilai hasil ujian secara langsung setelah ujian selesai.
3. Sistem dapat menampilkan riwayat hasil ujian untuk siswa.
4. Sistem dapat menampilkan laporan rekap nilai ujian.

Adapun kebutuhan fungsional *admin* dari sistem yang dikembangkan adalah sebagai berikut :

1. *Admin* dapat melakukan *login* untuk masuk ke sistem.
2. *Admin* dapat menambah, mengubah, dan menghapus tahun ajaran
3. Admin dapat menambah dan menghapus data kelas
4. *Admin* dapat menambah, merubah, dan menghapus data guru.
5. *Admin* dapat menambah, merubah, dan menghapus data siswa.
6. *Admin* dapat menambah, merubah, dan menghapus data mapel.
7. *Admin* dapat menambah dan menghapus data pengajar.
8. *Admin* dapat menambah, merubah, dan menghapus data paket soal.
9. *Admin* dapat menambah, mengubah, dan menghapus data pertanyaan.
10. *Admin* dapat menambah, mengubah, dan menghapus proyek ujian.
11. *Admin* dapat melihat seluruh data nilai hasil ujian.
12. *Admin* dapat mencetak hasil ujian.
13. *Admin* dapat mencetak paket soal.

Adapun kebutuhan fungsional guru dari sistem yang dikembangkan adalah sebagai berikut :

1. Guru dapat melakukan *login* untuk masuk ke sistem.
2. Guru dapat melihat data siswa.
3. Guru dapat melihat data mapel dan pengajar.
4. Guru dapat melihat data proyek ujian.
5. Guru dapat menambah, mengubah, dan menghapus data paket soal.
6. Guru dapat menambah, mengubah, dan menghapus data pertanyaan.
7. Guru dapat melihat seluruh data nilai hasil ujian.
8. Guru dapat mencetak hasil ujian.
9. Guru dapat mencetak soal ujian.

Adapun kebutuhan fungsional siswa dari sistem yang dikembangkan adalah sebagai berikut :

1. Siswa dapat melakukan *login* untuk masuk ke sistem.
2. Siswa dapat melakukan ujian sesuai dengan kelasnya masing-masing.
3. Siswa dapat melihat hasil nilai semua ujian yang telah diikuti.

4.1.4 Analisis Kebutuhan Non-fungsional

Analisis kebutuhan non fungsional menjelaskan kebutuhan di luar lingkungan sistem yang mendukung sistem aplikasi berjalan. Adapun kebutuhan non-fungsional dari sistem yang dikembangkan sebagai berikut :

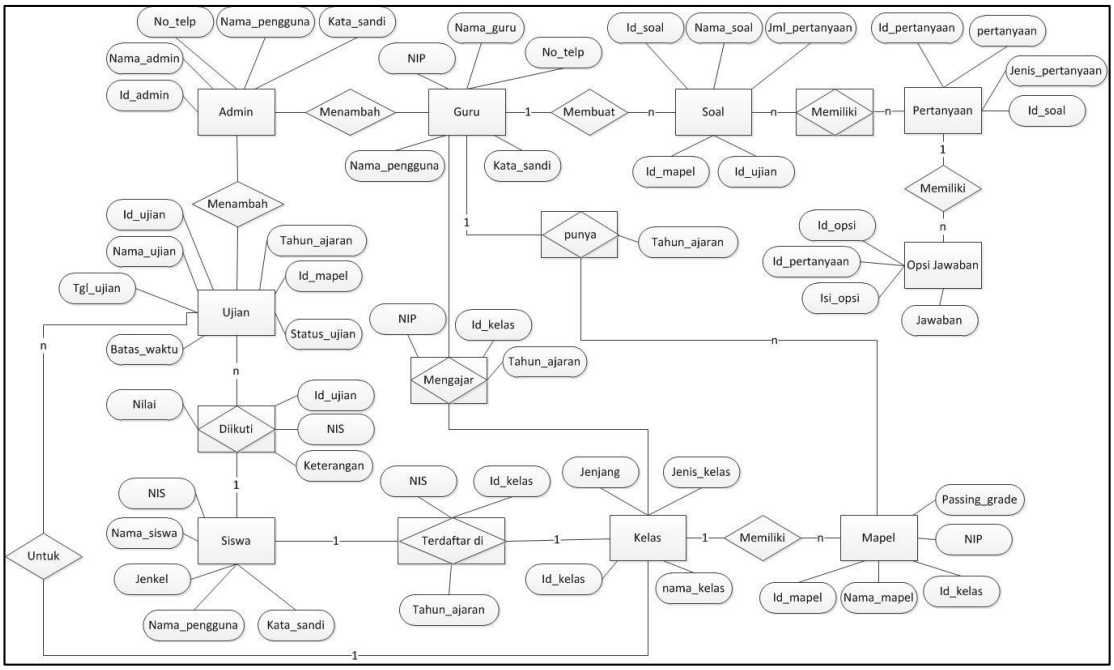
1. Sistem harus dapat berjalan pada komputer yang memiliki spesifikasi minimal RAM 512 MB, OS Windows 7.
2. Sistem dapat dijalankan menggunakan *web browser*.
3. Sistem dapat digunakan pada php versi 5 keatas.
4. Sistem hanya dapat diakses melalui komputer yang terhubung dengan jaringan lokal sekolah.
5. Sistem dapat diakses oleh banyak user dalam waktu yang bersamaan.
6. Perlu adanya pelatihan bagi calon pengguna.

4.2 Perancangan Sistem

Penelitian ini menggunakan perancangan basis data dan perancangan proses menggunakan UML *(use case diagram, activity diagram, class diagram).*

4.2.1 Perancangan Basis Data

Perancangan basis data pada sistem ini dibuat menggunakan *Entity Relationship Diagram* (ERD) yang nantinya dapat diimplementasikan dalam pembuatan tabel-tabel basis data. *Entity Relationship Diagram* (ERD) merupakan perangkat yang digunakan untuk menggambarkan diagram yang berisi entitas maupun relasi yang ada dalam basis data. ERD pada Sistem Ujian Online ini menggambarkan hubungan antar entitas yang kemudian dapat dijadikan acuan dalam pembuatan tabel-tabel basis data. Relasi utama dari ERD ini adalah acuan dalam pembuatan tabel-tabel basis data. ERD Sistem Ujian Online Berbasis Web (Studi kasus: SMA Negeri 1 Gombong) dapat dilihat pada Gambar 4.1.



Gambar 4.1 Entity Relationship Diagram (ERD)

Berdasarkan rancangan table ERD yang telah digambarkan sebelumnya, maka Sistem Ujian Online mempunyai beberapa tabel diantaranya adalah sebagai berikut :

1. Tabel User

Tabel User merupakan tabel untuk menyimpan data user admin, guru, dan siswa. Tabel user ini terdiri dari username, password, dan level. Tabel User ditujukan pada Tabel 4.1.

Tabel 4.1 Tabel User

Atribut	Tipe Data	Panjang	Keterangan
Username	Varchar	50	*Primary key*
Password	Varchar	50	Password
Level	Enum	('Admin','Guru', 'Siswa')	Level atau status user

2. Tabel Admin

Tabel Admin merupakan tabel untuk menyimpan data admin. Tabel admin ini terdiri dari id_admin, nama_admin, no_telp, nama_pengguna, dan kata_sandi. Tabel Admin ditujukan pada Tabel 4.2.

Tabel 4.2 Tabel Admin

Atribut	Tipe Data	Panjang	Keterangan
id_admin	Integer		*Primary key auto increment*
nama_admin	Varchar	50	Nama lengkap admin
no_telp	Varchar	15	No telepon admin
Username	Varchar	20	*Foreign key* dari tabel user

3. Tabel Guru

Tabel guru merupakan tabel untuk menyimpan data Guru di SMA Negeri 1 Gombong. Tabel Guru terdiri dari NIP, nama_guru, no_telp, dan username. Tabel Guru ditujukan pada Tabel 4.3

Tabel 4.3 Tabel Guru

Atribut	Tipe Data	Panjang	Keterangan
Nip	Bigint	20	*Primary key*
nama_guru	Varchar	50	Nama lengkap guru

no_telp	Varchar	15	No telepon guru
username	Varchar	20	username guru

4. Tabel Siswa

Tabel siswa merupakan tabel untuk menyimpan data siswa di SMA Negeri 1 Gombong. Tebal siswa terdiri dari NIS, nama_siswa, jenkel, dan username. Tabel Siswa ditujukan pada Tabel 4.4.

Tabel 4.4 Tabel Siswa

Atribut	Tipe Data	Panjang	Keterangan
Nis	Bigint	20	*Primary key auto increment*
nama_siswa	Varchar	50	Nama lengkap siswa
jenkel	enum	('L','P')	Jenis kelamin siswa
username	Varchar	20	username siswa

5. Tabel Ujian

Tabel Ujian merupakan tabel untuk menyimpan data proyek ujian yang dibuat oleh admin untuk setiap Ujian yang akan dilaksanakan. Tabel Ujian terdiri dari id_ujian, nama_ujian, tgl_ujian, id_tahun, tgl_mulai, tgl_selesai, dan status_ujian. Tabel Ujian ditujukan pada Tabel 4.5

Tabel 4.5 Tabel Ujian

Atribut	Tipe Data	Panjang	Keterangan
id_ujian	Varchar	10	*Primary key auto increment*
nama_ujian	Varchar	50	Nama judul ujian
tgl_ujian	Date		Tanggal pelaksanaan ujian
id_tahun	Year	10	*Foreign key* dari tabel Tahun Ajaran
tgl_mulai	date		Tanggal dimulainya ujian
tgl_selesai	date	10	*Tanggal diakhirinya ujian*
status_ujian	enum	(‘Aktif’,‘Tidak Aktif’)	Status Ujian apakah sedang berlangsung atau tidak.

6. Tabel Tahun Ajaran

Tabel Tahun adalah tabel untuk menyimpan data tahun ajaran di SMA Negeri 1 Gombong saat pelaksanaan ujian. Tabel Tahun terdiri dari id_tahun, tahun_ajaran, dan status_tahun. Tabel Tahun ditujukan pada Tabel 4.6.

Tabel 4.6 Tabel Tahun Ajaran

Atribut	Tipe Data	Panjang	Keterangan
id_tahun	Integer	11	*Primary key auto increment*
tahun_ajaran	Varchar	15	Tahun ajaran
status_tahun	Enum	('Aktif','Tidak Aktif')	Status apakah aktif atau tidak

7. Tabel Mapel

Tabel Mapel merupakan table untuk menyimpan data Mata Pelajaran yang ada di SMA Negeri 1 Gombong yang digunakan untuk pelaksanaan ujian. Tabel Mapel terdiri dari id_mapel, nama_mapel, dan kkm. Tabel Mapel ditujukan pada Tabel 4.7.

Tabel 4.7 Tabel Mapel

Atribut	Tipe Data	Panjang	Keterangan
id_mapel	Varchar	10	*Primary key auto increment*
nama_mapel	Varchar	50	Nama Mata Pelajaran
Kkm	Integer	11	Kriteria ketuntasan minimal Ujian di Mapel tertentu

8. Tabel Guru Mapel

Tabel Guru Mapel merupakan tabel untuk menyimpan data Pengajar mata pelajaran di suatu kelas pada tahun tertentu, yang terdiri dari nip, id_mapel, id_kelas dan id_tahun. Tabel Guru Mapel ditujukan pada Tabel 4.8.

Tabel 4.8 Tabel Guru Mapel

Atribut	Tipe Data	Panjang	Keterangan
Id_grmapel	Integer	11	*Primary key auto increment*

Nip	Bigint	20	*Foreign key* dari tabel Guru
Id_mapel	Varchar	10	*Foreign key* dari tabel Mapel
id_kelas	Integer	11	*Foreign key* dari tabel Kelas
Id_tahun	Integer	11	*Foreign key* dari tabel Tahun

9. Tabel Kelas

Tabel Kelas merupakan table untuk menyimpan data Kelas yang ada di SMA Negeri 1 Gombong. Tabel Kelas terdiri dari id_kelas, jenjang, jenis_kelas, dan nama kelas. Tabel Kelas ditujukan pada Tabel 4.9.

Tabel 4.9 Tabel Kelas

Atribut	Tipe Data	Panjang	Keterangan
id_kelas	Integer	11	*Primary key auto increment*
Jenjang	enum	('X', 'XI', 'XII')	Jenjang Kelas X, XI, dan XII.
jenis_kelas	enum	('MIPA', 'IPS')	Jenis MIPA atau IPS untuk setiap jenjang
nama_kelas	Char	2	Nama dari kelas (1,2,3,4,5, dst)

10. Tabel Kelas Siswa

Tabel Kelas Siswa merupakan tabel untuk menyimpan data Kelas Siswa yang terdiri dari nis, id_kelas, dan id_tahun. Tabel Kelas Siswa ditujukan pada Tabel 4.10.

Tabel 4.10 Tabel Kelas Siswa

Atribut	Tipe Data	Panjang	Keterangan
Nis	Bigint	20	*Foreign key* dari tabel Siswa
id_kelas	Integer	11	*Foreign key* dari tabel Kelas
Id_tahun	Integer	11	*Foreign key* dari tabel Tahun

11. Tabel Hasil Ujian

Tabel Hasil Ujian merupakan tabel untuk menyimpan data hasil Ujian Siswa seperti nilai dan keterangan lulus atau tidaknya dengan kkm yang sudah ditentukan. Tabel Ujian Siswa terdiri dari id_hasil, id_soal, nis, nilai_ujian, dan keterangan. Tabel Ujian Siswa ditujukan pada Tabel 4.11.

Tabel 4.11 Tabel Hasil Ujian

Atribut	Tipe Data	Panjang	Keterangan
Id_hasil	Integer		*Primary key auto increment*
id_soal	Integer	11	*Foreign key* dari tabel Soal
Nis	Bigint	20	*Foreign key* dari tabel Siswa
nilai_ujian	Integer		Nilai hasil ujian siswa
keterangan	Enum	('Lulus','Tidak Lulus')	Keterangan Lulus atau tidak

12. Tabel Soal

Tabel Soal merupakan tabel untuk menyimpan data paket Soal. Tabel Soal terdiri dari id_soal, nama_soal, waktu_mulai, waktu_akhir, id_mapel, id_kelas dan id_ujian. Tabel Soal ditujukan pada Tabel 4.12.

Tabel 4.12 Tabel Soal

Atribut	Tipe Data	Panjang	Keterangan
id_soal	Varchar	10	*Primary key auto increment*
nama_soal	Varchar	50	Nama Soal sesuai Mata Pelajaran
waktu_mulai	timestamp		Waktu mulai pengerjaan soal
waktu_akhir	timestamp		Waktu akhir pengerjaan soal
id_mapel	Varchar	10	*Foreign key* dari tabel Mapel
id_kelas	Integer	11	*Foreign key* dari tabel Kelas
id_ujian	Varchar	10	*Foreign key* dari tabel Ujian

13. Tabel Pertanyaan

Tabel Pertanyaan merupakan tabel untuk menyimpan data pertanyaan dari soal. Tabel Pertanyaan ini terdiri dari id_pertanyaan, pertanyaan, jenis_pertanyaan, id_soal, dan gambar. Tabel Pertanyaan ditujukan pada Tabel 4.13

Tabel 4.13 Tabel Pertanyaan

Atribut	Tipe Data	Panjang	Keterangan
id_pertanyaan	Integer	11	*Primary key auto increment*
Pertanyaan	longtext		Text pertanyaan
jenis_pertanyaan	Varchar	20	Jenis dari pertanyaan
id_soal	Varchar	11	*Foreign key* dari tabel Soal
Gambar	text		Gambar untuk pertanyaan

14. Tabel Opsi Jawaban

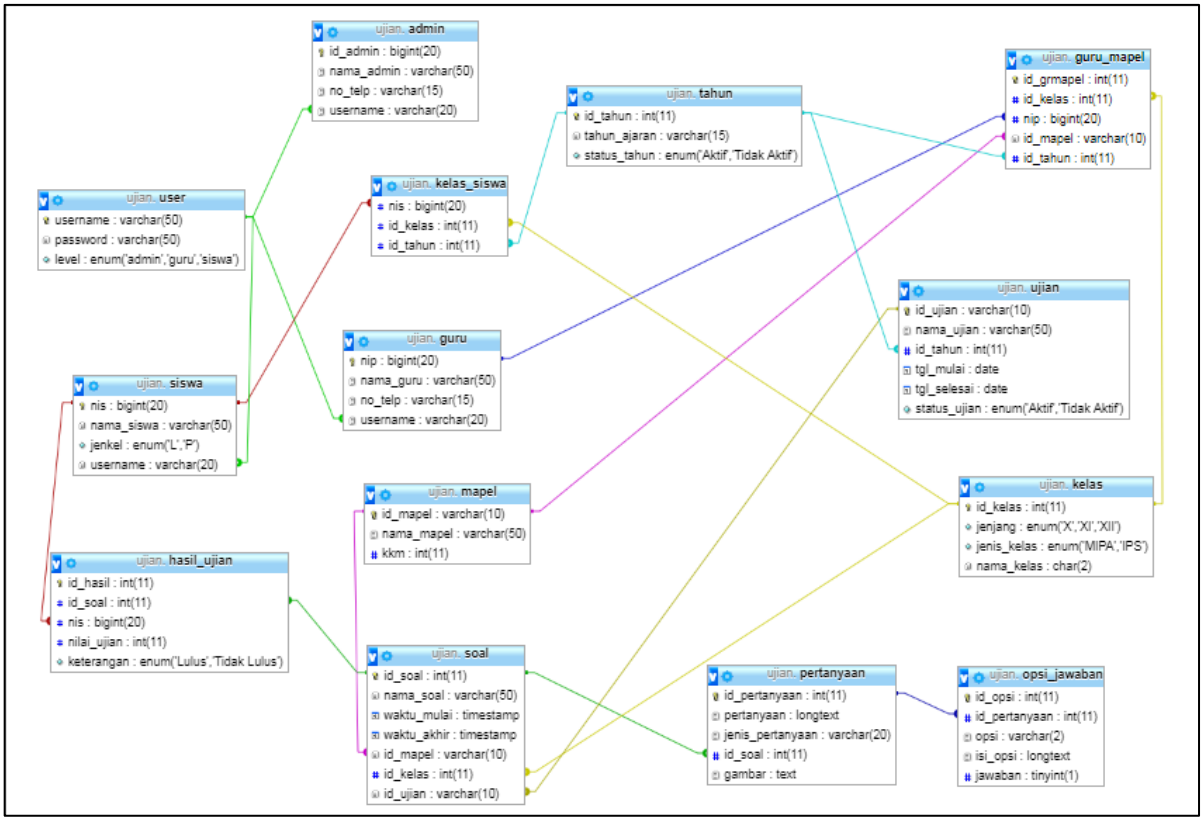
Tabel Opsi Jawaban merupakan tabel untuk menyimpan data opsi jawaban yang ada pada Soal. Tabel Opsi Jawaban terdiri dari id_opsi, id_pertanyaan, isi_jawaban, gambar dan jawaban. Tabel Opsi Jawaban ditujukan pada Tabel 4.14

Tabel 4.14 Tabel Opsi Jawaban

Atribut	Tipe Data	Panjang	Keterangan
id_opsi	Integer	10	*Primary key auto increment*
id_pertanyaan	Varchar	10	*Foreign key* dari tabel Pertanyaan
isi_opsi	Varchar	100	Isi dari pilihan opsi jawaban
gambar	text		Gambar untuk setiap opsi
jawaban	Boolean	(‘0’,‘1’)	Jawaban yang benar

Relasi antar tabel berisi tabel-tabel yang merupakan implementasi dari ERD yang telah dibuat. Setiap tabel dihubungkan oleh sebuah garis yang menggambarkan relasi dan kardinalitasnya. Garis ini menghubungkan foreign key suatu tabel dengan tabel lainnya. Dari tabel-tabel yang telah dirancang

sebelumnya, maka dapat dibuat rancangan relasi antar tabel yang ditunjukan pada Gambar 4.2.



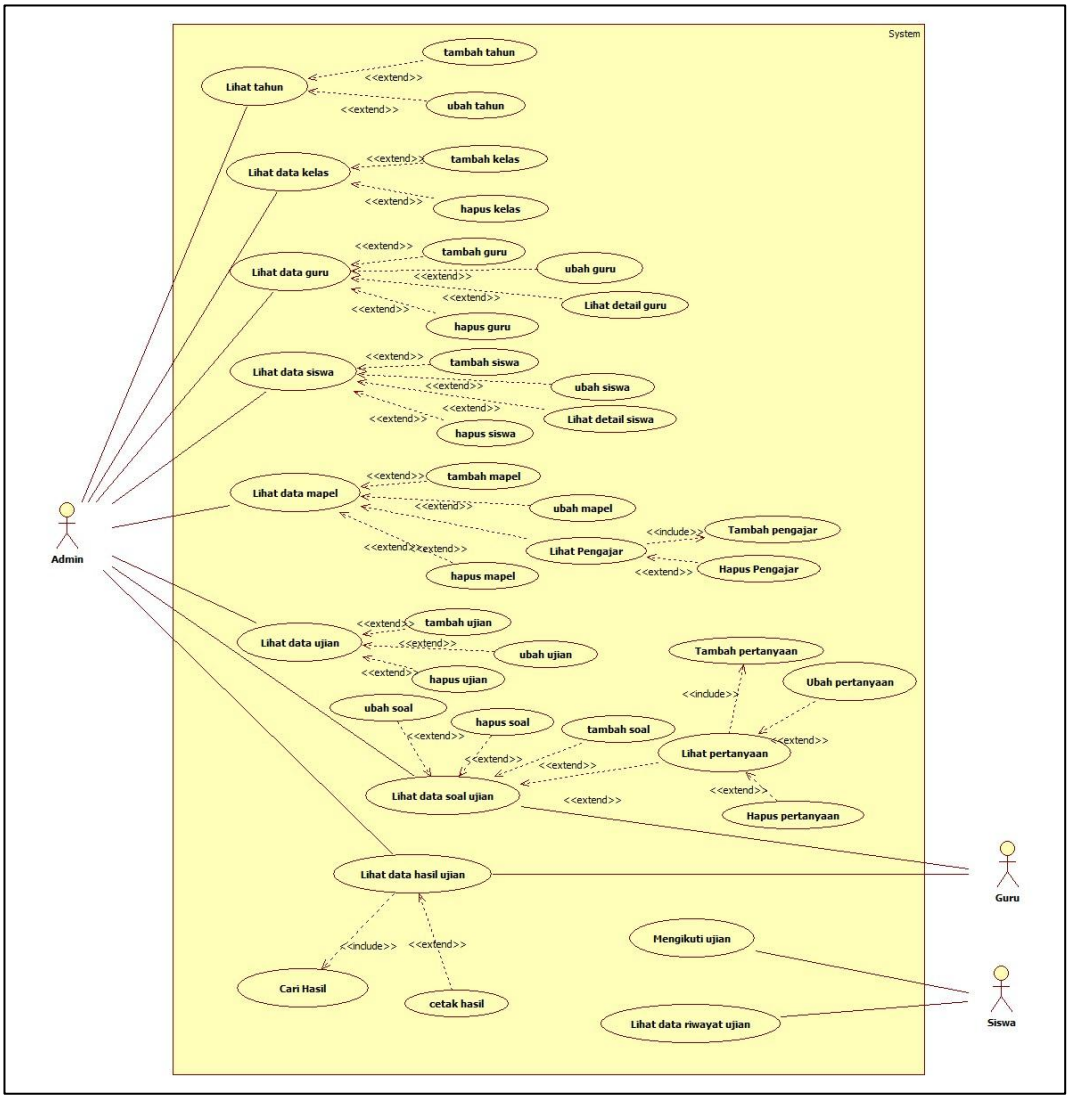
Gambar 4.2 Relasi Antar Tabel

4.2.2 Perancangan Proses

Perancangan proses menjelaskan gambaran proses-proses atau alur berjalan pada Sistem Ujian Online yang akan dibangun sehingga tercipta suatu informasi yang benar. Perancangan antarmuka proses dalam pembangunan Sistem Ujian Online dibuat dengan menggunakan *Unified Modelling Language* (UML) yang dapat digambarkan dengan *use case* diagram, *activity* diagram, dan *class* diagram.

1. *Use case* Diagram

Use case diagram digunakan untuk menggambarkan interaksi sistem, sistem eksternal, dan pengguna. Berdasarkan hasil analisis kebutuhan *user*, *use case* diagram pada Sistem Informasi Ujian Online yang akan dibangun terdapat tiga aktor, yaitu *admin, guru dan siswa*. *Use case* diagram Sistem Informasi Ujian Online di perlihatkan pada Gambar 4.3.



Gambar 4.3 ***Usecase Diagram***

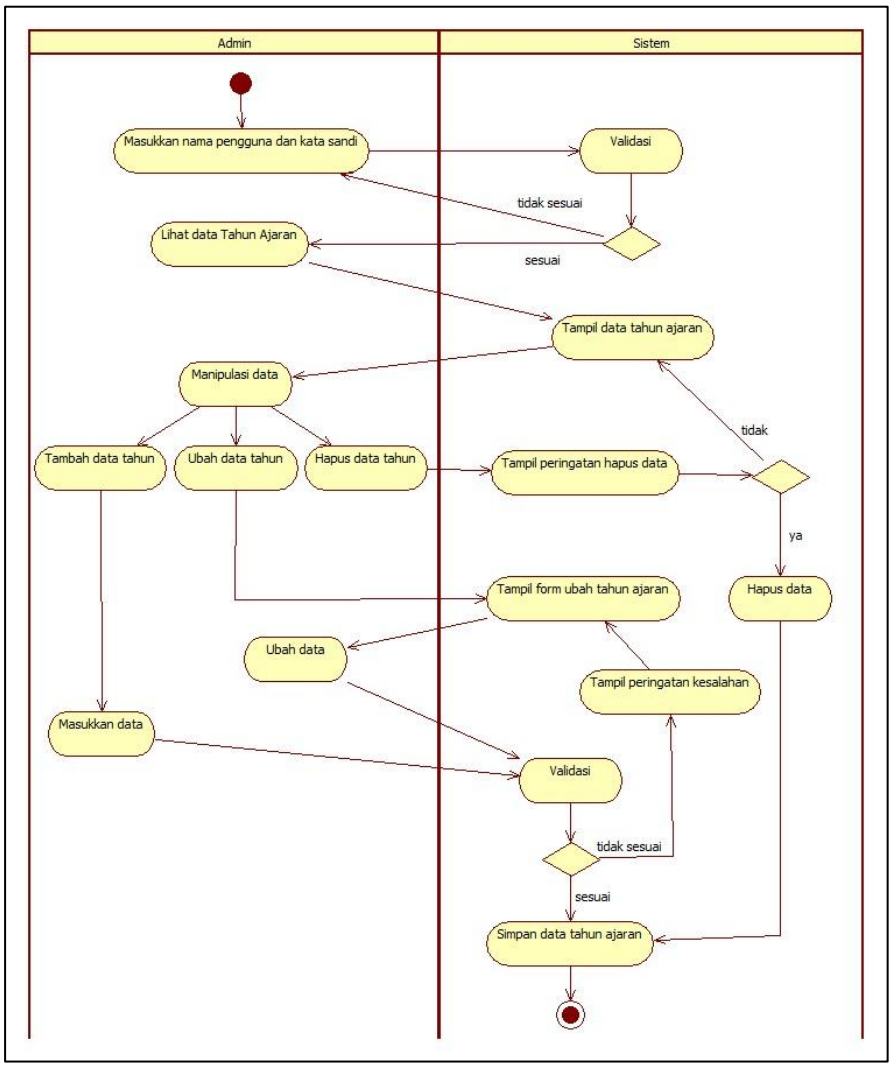
2. *Activity* Diagram

Activity diagram digunakan untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis atau use case yang ada pada Sistem Informasi Ujian Online.

a. *Activity* diagram *admin* mengelola data Tahun Ajaran

Dalam *activity* diagram *admin* mengelola data tahun ajaran mempunyai fitur yaitu admin dapat melihat, menambah, mengubah, dan menghapus data tahun ajaran. Setelah melakukan pengolahan data maka data yang baru akan disimpan di database. *Activity* diagram admin mengelola data tahun ajaran diperlihatkan pada

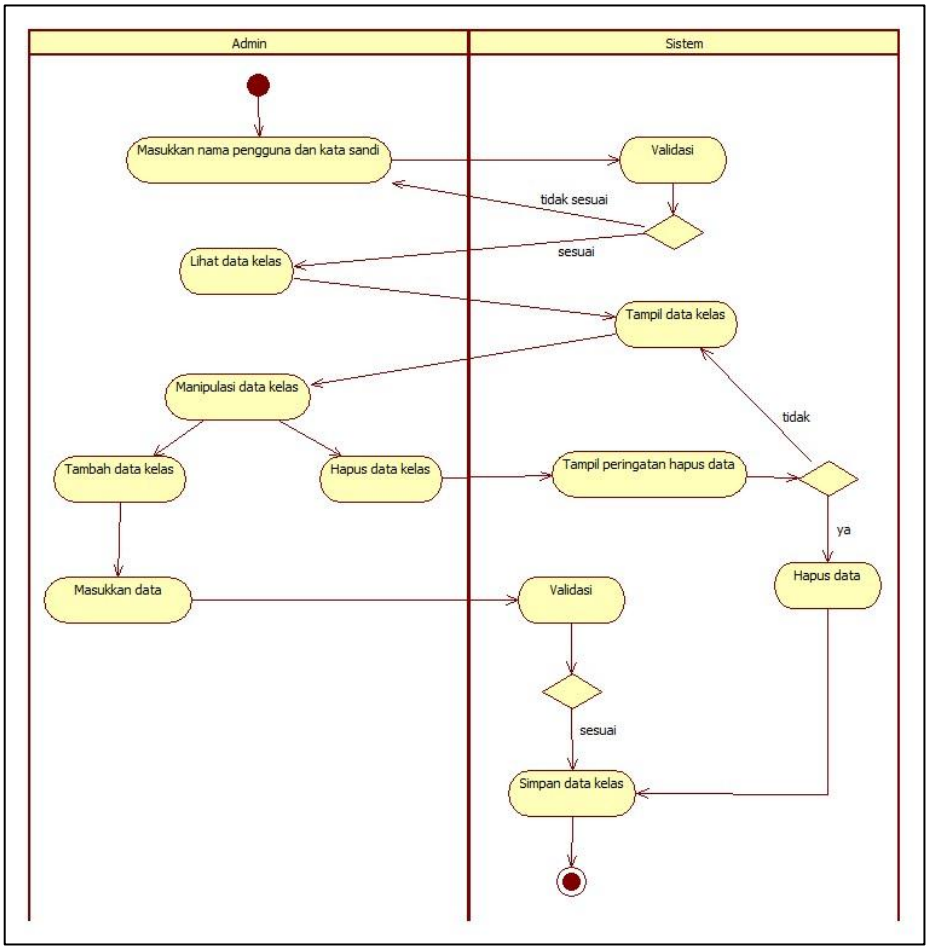
gambar 4.4.



Gambar 4.4 Activity Diagram Admin Mengelola Data Tahun Ajaran

b. *Activity* diagram *admin* mengelola data kelas

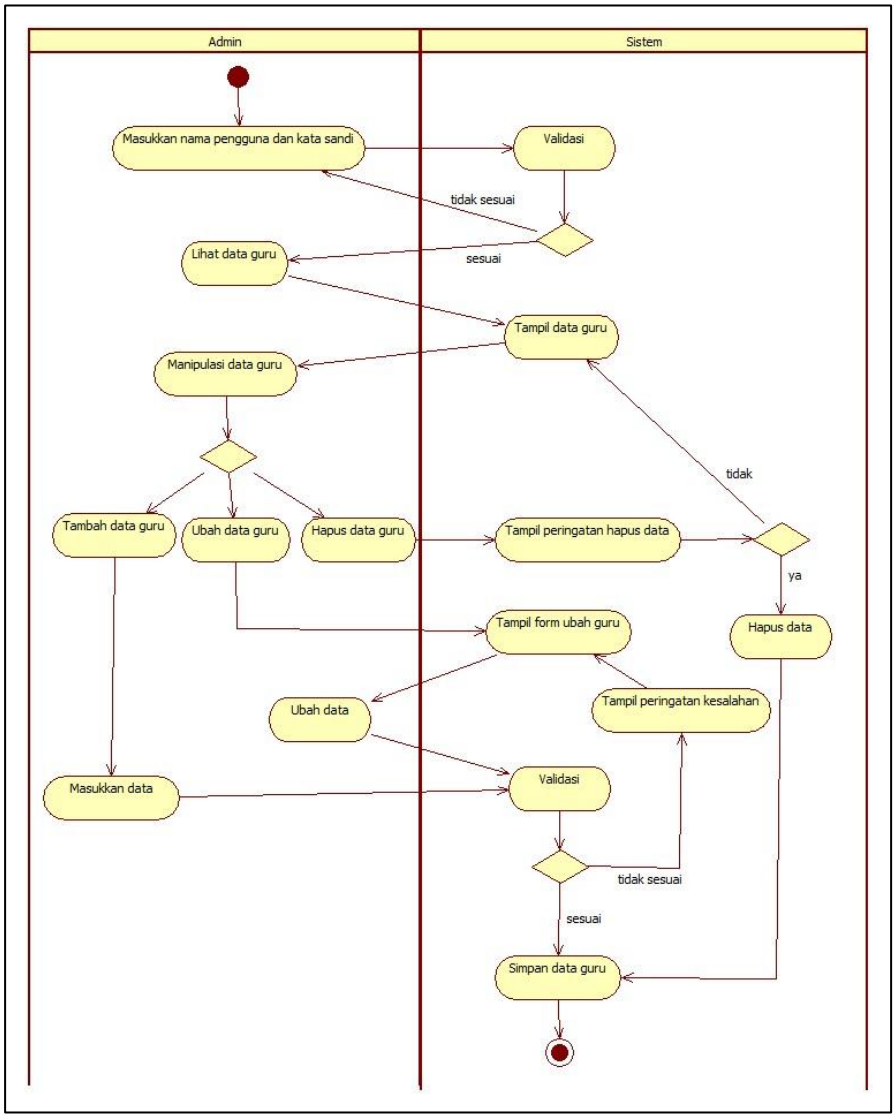
Dalam *activity* diagram admin mengelola data kelas mempunyai fitur yaitu admin dapat melihat, menambah, dan menghapus data kelas. Setelah melakukan pengolahan data maka data yang baru akan disimpan di database. *Activity* diagram *admin* mengelola data kelas diperlihatkan pada gambar 4.5.



Gambar 4.5 Activity Diagram Admin Mengelola Data Kelas

c. *Activity* diagram *admin* mengelola data guru

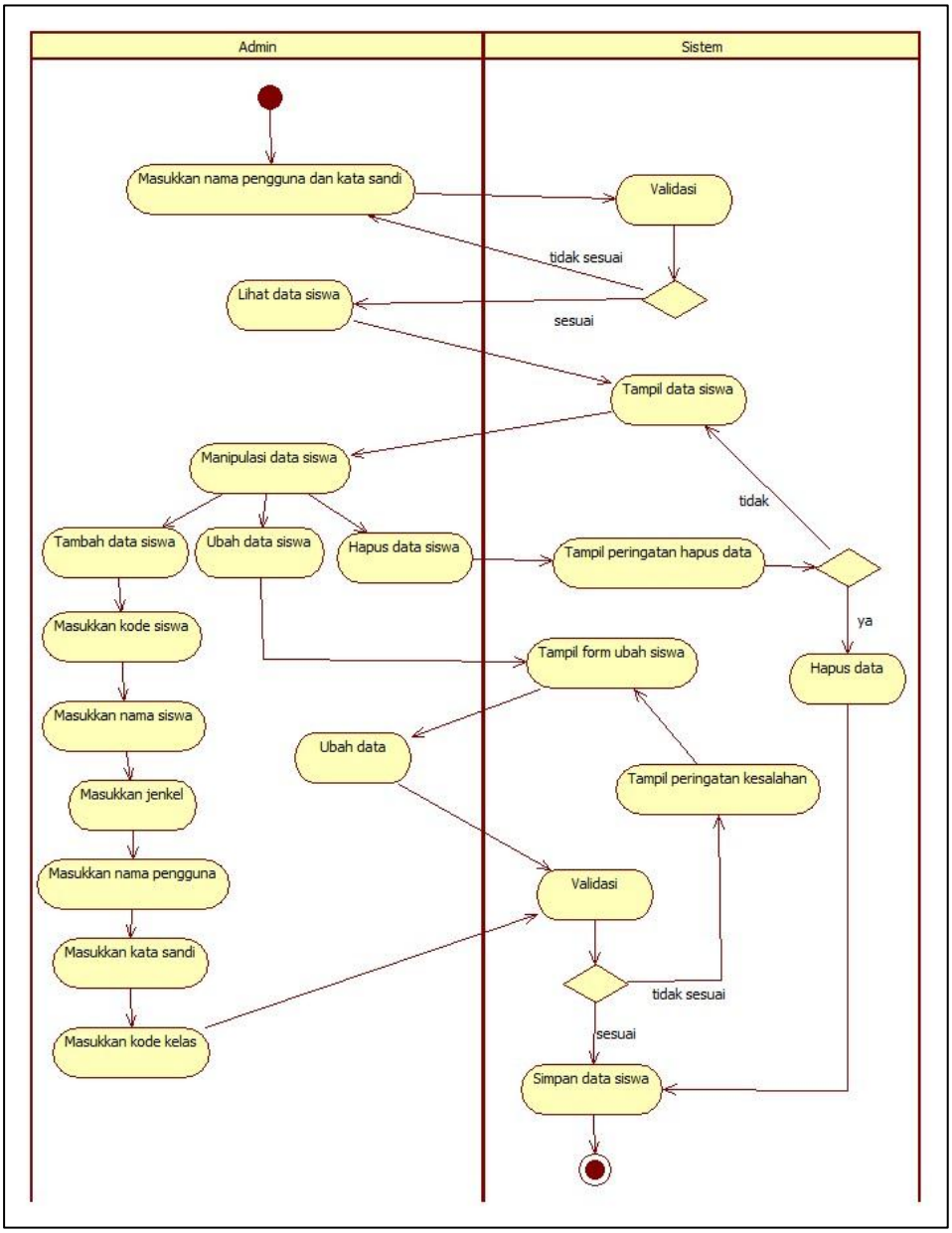
Dalam *activity* diagram *admin* mengelola data guru mempunyai fitur yaitu admin dapat melihat, menambah, mengubah, dan menghapus data guru. Setelah melakukan pengolahan data maka data yang baru akan disimpan di database. *Activity* diagram *admin* mengelola data guru diperlihatkan pada gambar 4.6.



Gambar 4.6 Activity Diagram Admin Mengelola Data Guru

d. *Activity* diagram *admin* mengelola data siswa

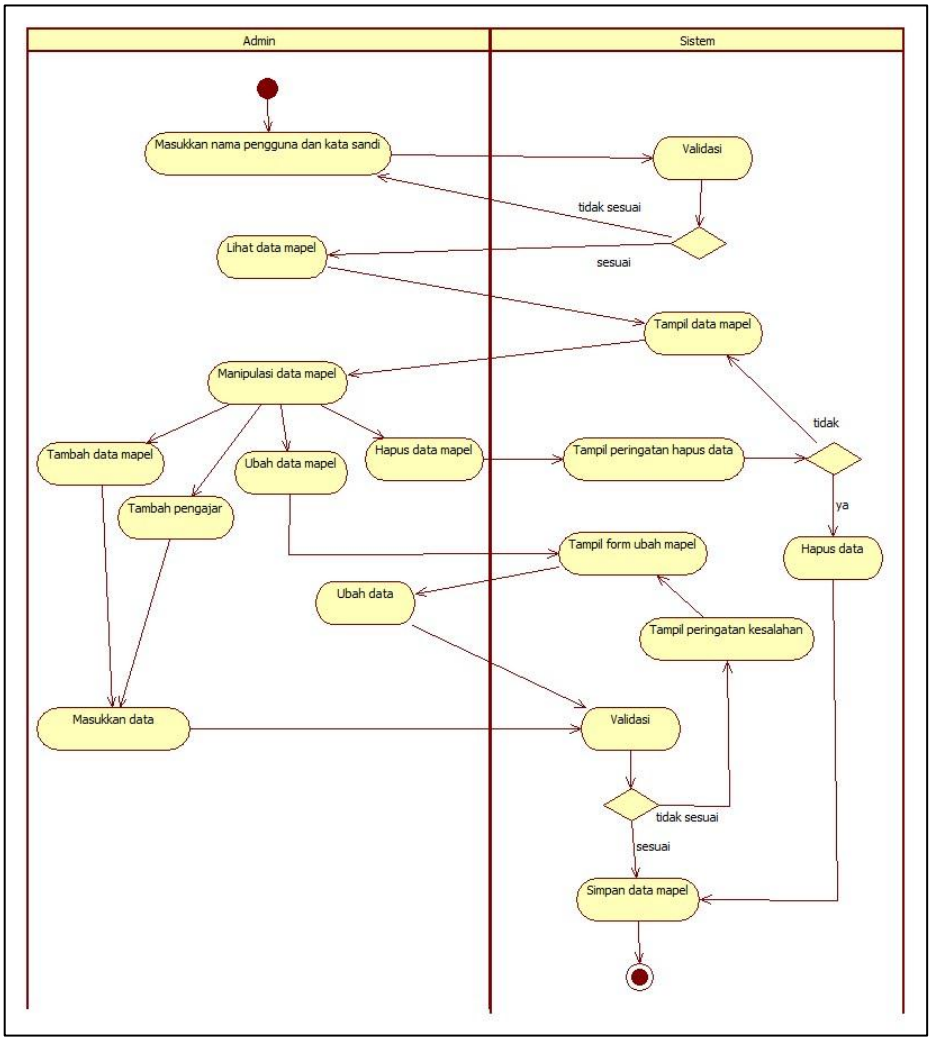
Dalam *activity* diagram admin mengelola data siswa mempunyai fitur yaitu admin dapat melihat, menambah, mengubah, dan menghapus data siswa. Setelah melakukan pengolahan data maka data yang baru akan disimpan di database. *Activity* diagram *admin* mengelola data siswa diperlihatkan pada gambar 4.7.



Gambar 4.7 Activity Diagram Admin Mengelola Data Siswa

e. *Activity* diagram *admin* mengelola data mapel

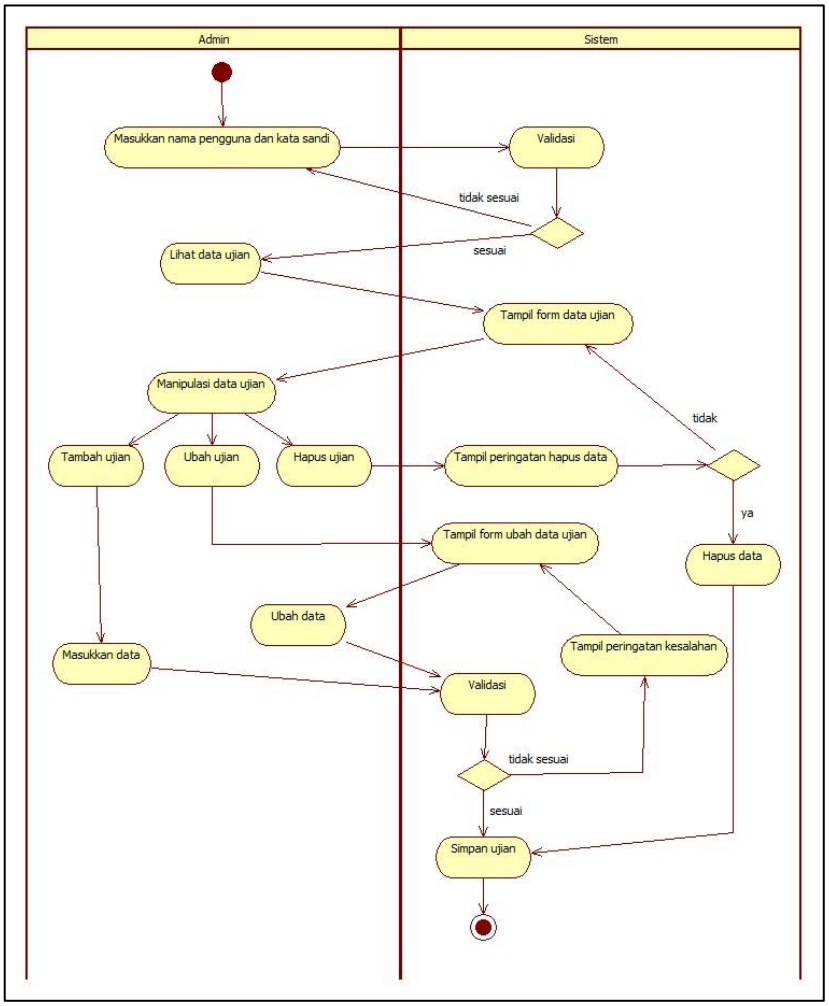
Dalam *activity* diagram admin mengelola data mapel mempunyai fitur yaitu admin dapat melihat, menambah, mengubah, menghapus data mapel, dan menambahkan pengajar. Setelah melakukan pengolahan data maka data yang baru akan disimpan di database. *Activity* diagram *admin* mengelola data mapel diperlihatkan pada gambar 4.8.



Gambar 4.8 Activity Diagram Admin Mengelola Data Mapel

f. *Activity* diagram *admin* mengelola data ujian

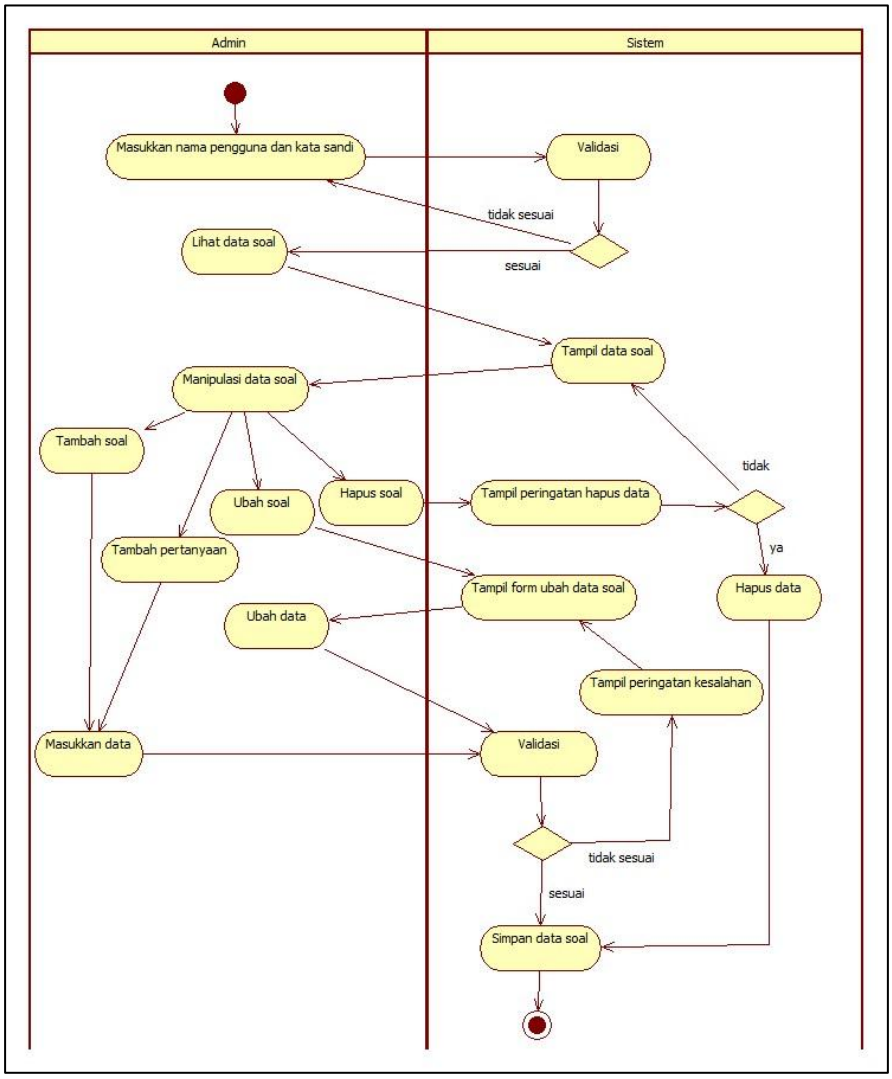
Dalam *activity* diagram admin mengelola data ujian mempunyai fitur yaitu admin dapat melihat, menambah, mengubah, dan menghapus data ujian. Admin dapat menambahkan peserta atau siapa saja siswa yang mengikuti ujian. Setelah melakukan pengolahan data maka data yang baru akan disimpan di database. Activity diagram *admin* mengelola data ujian diperlihatkan pada gambar 4.9.



Gambar 4.9 Activity Diagram Admin Mengelola Data Ujian

g. *Activity* diagram *admin* mengelola data soal ujian

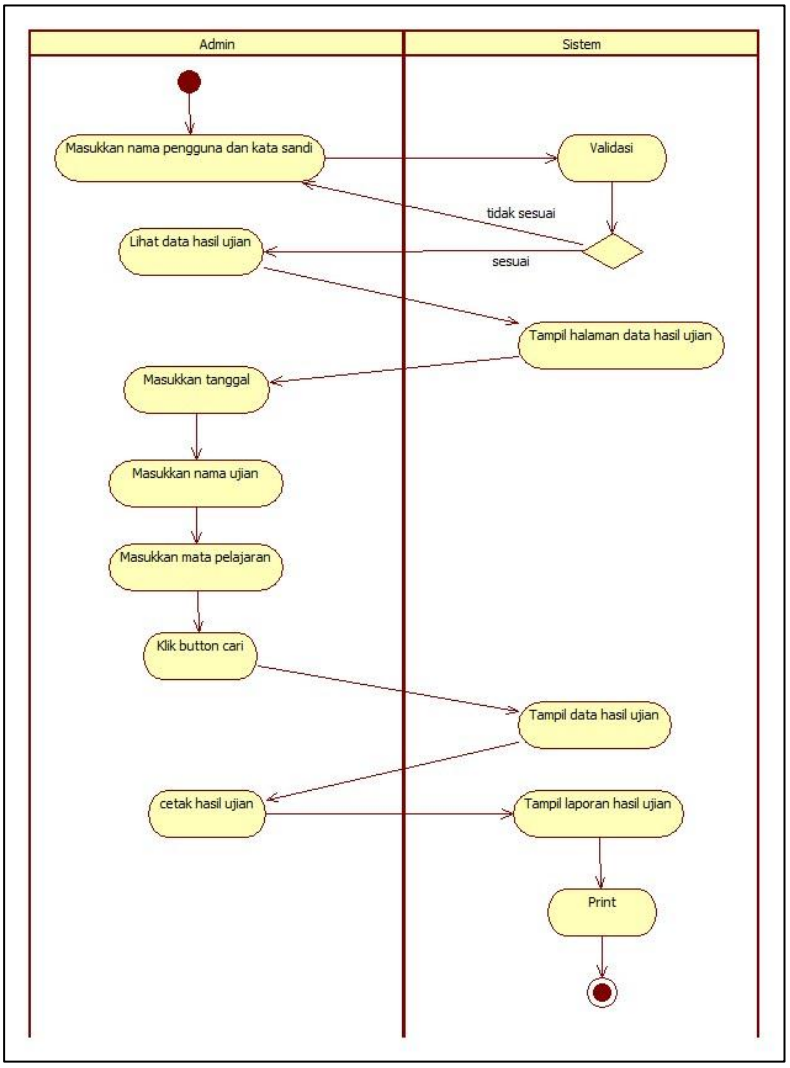
Dalam *activity* diagram *admin* mengelola data soal ujian mempunyai fitur yaitu admin dapat melihat, mengubah, menghapus, menambah data soal dan menambahkan data pertanyaan. *Activity* diagram *admin* mengelola data soal ujian diperlihatkan pada gambar 4.10.



Gambar 4.10 Activity Diagram Admin Meengelola Data Soal Ujian

h. *Activity* diagram *admin* mengelola data hasil ujian

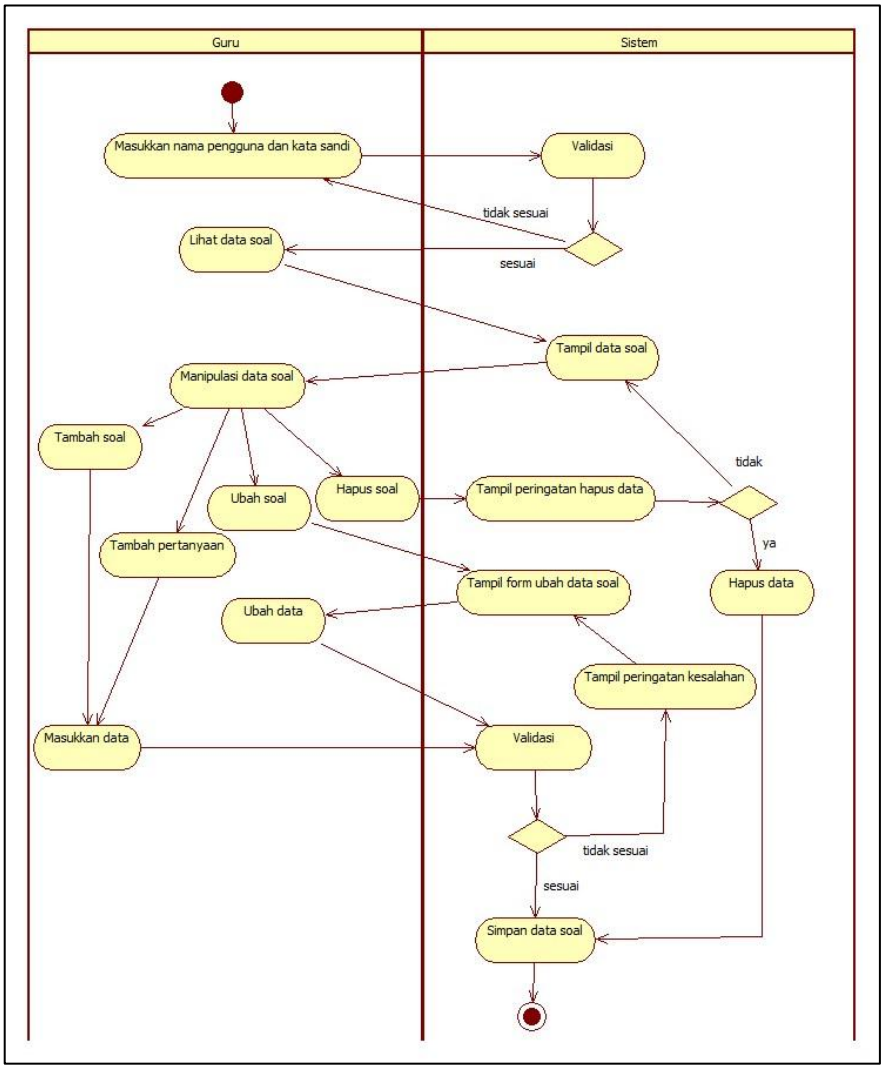
Dalam *activit*y diagram *admin* mengelola hasil ujian mempunyai fitur yaitu admin dapat hanya dapat melihat data hasil ujian berdasarkan kode ujian, nama mata pelajaran dan kelas yang diinputkan. Admin juga dapat mencetak hasil ujian. *Activity* diagram *admin* mengelola data siswa diperlihatkan pada gambar 4.11.



Gambar 4.11 Activity Diagram Admin Mengelola Data Hasil Ujian

i. *Activity* diagram guru mengelola data soal ujian

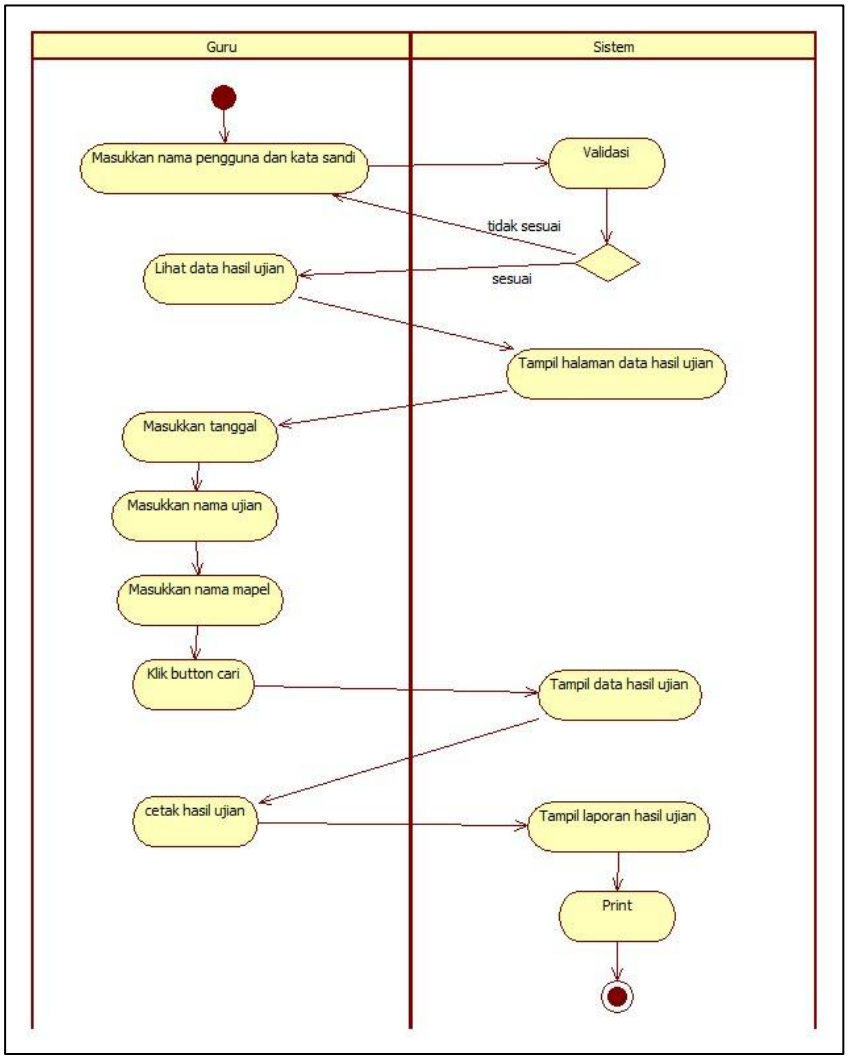
Dalam *activity* diagram guru mengelola data soal ujian mempunyai fitur sama dengan admin yaitu guru dapat melihat, mengubah, menghapus, dan menambah data soal ujian berdasarkan tahun ajaran, nama ujian, dan nama mata pelajaran yang diinputkan. *Activity* diagram guru mengelola data siswa diperlihatkan pada gambar 4.12.



Gambar 4.12 Activity Diagram Guru Mengelola Data Soal Ujian

j. *Activity* diagram guru mengelola data hasil ujian

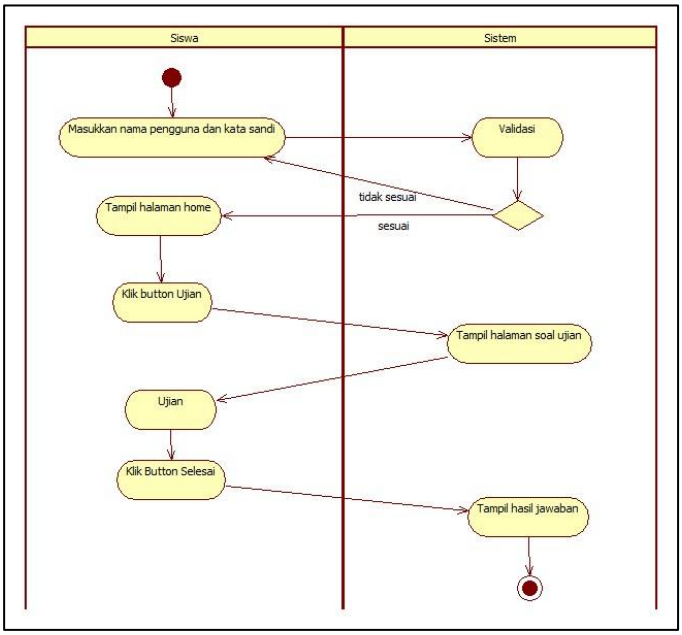
Dalam *activity* diagram guru mengelola hasil ujian mempunyai fitur yaitu guru dapat hanya dapat melihat dan menghapus data hasil ujian berdasarkan kode ujian, nama mata pelajaran dan kelas yang diinputkan. Admin juga dapat mencetak hasil ujian. *Activity* diagram admin mengelola data siswa diperlihatkan pada gambar 4.13.



Gambar 4.13 Activity Diagram Guru Mengelola Data Hasil Ujian

k. *Activity* diagram siswa melakukan ujian

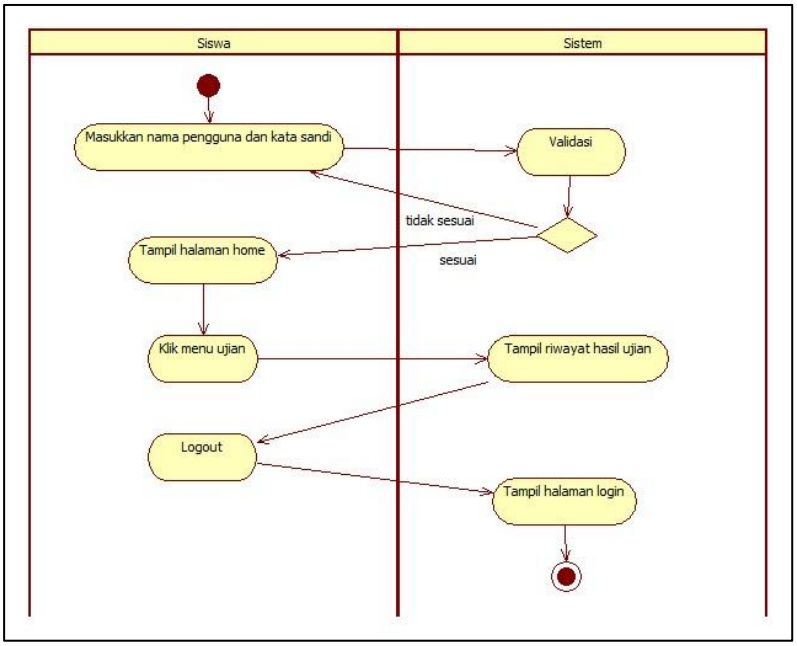
Dalam *activity* diagram siswa melakukan ujian mempunyai fitur yaitu siswa dapat mengerjakan soal-soal ujian yang meliputi *multiple choices single answer.* Siswa juga dapat melihat nilai atau hasil jawaban dari ujian secara langsung setelah mengklik button selesai. *Activity* diagram siswa melakukan ujian diperlihatkan pada gambar 4.14.



Gambar 4.14 Activity Diagram Siswa Melakukan Ujian

l. *Activity* diagram siswa melihat riwayat ujian

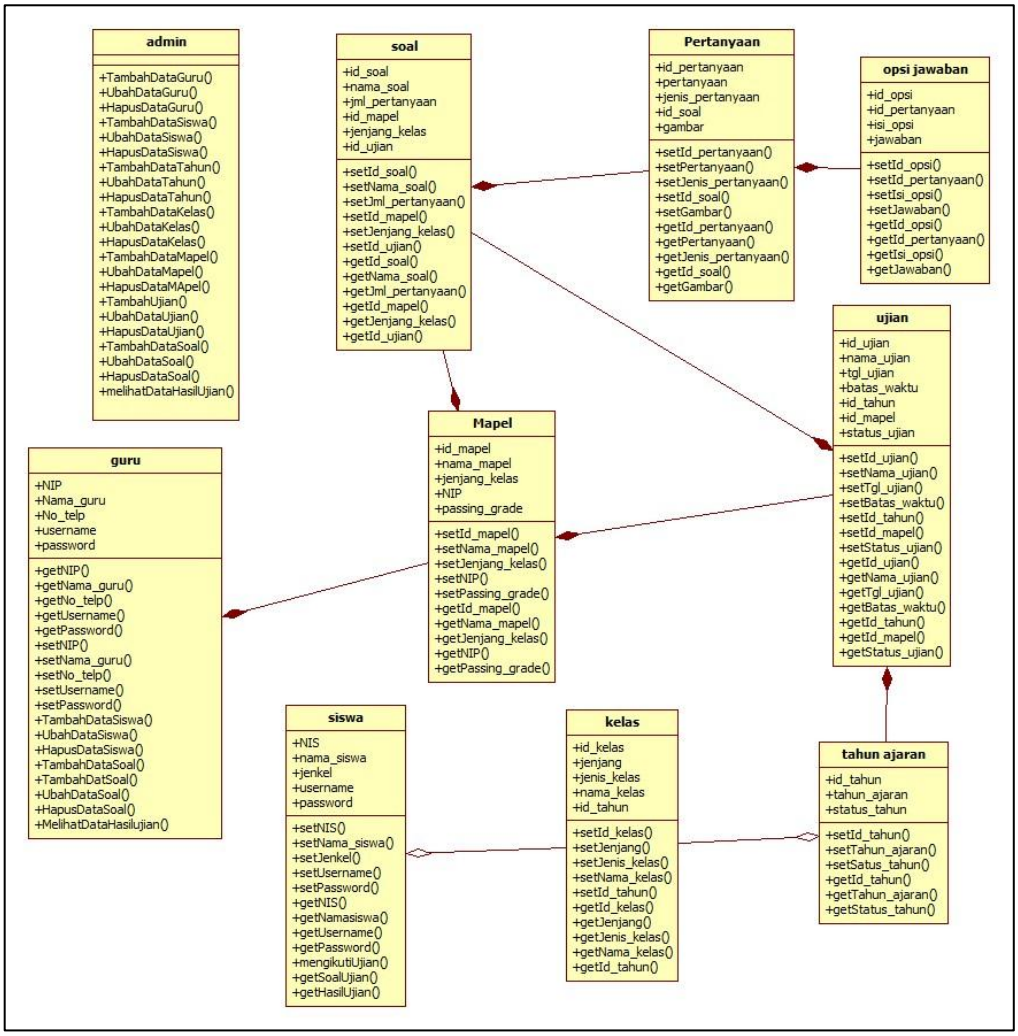
Dalam *activity* diagram siswa melihat riwayat ujian mempunyai fitur yaitu siswa dapat melihat semua nilai atau hasil jawaban ujian yang telah diikuti sebelumnya. *Activity* diagram siswa melihat riwayat ujian diperlihatkan pada gambar 4.15.



Gambar 4.15 Activity Diagram Siswa melihat Riwayat Ujian

3. *Class* Diagram

Class Diagram digunakan untuk mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terjadi. *Class* Diagram Sistem Informasi Ujian Online dapat dilihat pada Gambar 4.17.

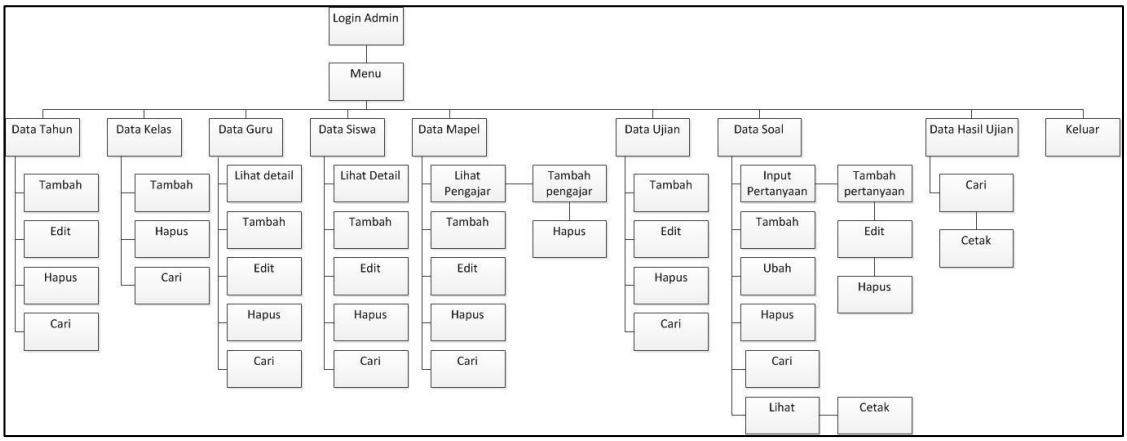


Gambar 4.17 Class Diagram

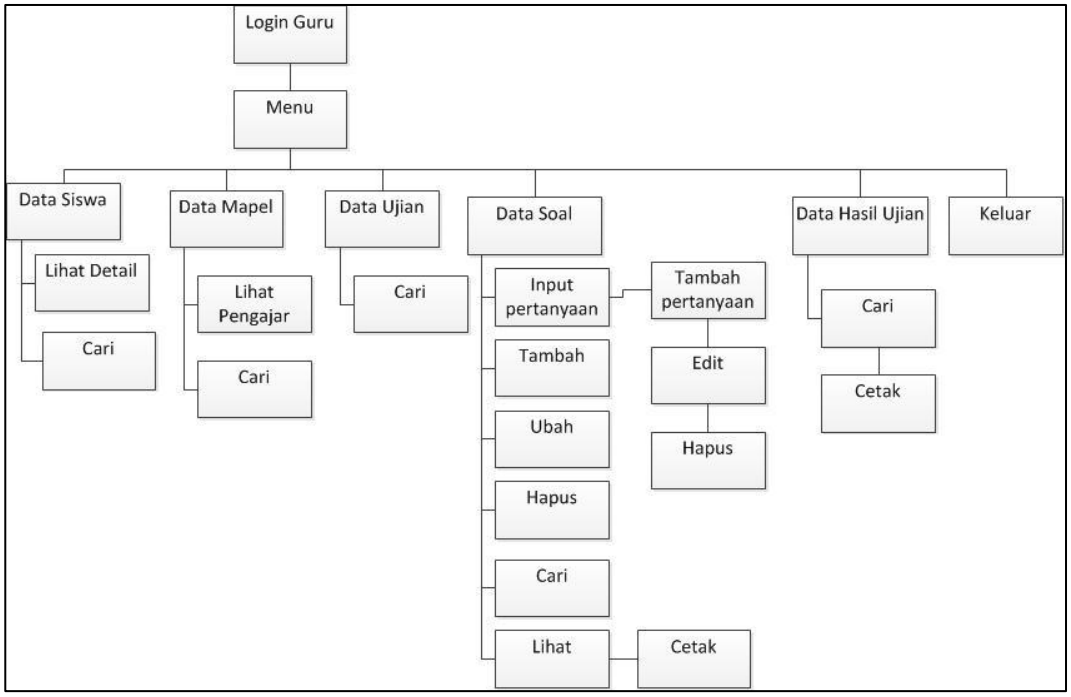
4.2.3 Perancangan Struktur Menu

Sistem Informasi Ujian Online memiliki tiga pengguna, yaitu admin, guru, dan siswa. Masing-Masing pengguna memiliki kebutuhan akan bisnis proses yang berbeda. Berdasarkan hal tersebut struktur menu untuk masing-masing pengguna dirancang sesuai kebutuhan masing-masing pengguna. Namun secara umum,

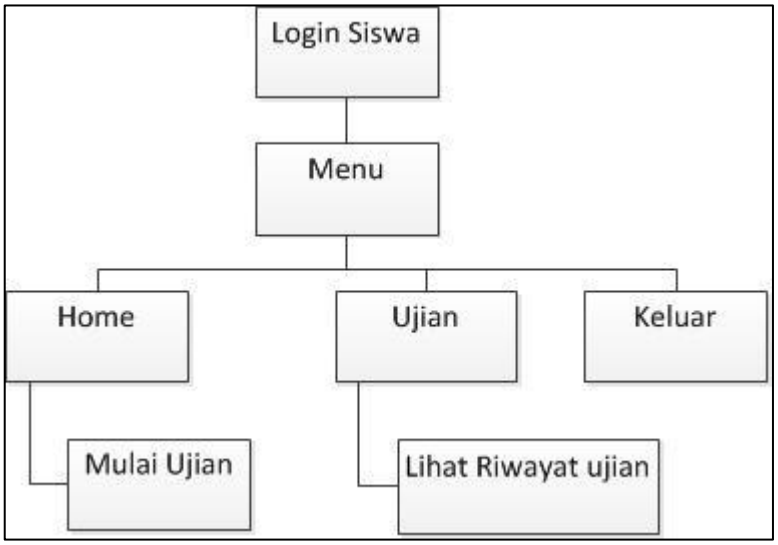
menu yang ada pada guru merupakan menu yang ada juga pada admin. Penjelasannya dapat dilihat pada Gambar 4.18, 4.19, dan 4.20.



Gambar 4.18 Rancangan struktur menu admin



Gambar 4.19 Rancangan struktur menu guru



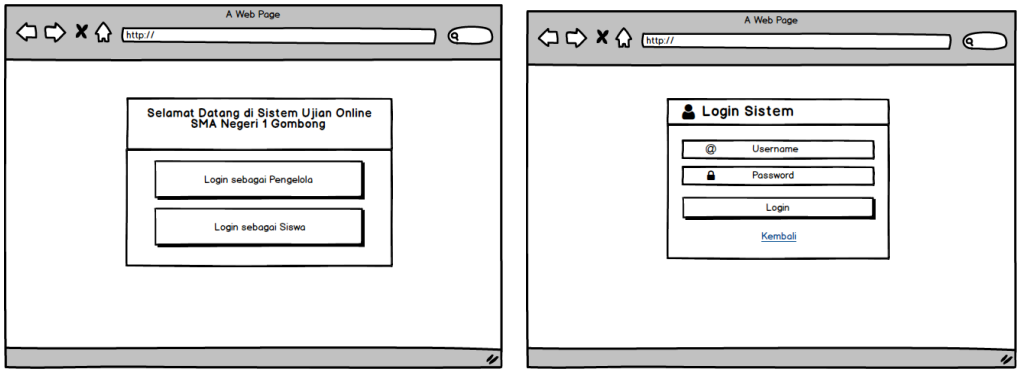
Gambar 4.20 Rancangan struktur menu siswa

4.2.4 Perancangan Antarmuka Pengguna

Perancangan antarmuka diperlukan dalam pembuatan sebuah sistem informasi. Antarmuka merupakan media yang akan menghubungkan interaksi antara sistem dengan pengguna. Rancangan antarmuka akan menunjukkan bagaimana komunikasi antara pengguna sistem dengan komputer. Komunikasi tersebut dapat berupa aktivitas memasukkan data ke sistem, menampilkan informasi atau meminta keluaran dari sistem.

1. Rancangan antarmuka halaman masuk / *login*

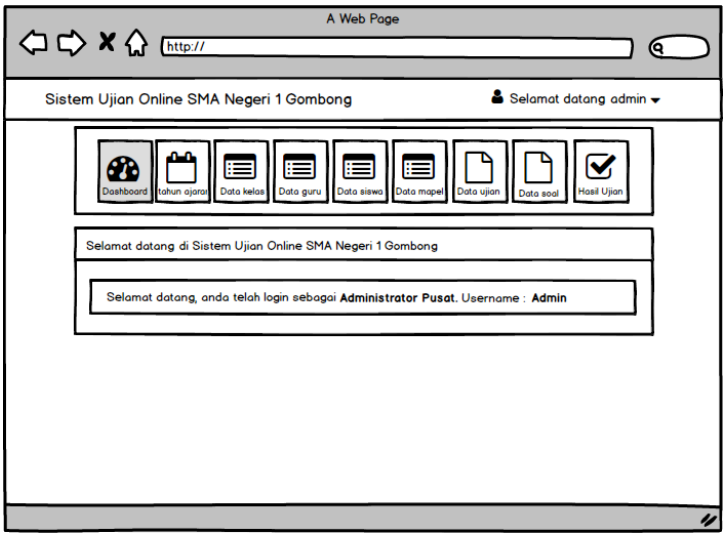
Pada halaman utama login terdapat 2 button yang terdiri dari button untuk login sebagai pengelola dan sebagai siswa. Setelah pengguna memilih salah satu button, kemudian pengguna akan dialihkan menuju halaman login pengelola atau login siswa. Pada halaman login pengelola ataupun siswa terdapat username, password, tombol login, dan text link kembali. Rancangan antarmuka halaman masuk / *login* dapat dilihat pada Gambar 4.21.



Gambar 4.21 Rancangan Antarmuka Halaman Login

2. Rancangan antarmuka halaman dashboard (utama)

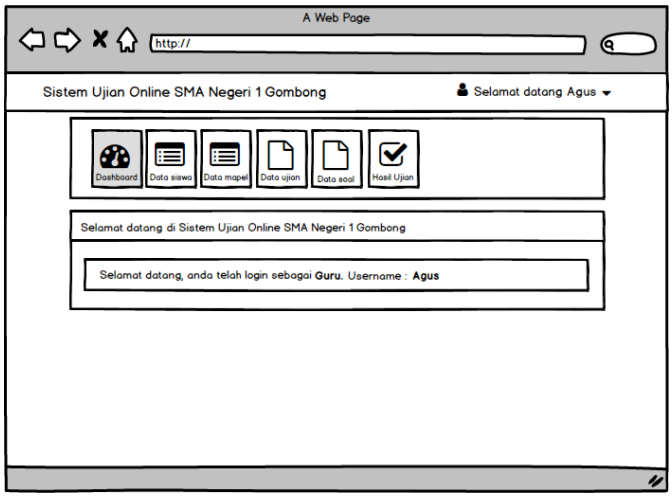
Halaman utama (*dashboard)* merupakan halaman yang akan ditampilkan pertama kali setelah pengguna berhasil melakukan *login* pada halaman *login*. Pada dasarnya, halaman utama memiliki tampilan yang sama, tetapi memiliki perbedaan pada menu-menu yang terdapat pada masing-masing hak akses. Pada halaman utama admin menampilkan sembilan menu pada bagian sidebar dan content selamat datang. Rancangan antarmuka halaman utama admin dapat dilihat pada Gambar 4.22.



Gambar 4.22 Rancangan antarmuka Halaman Utama Admin

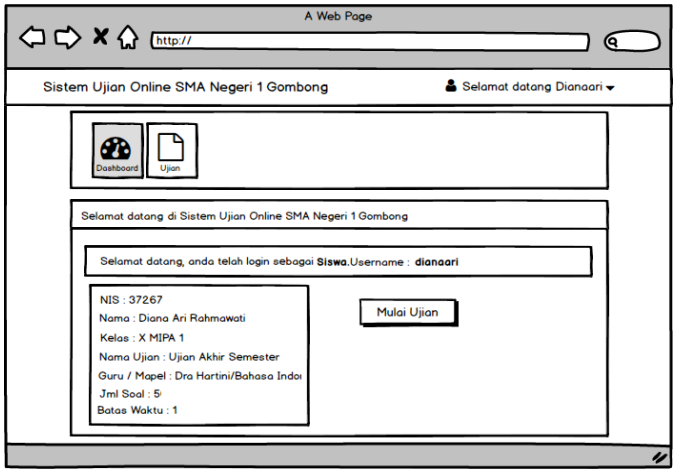
Sedangkan rancangan halaman utama guru sebenarnya hampir sama dengan halaman utama admin. Perbedaannya adalah menu pada bagian sidebar

dan dibagian content yang menampilkan *username* guru. Pada menu halaman utama guru terdapat enam menu yaitu *dashboard*, data siswa, data mapel, data ujian, data soal, dan hasil ujian. Rancangan antarmuka halaman utama guru dapat dilihat pada Gambar 4.23.



Gambar 4.23 Rancangan Antarmuka Halaman Utama Guru

Rancangan halaman utama siswa sedikit berbeda dengan halaman utama admin dan guru. Perbedaan pada halaman ini adalah menu pada bagian *sidebar* yang terdiri dari dua menu dan isi *content*. Rancangan antarmuka halaman utama siswa dapat dilihat pada Gambar ...

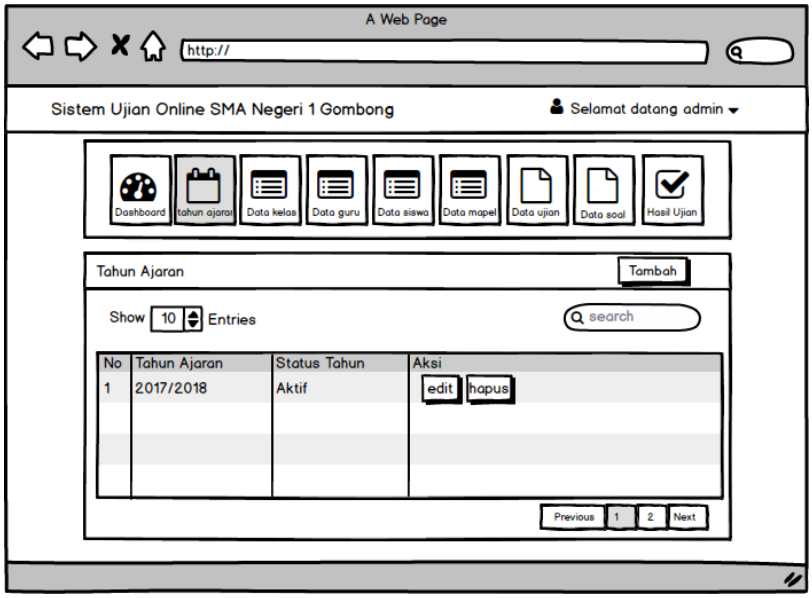


Gambar 4.24 Rancangan Antarmuka Halaman Utama Siswa

3. Rancangan antarmuka halaman tahun ajaran

1. Data tahun ajaran

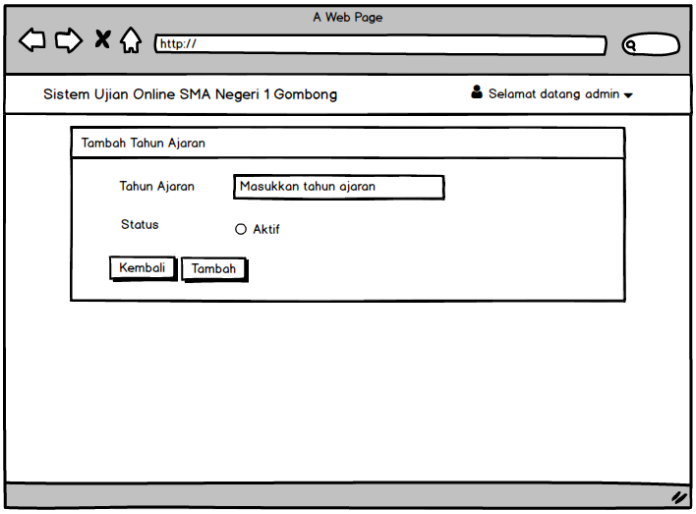
Rancangan antarmuka halaman tahun ajaran merupakan halaman yang menampilkan data tahun ajaran beserta status tahun ajaran yang meliputi kode tahun, tahun ajaran, dan status tahun. Halaman ini dirancang untuk menambah, mengubah, dan menghapus data tahun ajaran. Terdapat fitur search untuk mencari data berdasarkan *keyword* yang dimasukkan. Rancangan antarmuka halaman tahun ajaran oleh *admin* dapat dilihat pada Gambar 4.25.



Gambar 4.25 Rancangan Antarmuka Halaman Data Tahun Ajaran

2. Tambah Tahun Ajaran

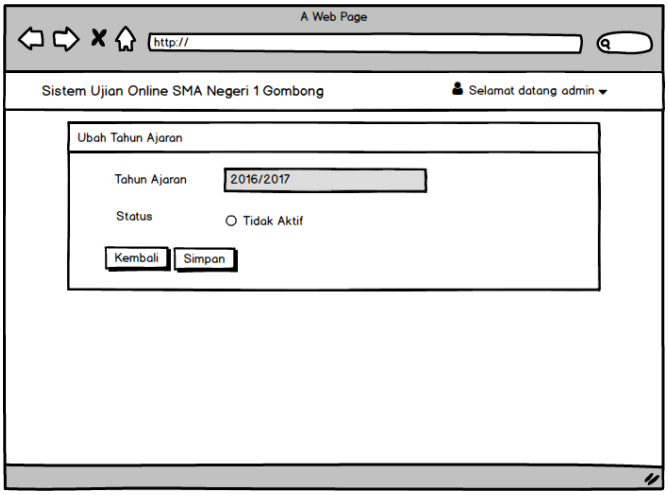
Halaman tambah tahun ajaran digunakan untuk menampilkan form yang wajib diinputkan oleh admin ketika memasukkan tahun ajaran baru. Rancangan antarmuka halaman tambah tahun ajaran dapat dilihat pada Gambar 4.26.



Gambar 4.26 Rancangan Antarmuka Halaman Tambah Tahun Ajaran

3. Ubah Tahun Ajaran

Halaman ubah tahun ajaran merupakan halaman yang berisi form yang digunakan untuk mengubah status tahun ajaran berdasarkan id tahun. Rancangan antarmuka halaman ubah tahun ajaran dapat dilihat pada Gambar 4.27.



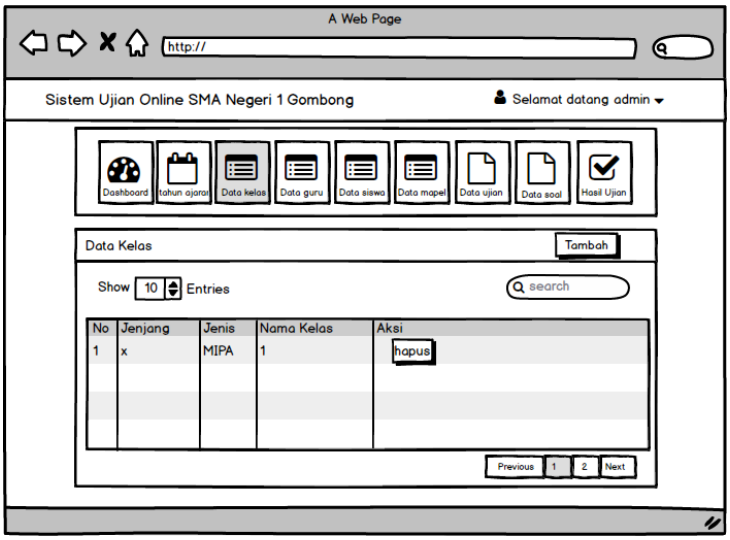
Gambar 4.27 Rancangan Antarmuka Halaman Ubah Tahun Ajaran

4. Rancangan antarmuka halaman data kelas

1. Data Kelas

Rancangan antarmuka halaman data kelas merupakan halaman yang menampilkan data kelas di SMA Negeri 1 Gombong yang meliputi jenjang, jenis

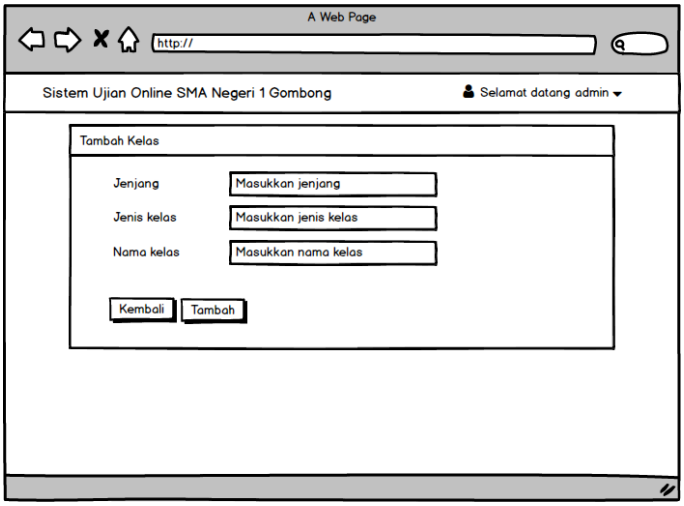
kelas, dan nama kelas. Halaman ini dirancang menambah dan menghapus data kelas. Terdapat fitur search untuk mencari data berdasarkan *keyword* yang dimasukkan. Rancangan antarmuka halaman data kelas oleh *admin* dapat dilihat pada Gambar 4.28.



Gambar 4.28 Rancangan Antarmuka Halaman Data Kelas

2. Tambah Kelas

Halaman tambah kelas digunakan untuk menampilkan form yang wajib diinputkan oleh admin ketika memasukkan data kelas baru. Rancangan antarmuka halaman tambah kelas dapat dilihat pada Gambar 4.29.

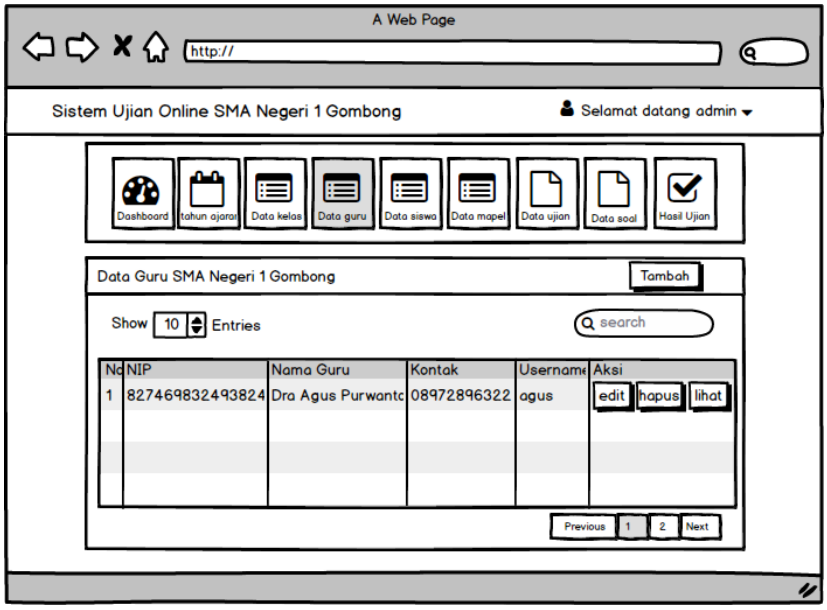


Gambar 4.29 Rancangan Antarmuka Halaman Tambah Kelas

5. Rancangan antarmuka halaman data guru

1. Data Guru

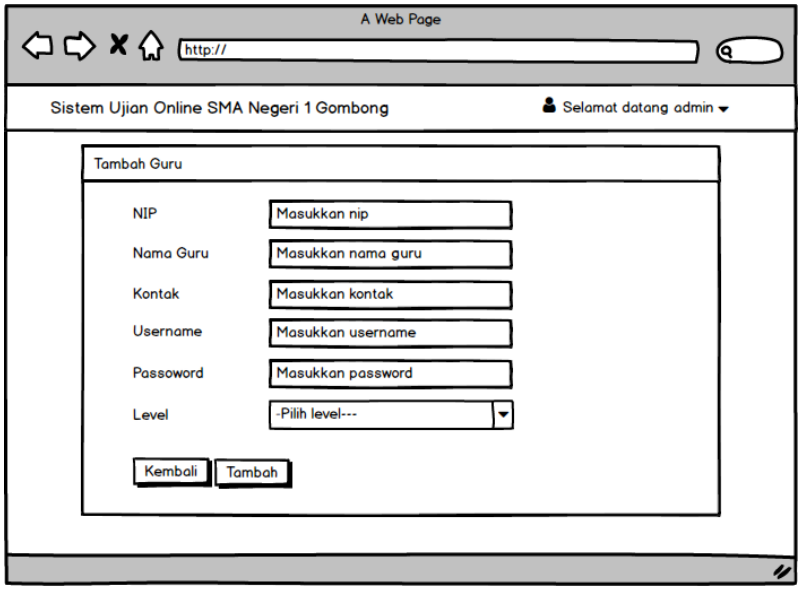
Rancangan antarmuka halaman data guru merupakan halaman awal yang ditampilkan setelah admin *login* ke dalam sistem. Disini akan ditampilkan data tabel guru SMA Negeri 1 Gombong yang meliputi NIP, nama guru, kontak, dan username. Halaman ini dirancang untuk menambah, mengubah, dan menghapus data guru. Terdapat fitur search untuk mencari data berdasarkan *keyword* yang dimasukkan. Rancangan antarmuka halaman data guru dapat dilihat pada Gambar 4.30.



Gambar 4.30 Rancangan Antarmuka Halaman Data Guru

2. Tambah Guru

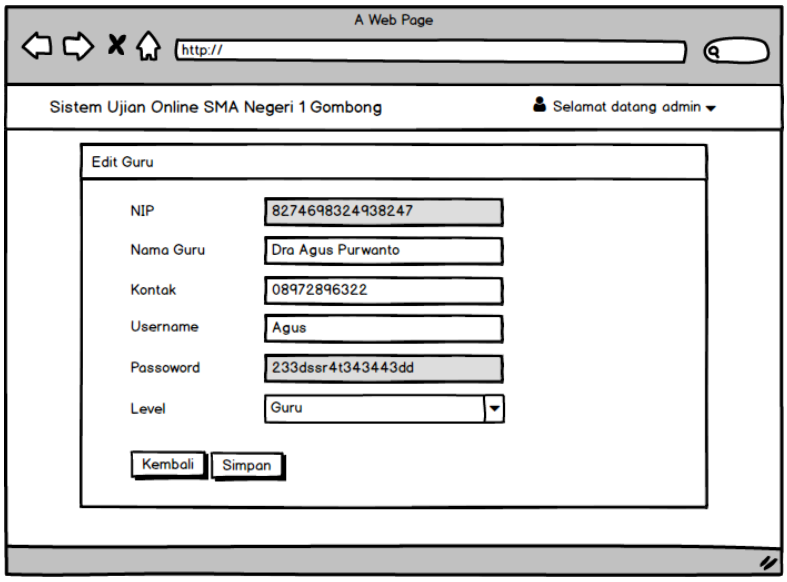
Halaman tambah guru digunakan untuk menampilkan form yang wajib diinputkan oleh admin ketika memasukkan guru baru. Rancangan antarmuka halaman tambah guru dapat dilihat pada Gambar 4.31.



Gambar 4.31 Rancangan Antarmuka Halaman Tambah Guru

3. Ubah Guru

Halaman Ubah Guru merupakan halaman yang berisi form yang digunakan untuk mengubah data guru sebelumnya berdasarkan nip. Rancangan antarmuka halaman ubah guru dapat dilihat pada Gambar 4.32.

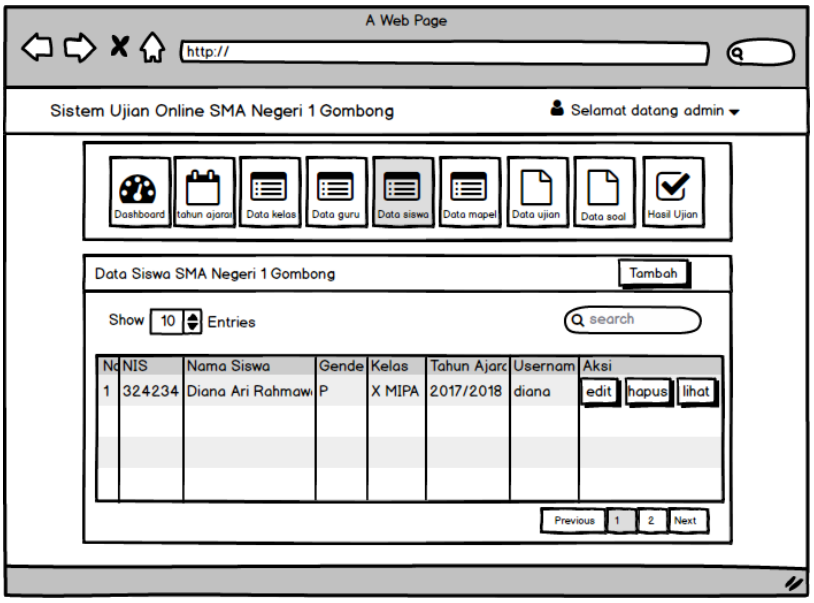


Gambar 4.32 Rancangan Antarmuka Halaman Ubah Guru

6. Rancangan antarmuka halaman data siswa

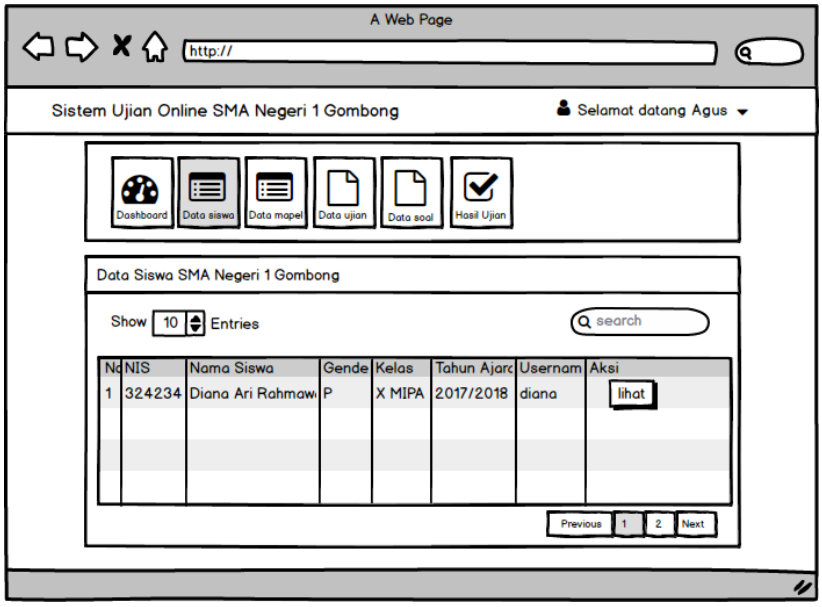
1. Data Siswa

Halaman data siswa merupakan halaman yang menampilkan data tabel siswa SMA Negeri 1 Gombong yang meliputi NIS, nama siswa, gender, kelas, tahun ajaran, dan username. Halaman ini dirancang untuk menambah, mengubah, dan menghapus data siswa. Terdapat fitur search untuk mencari data berdasarkan *keyword* yang dimasukkan. Rancangan antarmuka halaman data siswa dapat dilihat pada Gambar 4.33.



Gambar 4.33 Rancangan Antarmuka Halaman Data Siswa oleh Admin

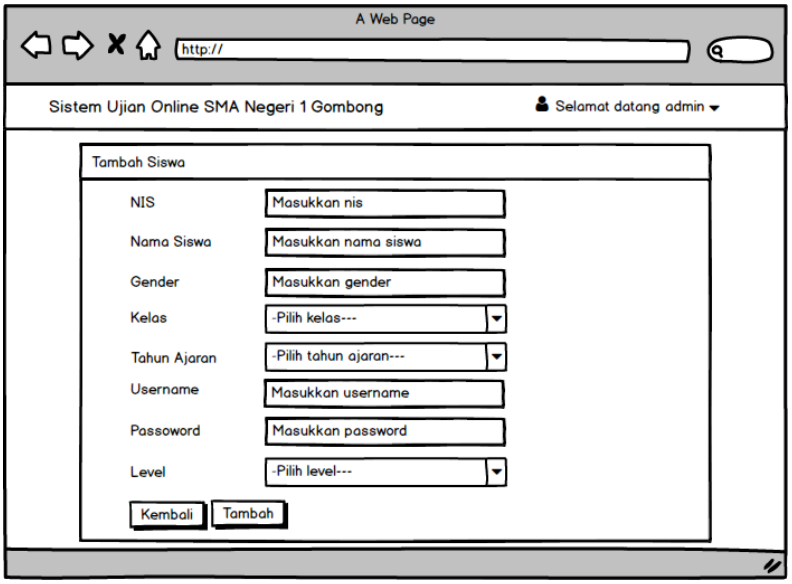
Sedangkan rancangan antarmuka halaman data siswa oleh guru hanya dapat melihat detail informasi data siswa. Rancangan antarmuka halaman data siswa oleh guru dapat dilihat pada Gambar 4.34.



Gambar 4.34 Rancangan Antarmuka Halaman Data Siswa oleh Guru

2. Tambah Siswa

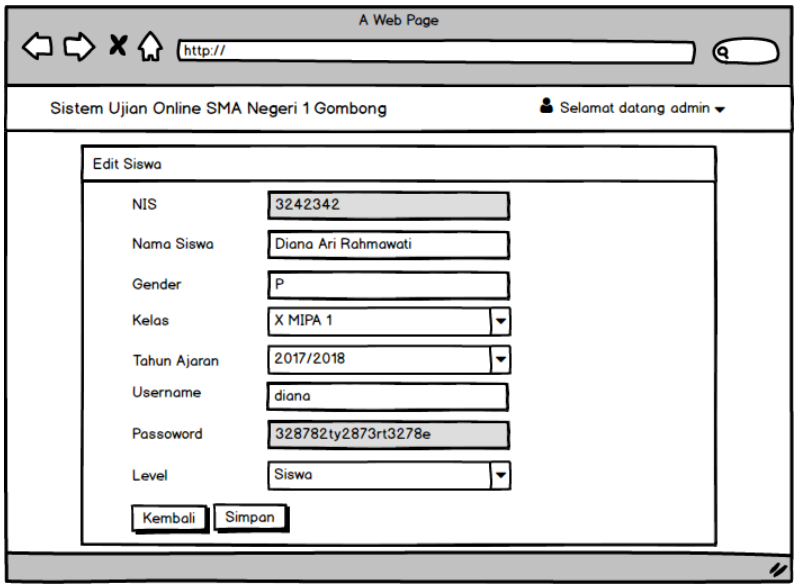
Halaman tambah siswa digunakan untuk menampilkan form yang wajib diinputkan oleh admin ketika memasukkan siswa baru. Rancangan antarmuka halaman tambah siswa dapat dilihat pada Gambar 4.35.



Gambar 4.35 Rancangan Antarmuka Halaman Tambah Siswa

3. Ubah Siswa

Halaman Ubah Siswa merupakan halaman yang berisi form yang digunakan untuk mengubah data siswa sebelumnya berdasarkan nis. Rancangan antarmuka halaman ubah siswa dapat dilihat pada Gambar 4.36.

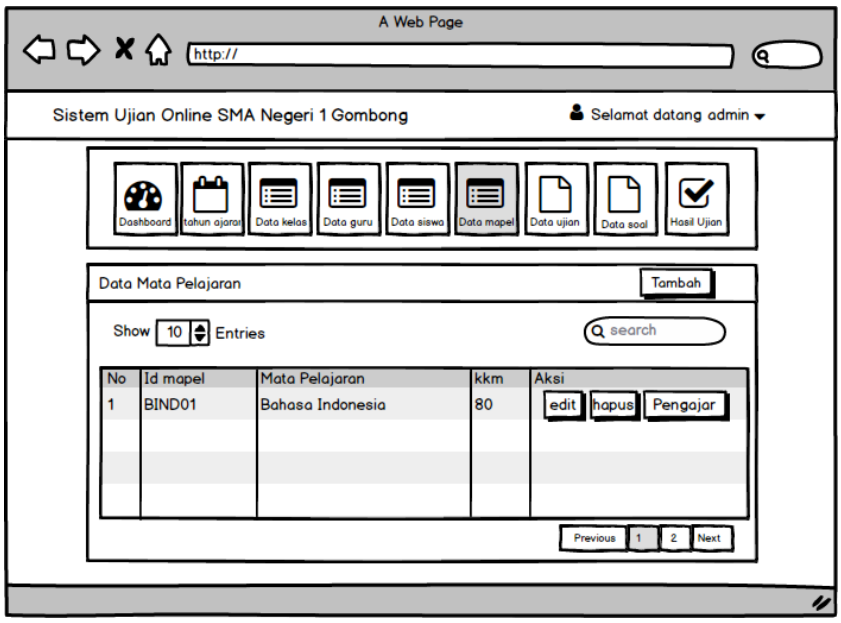


Gambar 4.36 Rancangan Antarmuka Halaman Ubah Siswa

7. Rancangan antarmuka halaman data mapel

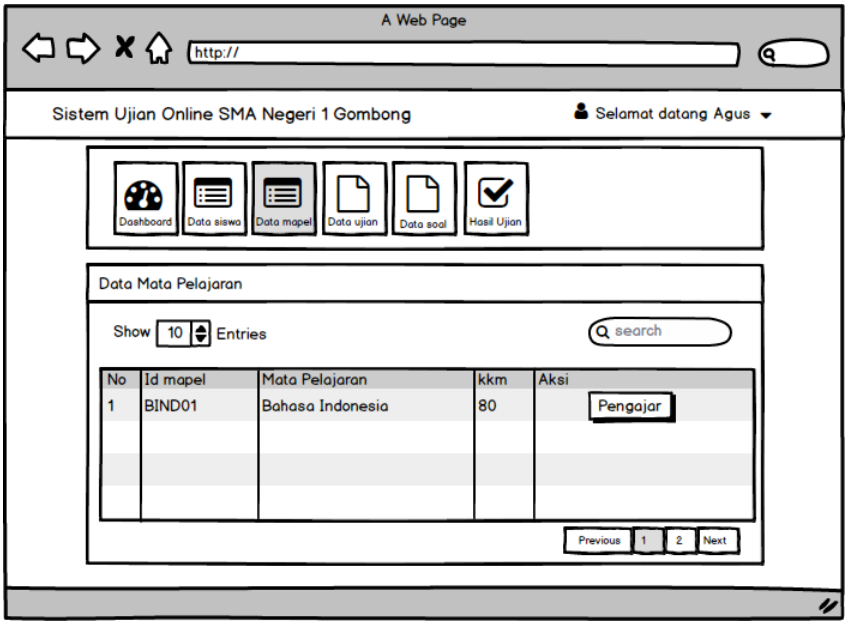
1. Data Mapel

Rancangan antarmuka halaman data mapel merupakan halaman yang menampilkan data tabel mata pelajaran yang ada di SMA Negeri 1 Gombong yang meliputi kode mapel, nama mapel, dan kkm. Halaman ini dirancang untuk menambah, mengubah, menghapus data mapel serta melihat, menambah, dan menghapus pengajar. Rancangan antarmuka halaman data mapel untuk *admin* dapat dilihat pada Gambar 4.37.



Gambar 4.37 Rancangan Antarmuka Halaman Data Mapel untuk Admin

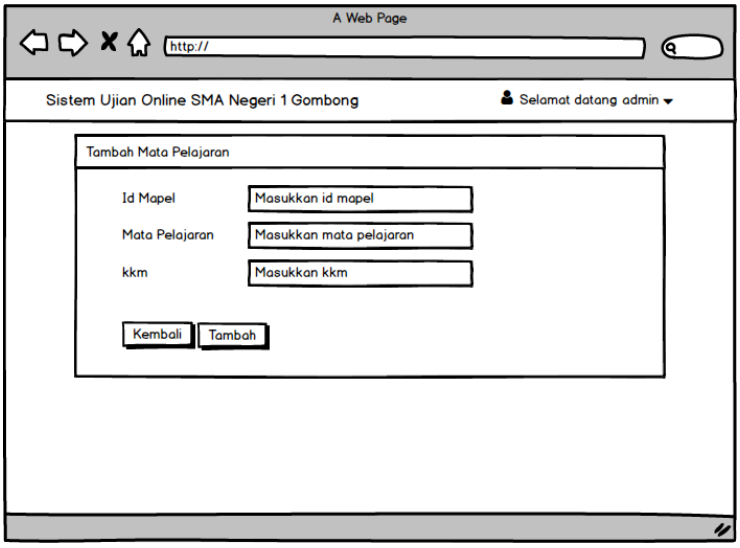
Sedangkan rancangan antarmuka halaman data mapel oleh guru hanya dapat melihat detail informasi pengajar. Rancangan antarmuka halaman data mapel untuk guru dapat dilihat pada Gambar 4.38.



Gambar 4.38 Rancangan Antarmuka Halaman Data Mapel untuk Guru

2. Tambah Mapel

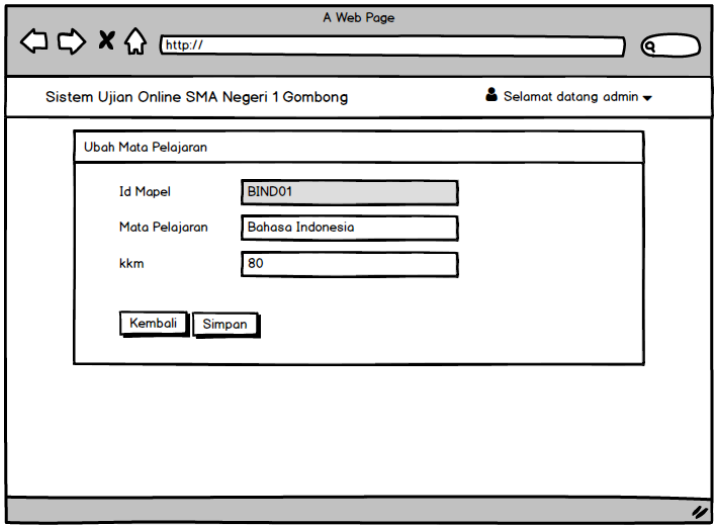
Halaman tambah mapel digunakan untuk menampilkan form yang wajib diinputkan oleh admin ketika memasukkan mata pelajaran baru. Rancangan antarmuka halaman tambah mapel dapat dilihat pada Gambar 4.39.



Gambar 4.39 Rancangan Antarmuka Halaman Tambah Mapel

3. Ubah Mapel

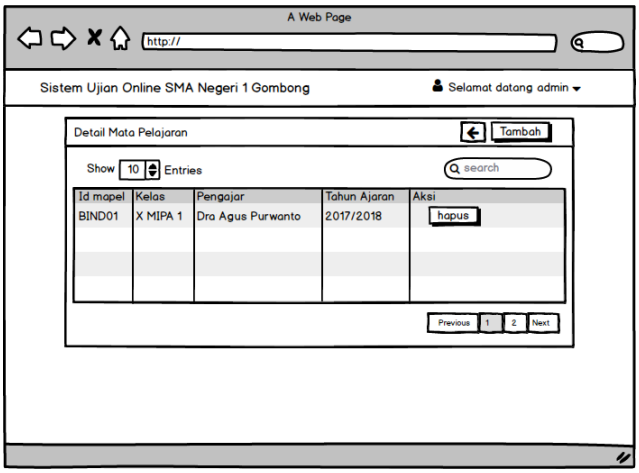
Halaman Ubah Mapel merupakan halaman yang berisi form yang digunakan untuk mengubah data mapel sebelumnya berdasarkan id mapel. Rancangan antarmuka halaman ubah mapel dapat dilihat pada Gambar 4.40.



Gambar 4.40 Rancangan Antarmuka Halaman Ubah Mapel

4. Lihat Pengajar

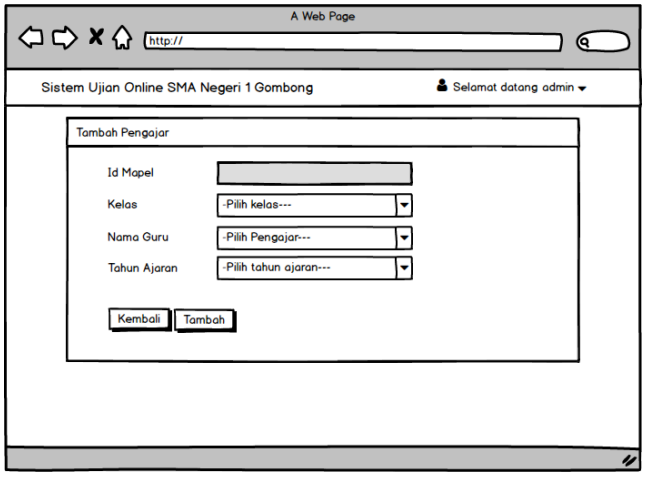
Halaman lihat pengajar merupakan halaman yang menampilkan data tabel guru pengajar mata pelajaran yang ada di SMA Negeri 1 Gombong yang meliputi id mapel, kelas, dan tahun ajaran. Halaman ini dirancang untuk menambah dan menghapus pengajar. Rancangan antarmuka halaman data pengajar oleh admin dapat dilihat pada Gambar 4.41.



Gambar 4.41 Rancangan Antarmuka Halaman Lihat Pengajar

5. Tambah Pengajar

Halaman tambah pengajar digunakan untuk menampilkan form yang wajib diinputkan oleh *admin* ketika memasukkan guru pengajar mapel baru. Rancangan antarmuka halaman tambah pengajar dapat dilihat pada Gambar 4.42.

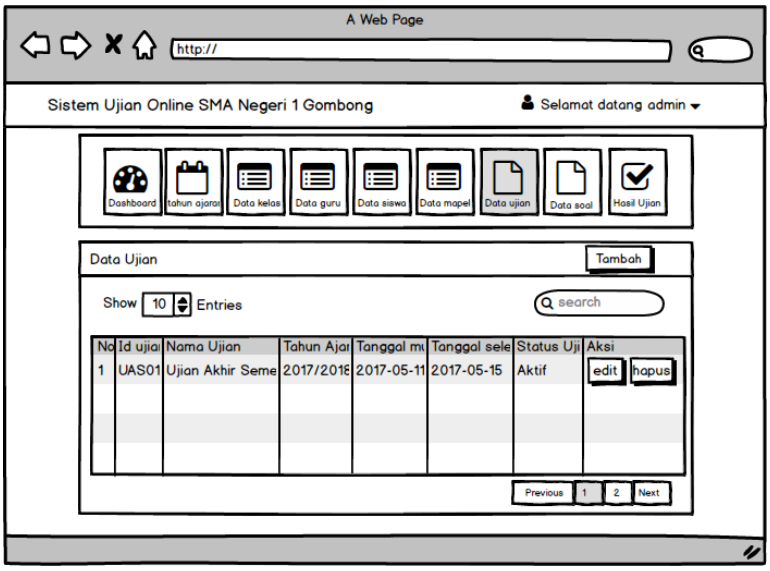


Gambar 4.42 Rancangan Antarmuka Halaman Tambah Pengajar

8. Rancangan antarmuka halaman data ujian

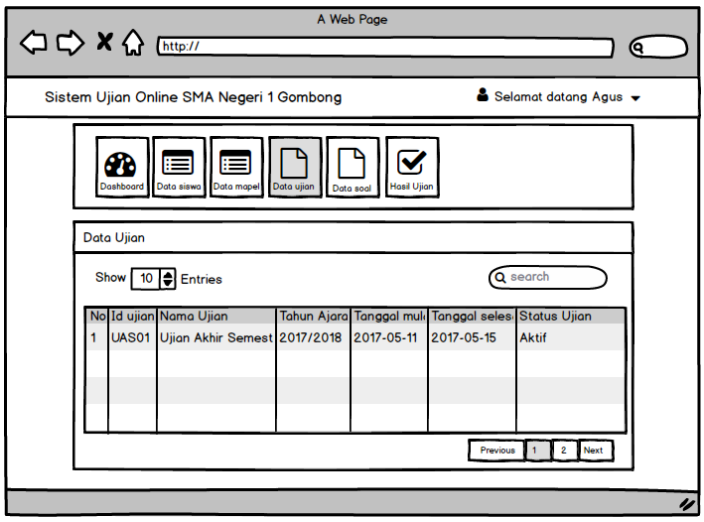
1. Data Ujian

Rancangan antarmuka halaman data ujian merupakan halaman yang menampilkan data ujian yang akan digunakan untuk pelaksanaan proses ujian online yang meliputi id ujian, nama ujian, tahun ajaran, tanggal mulai, tanggal selesai, dan status ujian. Halaman ini dirancang untuk menambah, mengubah, dan menghapus data ujian. Rancangan antarmuka halaman data ujian untuk *admin* dapat dilihat pada Gambar 4.43.



Gambar 4.43 Rancangan Antarmuka Halaman Data Ujian untuk Admin

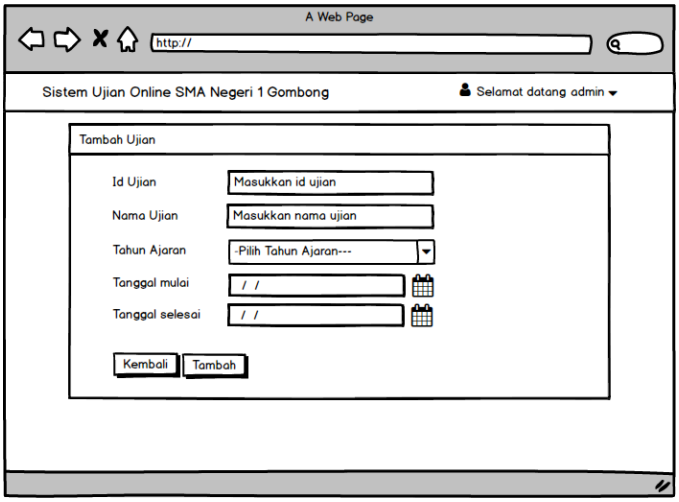
Sedangkan rancangan antarmuka halaman data ujian oleh guru hanya dapat melihat data ujian. Rancangan antarmuka halaman data ujian untuk guru dapat dilihat pada Gambar 4.44.



Gambar 4.44 Rancangan Antarmuka Halaman Data Ujian untuk Guru

2. Tambah Ujian

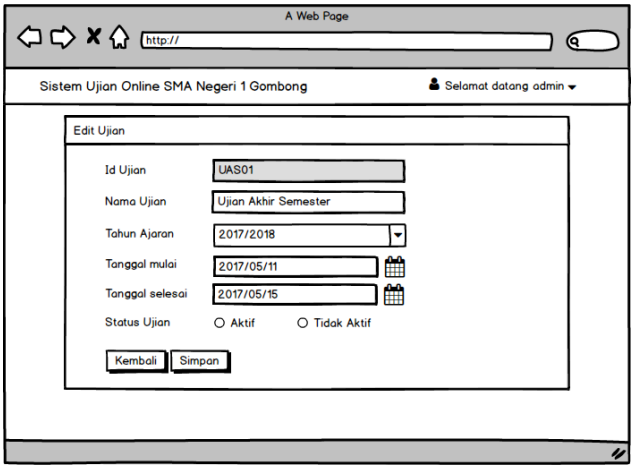
Halaman tambah ujian digunakan untuk menampilkan form yang wajib diinputkan oleh admin ketika memasukkan data ujian baru. Rancangan antarmuka halaman tambah ujian dapat dilihat pada Gambar 4.45.



Gambar 4.45 Rancangan Antarmuka Halaman Tambah Ujian

3. Ubah Ujian

Halaman Ubah Ujian merupakan halaman yang berisi form yang digunakan untuk mengubah data ujian sebelumnya berdasarkan id ujian. Rancangan antarmuka halaman ubah ujian dapat dilihat pada Gambar 4.46.

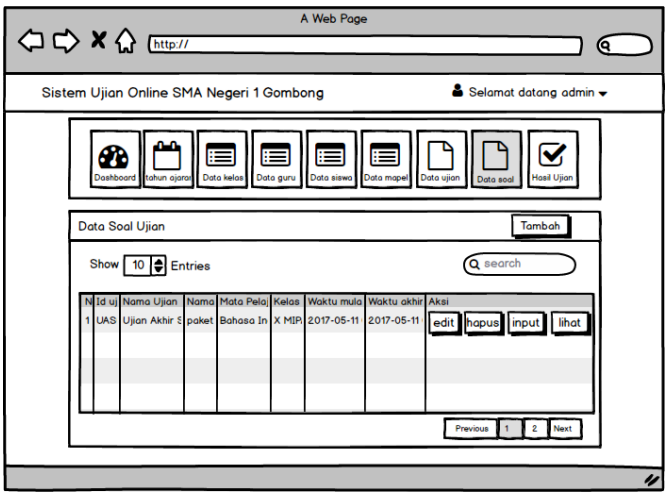


Gambar 4.46 Rancangan Antarmuka Halaman Ubah Ujian

9. Rancangan antarmuka halaman data soal ujian

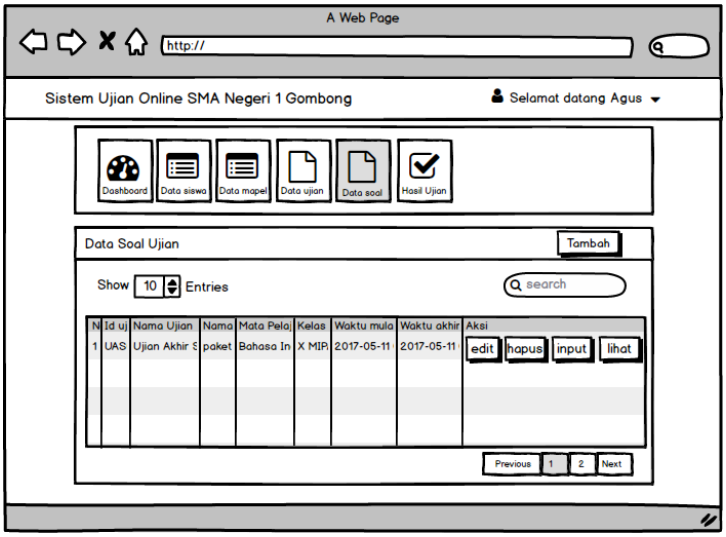
1. Data Soal

Rancangan antarmuka halaman data soal merupakan halaman yang menampilkan data soal untuk pelaksanaan ujian di SMA Negeri 1 Gombong yang meliputi id ujian, nama ujian, nama soal, mata pelajaran, kelas, waktu mulai, dan waktu akhir. Halaman ini dirancang untuk menambah, mengubah, dan menghapus data soal serta melihat daftar pertanyaan, tambah, ubah, dan hapus pertanyaan. Rancangan antarmuka halaman data soal untuk *admin* dapat dilihat pada Gambar 4.47.



Gambar 4.47 Rancangan Antarmuka Halaman Data Soal untuk Admin

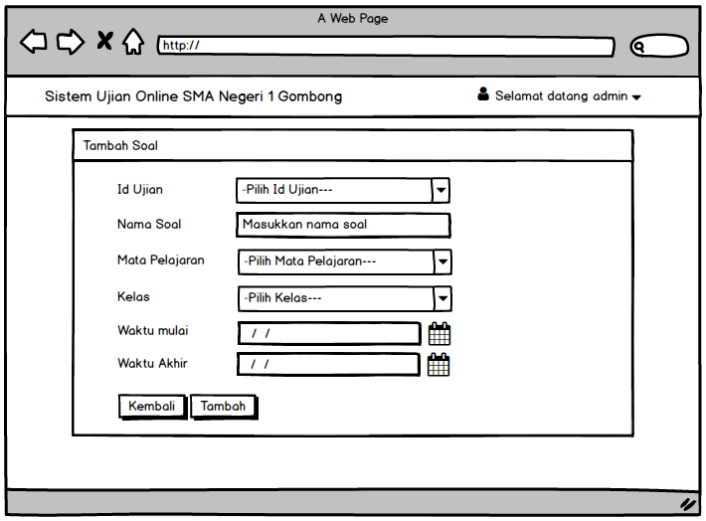
Sedangkan rancangan antarmuka halaman data soal oleh guru sama dengan halaman untuk admin. Rancangan antarmuka halaman data soal oleh guru dapat dilihat pada Gambar 4.48.



Gambar 4.48 Rancangan Antarmuka Halaman Data Soal untuk Guru

2. Tambah Soal

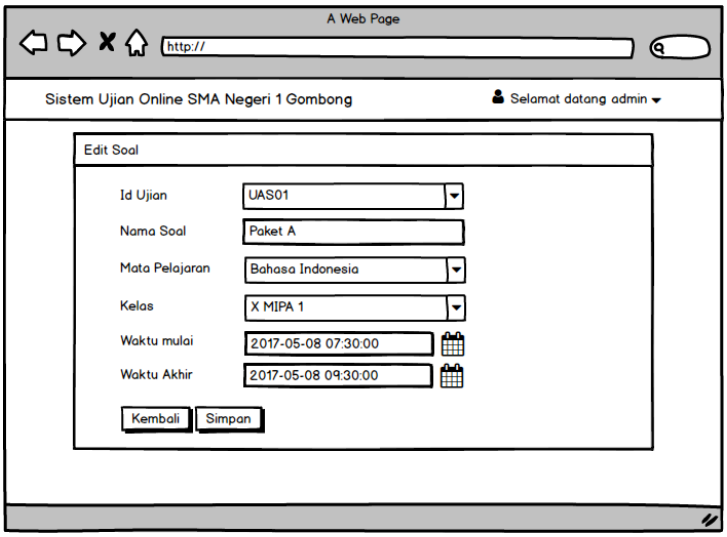
Halaman tambah soal digunakan untuk menampilkan form yang wajib diinputkan oleh admin ketika memasukkan data soal baru. Rancangan antarmuka halaman tambah soal dapat dilihat pada Gambar 4.49.



Gambar 4.49 Rancangan Antarmuka Halaman Tambah Soal

3. Ubah Soal

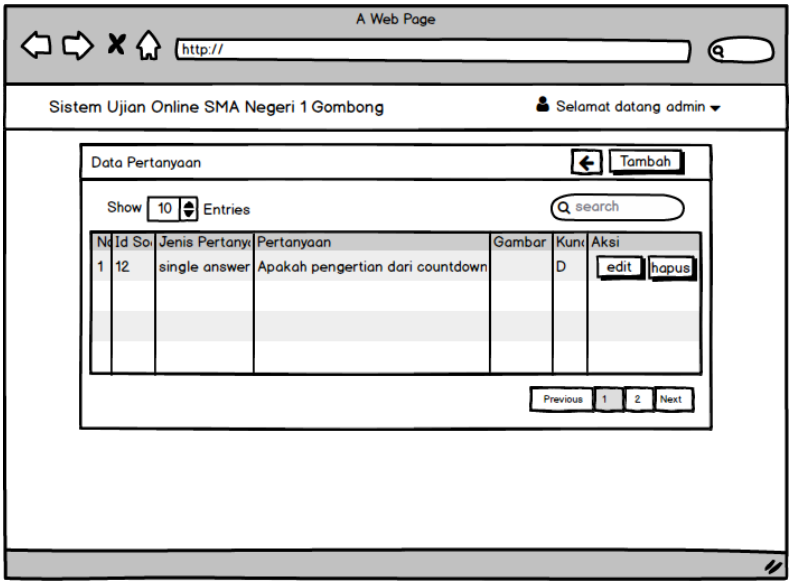
Halaman Ubah Soal merupakan halaman yang berisi form yang digunakan untuk mengubah data soal sebelumnya berdasarkan id soal. Rancangan antarmuka halaman ubah soal dapat dilihat pada Gambar 4.50.



Gambar 4.50 Rancangan Antarmuka Halaman Ubah Soal

4. Lihat Daftar Pertanyaan

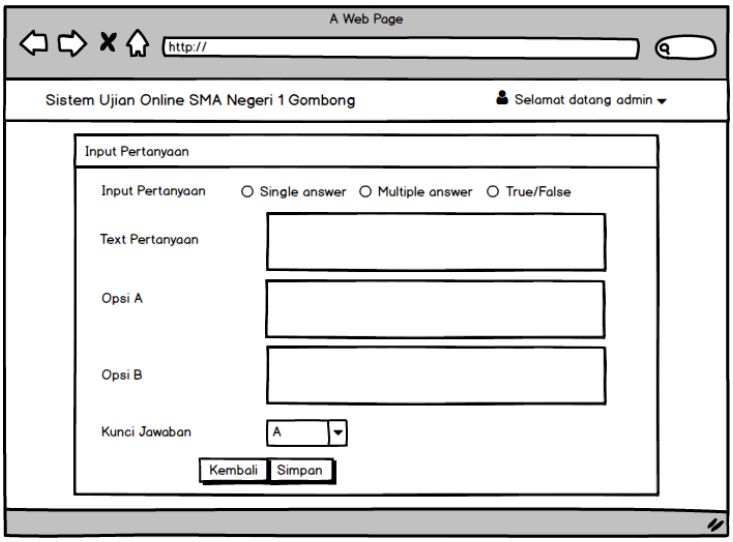
Halaman lihat daftar pertanyaan merupakan halaman yang menampilkan data tabel pertanyaan yang meliputi id soal, jenis pertanyaan, pertanyaan, gambar, dan kunci. Halaman ini dirancang untuk menambah, mengubah dan menghapus pertanyaan dari soal. Rancangan antarmuka halaman data pertanyaan dapat dilihat pada Gambar 4.51.



Gambar 4.51 Rancangan Antarmuka Halaman Lihat Daftar Pertanyaan

5. Tambah Pertanyaan

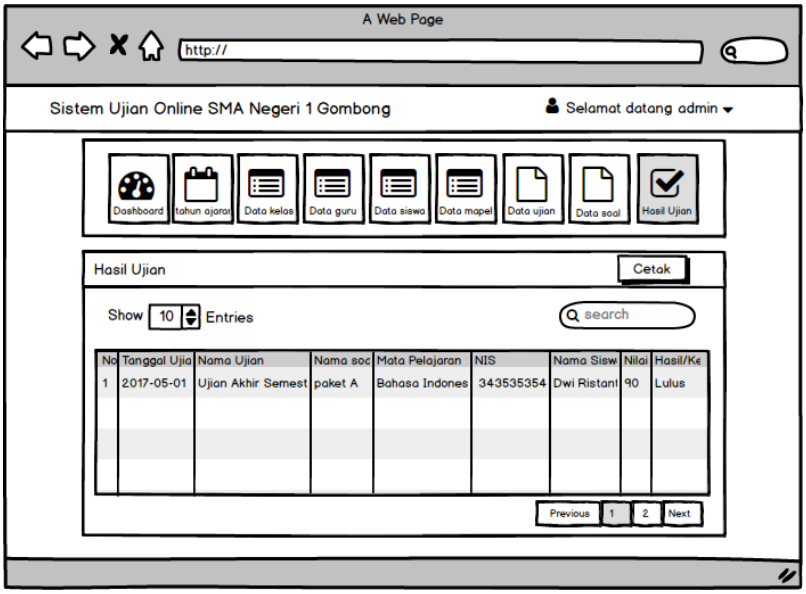
Halaman tambah pertanyaan digunakan untuk menampilkan form yang wajib diinputkan oleh admin ketika memasukkan data pertanyaan baru. Rancangan antarmuka halaman tambah pertanyaan dapat dilihat pada Gambar 4.52.



Gambar 4.52 Rancangan Antarmuka Halaman Tambah Pertanyaan

10. Rancangan antarmuka halaman data hasil ujian

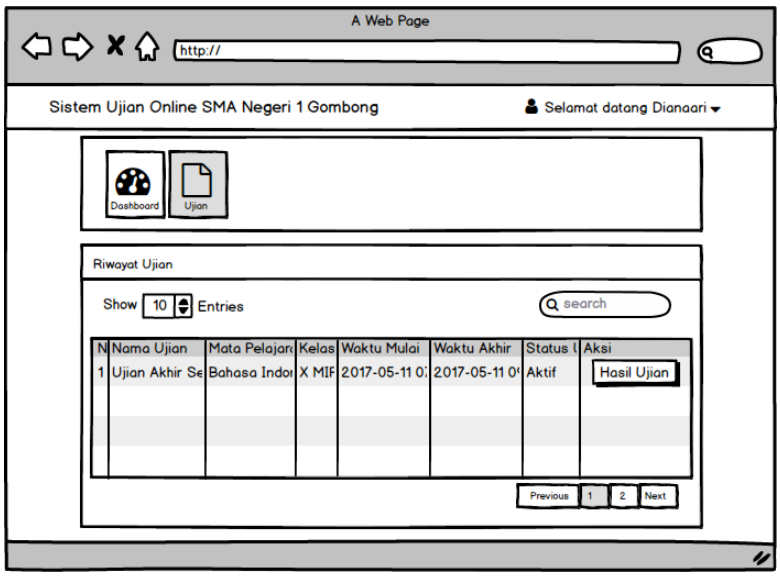
Rancangan antarmuka halaman data hasil ujian menurpakan halaman yang menampilkan data hasil ujian siswa yang sudah selesai melaksanakan ujian. Halaman ini terdiri dari data tabel dengan kolom tanggal ujian, nama ujian, nama soal, mata pelajaran, NIS, Nama Siswa, Nilai, Hasil/Ket. Halaman ini dirancang untuk melihat hasil ujian dan mencetak laporan hasil ujian. Rancangan antarmuka halaman data hasil ujian untuk *admin* dapat dilihat pada Gambar 4.53.



Gambar 4.53 Rancangan Antarmuka Halaman Data Hasil Ujian

11. Rancangan antarmuka halaman riwayat ujian siswa

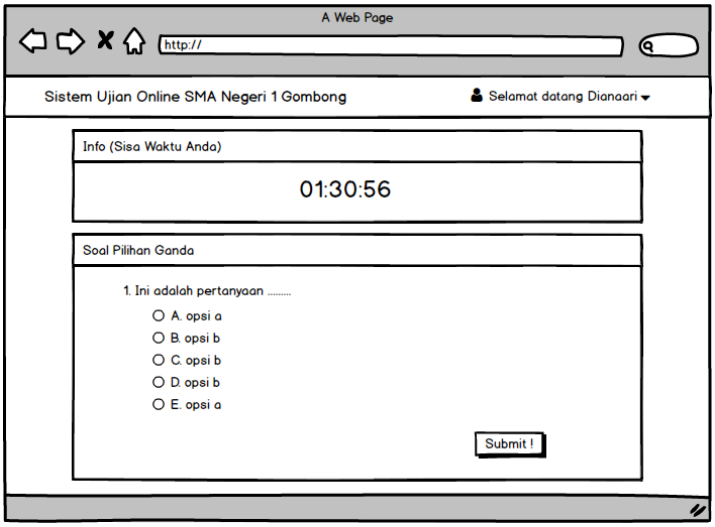
Rancangan antarmuka halaman riwayat ujian siswa merupakan halaman yang berisi data riwayat ujian yang sudah dilaksanakan. Siswa dapat melihat hasil dengan klik button hasil ujian pada kolom aksi. Rancangan antarmuka halaman riwayat ujian siswa dapat dilihat pada Gambar 4.54.



Gambar 4.54 Rancangan Antarmuka Halaman Riwayat Ujian Siswa

12. Rancangan antarmuka halaman soal ujian siswa

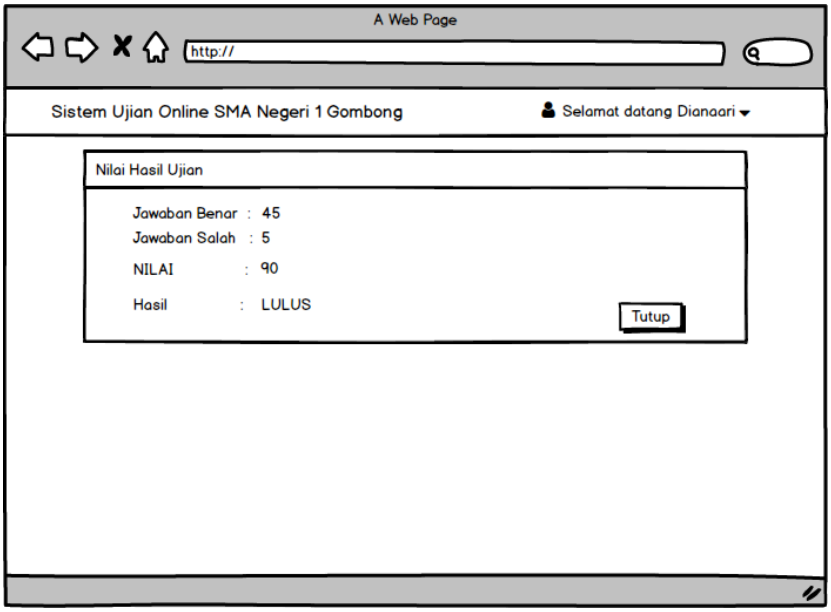
Rancangan antarmuka halaman soal siswa merupakan halaman soal yang tampil setelah siswa memulai ujian. Terdapat timer di bagian sebelah atas soal dan tampilan soal berada dibawahnya. Setelah selesai siswa klik button submit, kemudian baru siswa dialihkan ke halaman nilai hasil ujian untuk melihat hasilnya. Rancangan antarmuka halaman soal ujian siswa dilihat pada Gambar 4.55.



Gambar 4.55 Rancangan Antarmuka Halaman Soal Ujian Siswa

13. Rancangan antarmuka halaman nilai hasil ujian siswa

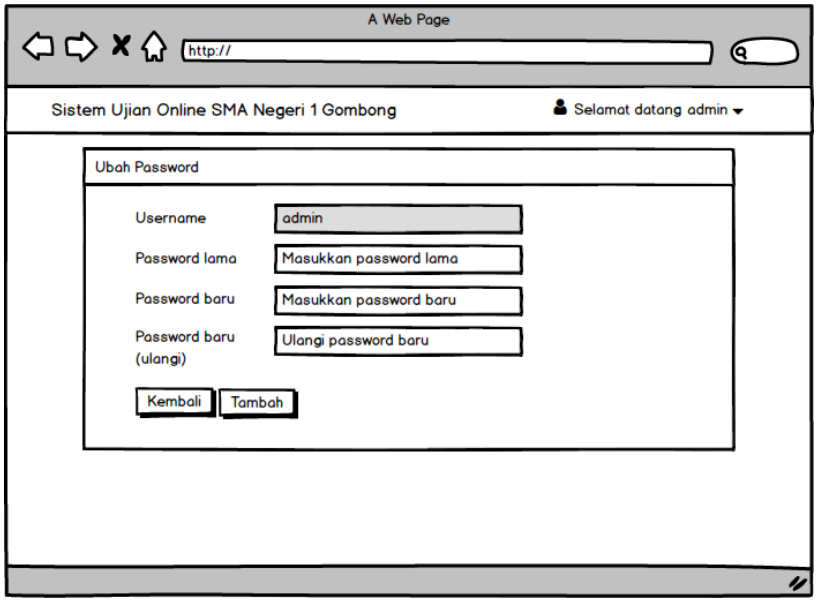
Rancangan antarmuka halaman nilai hasil ujian siswa merupakan rincian dari jawaban siswa dan nilai dari hasil ujian siswa. Siswa dapat melihat jumlah jawaban yang benar dan yang salah. Rancangan antarmuka halaman nilai hasil ujian siswa dapat dilihat pada Gambar 4.56.



Gambar 4.56 Rancangan Antarmuka Halaman Nilai Hasil Ujian Siswa

14. Rancangan antarmuka halaman ubah password

Halaman Ubah *Password* digunakan oleh admin, guru, dan siswa untuk mengubah *password* lama dengan *password* yang baru. Pada halaman ini terdapat form *password* lama, *password* baru dan konfirmasi *password* baru. Rancangan antarmuka halaman ubah *password* dapat dilihat pada Gambar 4.57.



Gambar 4.57 Rancangan Antarmuka Halaman Ubah Password

BAB V
IMPLEMENTASI SISTEM

Tahap implementasi sistem merupakan tahap penerapan dari analisis dan perancangan sistem. Hasil analisis dan rancangan sistem diterjemahkan ke dalam sebuah bahasa pemrograman dan beberapa perangkat lunak untuk membangun sistem. Tahapan implementasi pada penelitian ini meliputi implementasi perangkat lunak pembangun, implementasi perangkat keras pembangun, implementasi basis data, dan implementasi antarmuka.

5.1 Implementasi Perangkat Lunak Pembangun

Perangkat lunak pembangun merupakan perangkat lunak yang digunakan untuk membangun sistem informasi Ujian Online berbasis web pada SMA Negeri 1 Gombong. Adapun perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Windows 7 sebagai sistem operasi.
2. Codeigniter versi 3.0.6 sebagai PHP *framework*.
3. PHP versi 5.6.3 sebagai bahasa pemrograman sisi server.
4. HTML5, CSS3, *javascipt*, *jQuery* sebagai bahasa pemrograman sisi client.
5. Bootstrap sebagai CSS *framework*.
6. Sublime text 2 dan Notepad++ sebagai media perancangan sistem.
7. Balsamiq Mookups, StarUML, dan Microsoft visio sebagai media perancangan analisis dan antarmuka sistem.
8. MySQL sebagai server basis data.
9. Apache versi 2.4.10 sebagai *web server*.
10. XAMPP versi 3.2.1 sebAgai paket aplikasi Apache, MySQl, dan PHP.
11. Google Chrome sebagai *web browser*.

5.2 Implementasi Perangkat Keras Pembangun

Perangkat keras pembangun merupakan perangkat keras yang digunakan untuk membangun sebuah sistem. Perangkat keras yang digunakan untuk

membangun sistem informasi Ujian Online di SMA Negeri 1 Gombong pada penelitian ini adalah sebagai berikut:

1. Laptop Acer Aspire 4752
2. Processor Intel Core I3-2350M
3. RAM (*Random Access Memory*) 2 Gb.
4. Hardisk 500 Gb.
5. Mouse Logitech B100.

5.3 Implementasi Basis Data

Implementasi basis data merupakan implementasi hasil dari analisis rancangan basis data. Implementasi ini meliputi rancangan ERD, rancangan struktur tabel dan rancangan relasi antar tabel.

Pada penelitian ini, basis data yang digunakan adalah basis data ujian. Basis data ujian memiliki beberapa tabel, atribut, dan relasi. Tabel-tabel tersebut memiliki fungsi sesuai dengan kebutuhan penelitian. Yang menjadi perihal penting tentang pembuatan basis data ini adalah mengenai entitas, atribut dan kardinalitasnya. Adapun implementasi dari rancangan basis data yang telah dibuat pada bab sebelumnya adalah sebagai berikut:

5.3.1 Pembuatan Basis Data

Pada penelitian ini menggunakan sebuah basis data yang bernama ujian. Kode program untuk membuat basis data ujian dapat dilihat pada Gambar ....

```
CREATE DATABASE `ujian`;
```

Gambar 5.1 Kode Program Pembuatan Basis Data

5.3.2 Pembuatan Tabel

Basis data ujian memiliki beberapa tabel yang menjadi komponennya. Tabel tersebut diantaranya user, admin, guru, siswa, kelas, tahun, kelas_siswa, mapel, guru_mapel, ujian, soal, hasil_ujian, pertanyaan, dan opsi_jawaban.

1. Tabel User

Tabel user digunakan untuk menyimpan data user pada basisdata ujian. Tabel ini sekaligus digunakan untuk hak akses antara admin, guru, dan siswa. Tabel user terdiri dari username, password, dan level. Kolom username digunakan sebagai *primary key* dari tabel ini. Kode program untuk membuat tabel user dapat dilihat pada Gambar 5.2.

```
CREATE TABLE `user` (
`username` varchar(50) PRIMARY KEY NOT NULL,
`password` varchar(50) NOT NULL,
`level` enum('admin', 'guru', 'siswa') not null
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

Gambar 5.2 Kode Program Pembuatan Tabel User

2. Tabel Admin

Tabel admin digunakan untuk menyimpan data admin pada basisdata ujian. Tabel admin terdiri dari id_admin, nama_admin, no telp, dan username sebagai *foreign key* dari tabel user. Kode Program untuk membuat tabel admin dapat dilihat pada Gambar 5.3.

```
CREATE TABLE `admin` (
`id_admin` bigint(20) PRIMARY KEY NOT NULL
AUTO_INCREMENT,
`nama_admin` varchar(50) NOT NULL,
`no_telp` varchar(15) NOT NULL,
`username` varchar(20) NOT NULL,
CONSTRAINT `user_admin` FOREIGN KEY(`username`)
REFERENCES user(username)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

Gambar 5.3 Kode Proram Pembuatan Tabel Admin

3. Tabel Guru

Tabel guru digunakan untuk menyimpan data guru pada basisdata ujian. Tabel guru terdiri dari nip sebagai *primary key*, nama_guru, no_telp, dan username sebagai *foreign key* dari tabel user. Kode program untuk membuat tabel guru dapat dilihat pada Gambar 5.4.

```
CREATE TABLE `guru` (
`nip` bigint(20) PRIMARY KEY NOT NULL,
`nama_guru` varchar(50) NOT NULL,
```

```
`no_telp` varchar(15) NOT NULL,
`username` varchar(20) NOT NULL,
CONSTRAINT `user_guru` FOREIGN KEY(`username`)
REFERENCES `user` (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

Gambar 5.4 Kode Program Pembuatan Tabel Guru

4. Tabel Siswa

Tabel siswa digunakan untuk menyimpan data siswa pada basisdata ujian. Tabel siswa terdiri dari nis sebagai *primary key*, nama_siswa, jenkel, dan username sebagai *foreign key* dari tabel user. Kode program untuk membuat tabel siswa dapat dilihat pada Gambar 5.5.

```
CREATE TABLE `siswa` (
`nis` bigint(20) PRIMARY KEY NOT NULL,
`nama_siswa` varchar(50) NOT NULL,
`jenkel` enum('L', 'P') NOT NULL,
`username` varchar(20) NOT NULL,
CONSTRAINT  `user_siswa`  FOREIGN  KEY(`username`)
REFERENCES `user` (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

Gambar 5.5 Kode Program Pembuatan Tabel Siswa

5. Tabel Kelas

Tabel kelas digunakan untuk menyimpan data kelas pada basisdata ujian. Tabel kelas terdiri dari id_kelas sebagai *primary key*, jenjang, jenis_kelas, dan nama_kelas. Kode program untuk membuat tabel kelas dapat dilihat pada Gambar 5.6.

```
CREATE TABLE `kelas` (
`id_kelas`  integer(11)  PRIMARY  KEY  NOT  NULL
AUTO_INCREMENT,
`jenjang` enum('X', 'XI', 'XII') NOT NULL,
`jenis_kelas` enum('IPA', 'IPS') DEFAULT NULL,
`nama_kelas` char(2) NOT NULL,
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

Gambar 5.6 Kode Program Pembuatan Tabel Kelas

6. Tabel Mapel

Tabel mapel digunakan untuk menyimpan data mata pelajaran pada basisdata ujian. Tabel mapel terdiri dari id_mapel sebagai *primary key*, nama_mapel, dan kkm. Kode program untuk membuat tabel mapel dapat dilihat pada Gambar 5.7.

```
CREATE TABLE `mapel` (
`id_mapel` varchar(10) PRIMARY KEY NOT NULL,
`nama_mapel` varchar(50) NOT NULL,
`kkm` integer(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

Gambar 5.7 Kode Program Pembuatan Tabel Mapel

7. Tabel Tahun

Tabel tahun digunakan untuk menyimpan data tahun ajaran pada basisdata ujian. Tabel tahun terdiri dari id_tahun sebagai *primary key* dan tahun_ajaran. Kode program untuk membuat tabel tahun dapat dilihat pada Gambar 5.8.

```
CREATE TABLE `tahun` (
`id_tahun` integer(11) PRIMARY KEY NOT NULL
AUTO_INCREMENT,
`tahun_ajaran` varchar(15) NOT NULL,
`status_tahun` boolean
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

Gambar 5.8 Kode Program Pembuatan Tabel Tahun

8. Tabel kelas_siswa

Tabel kelas siswa digunakan untuk menyimpan data kelas siswa pada tahun ajaran tertentu. Tabel kelas siswa terdiri dari nis sebagai *foreign key* dari tabel siswa, id_kelas sebagai *foreign key* dari tabel kelas, dan id_tahun sebagai *foreign key* dari tabel tahun. Kode program untuk membuat tabel kelas siswa dapat dilihat pada Gambar 5.9.

```
CREATE TABLE `kelas_siswa` (
`nis` bigint(20) NOT NULL,
`id_kelas` integer(11) NOT NULL,
`id_tahun` integer(11) n NOT NULL ot null,
CONSTRAINT `tahun_kelsiswa` FOREIGN
KEY(`id_tahun`) REFERENCES `tahun` (`id_tahun`) ON
```

```
DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT `siswa_kelsiswa` FOREIGN KEY (`nis`)
REFERENCES `siswa` (`nis`) ON DELETE CASCADE ON
UPDATE CASCADE,
CONSTRAINT `kelas_kelsiswa` FOREIGN KEY
(`id_kelas`) REFERENCES `kelas` (`id_kelas`) ON
DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

Gambar 5.9 Kode Program Pembuatan Tabel Kelas Siswa

9. Tabel Guru Mapel

Tabel guru mapel digunakan untuk menyimpan data pengajar mata pelajaran di kelas pada tahun ajaran tertentu. Tabel guru mapel terdiri dari nip sebagai *foreign key* dari tabel guru, id_mapel sebagai *foreign key* dari tabel mapel, id_kelas sebagai foreign key dari tabel kelas, dan id_tahun sebagai *foreign key* dari tabel tahun. Kode program untuk membuat tabel guru mapel dapat dilihat pada Gambar 5.10.

```
CREATE TABLE `guru_mapel` (
`nip` bigint(20) PRIMARY KEY NOT NULL,
`id_mapel` varchar(10) NOT NULL,
`id_kelas` integer(11) NOT NULL,
`id_tahun` integer(11) NOT NULL,
CONSTRAINT `guru_grmapel` FOREIGN KEY(`nip`)
REFERENCES `guru` (`nip`),
CONSTRAINT `mapel_grmapel` FOREIGN KEY
(`id_mapel`) REFERENCES `mapel` (`id_mapel`),
CONSTRAINT `kelas_grmapel` FOREIGN KEY
(`id_kelas`) REFERENCES `kelas` (`id_kelas`),
CONSTRAINT `tahun_grmapel` FOREIGN KEY
(`id_tahun`) REFERENCES `tahun` (`id_tahun`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

Gambar 5.10 Kode Program Pembuatan Tabel Guru Mapel

10. Tabel Ujian

Tabel ujian digunakan untuk menyimpan data ujian pada basisdata ujian. Tabel ujian terdiri dari id_ujian sebagai *primary key*, nama_ujian, tanggal_mulai, tanggal_selesai, dan id_tahun sebagai *foreign key* dari tabel

tahun dan status_tahun. Kode program utnuk membuat tabel ujian dapat dilihat pada Gambar 5.11.

```
CREATE TABLE `ujian` (
`id_ujian` varchar(10) PRIMARY KEY NOT NULL,
`nama_ujian` varchar(50) NOT NULL,
`id_tahun` integer(11) NOT NULL,
`tgl_mulai` date NOT NULL,
`tgl_selesai` date NOT NULL,
`status_ujian` enum('Aktif', 'Tidak Aktif') NOT
NULL,
CONSTRAINT `thn_ujian` FOREIGN KEY(`id_tahun`)
REFERENCES `tahun` (`id_tahun`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

Gambar 5.11 Kode Program Pembuatan Tabel Ujian

11. Tabel Soal

Tabel soal digunakan untuk menyimpan data paket soal-soal pada basis data ujian. Tabel soal terdiri dari id_soal sebagai *primary key*, nama_soal, waktu_mulai, waktu_akhir, id_mapel sebagai *foreign key* dari tabel mapel, id_kelas sebagai *foreign key* dari tabel kelas, dan id_ujian sebagai *foreign key* dari tabel ujian. Kode program untuk membuat tabel soal dapat dilihat pada Gambar 5.12.

```
CREATE TABLE `soal` (
`id_soal` int(11) PRIMARY KEY NOT NULL
auto_increment,
`nama_soal` varchar(50) NOT NULL,
`waktu_mulai` timestamp,
`waktu_akhir` timestamp,
`id_mapel` varchar(10) NOT NULL,
`id_kelas` int(11) NOT NULL,
`id_ujian` varchar(10) NOT NULL,
CONSTRAINT `mapel_soal` FOREIGN KEY(`id_mapel`)
REFERENCES `mapel` (`id_mapel`),
CONSTRAINT `kelas_soal` FOREIGN KEY (`id_kelas`)
REFERENCES `kelas` (`id_kelas`),
CONSTRAINT `ujian_soal` FOREIGN KEY (`id_ujian`)
REFERENCES `ujian` (`id_ujian`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

Gambar 5.12 Kode Program Pembuatan Tabel Soal

12. Tabel Hasil Ujian

Tabel hasil ujian digunakan untuk menyimpan data hasil ujian siswa setelah mengerjakan soal ujian pada basisdata ujian. Tabel hasil ujian terdiri dari id_hasil, id_soal sebagai *foreign key* dari tabel soal, nis sebagai *foreign key* dari table siswa, nilai, dan keterangan. Kode program untuk membuat tabel hasil ujian dapat dilihat pada Gambar 5.13.

```
CREATE TABLE `hasil_ujian` (
`id_hasil` integer(11) PRIMARY KEY NOT NULL
AUTO_INCREMENT,
`id_soal` int(11) NOT NULL,
`nis` bigint(20) NOT NULL,
`nilai_ujian` integer NOT NULL,
`keterangan` enum('Lulus', 'Tidak Lulus')  NOT NULL,
CONSTRAINT `soal_ujisiswa` FOREIGN KEY(`id_soal`)
REFERENCES `soal` (`id_soal`),
CONSTRAINT `siswa_ujisiswa` FOREIGN KEY (`nis`)
REFERENCES `siswa` (`nis`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

Gambar 5.13 Kode Program Pembuatan Tabel Hasil Ujian

13. Tabel Pertanyaan

Tabel pertanyaan digunakan untuk menyimpan data pertanyaan pada basisdata ujian. Tabel pertanyaan terdiri dari id_pertanyaan sebagai *primary key*, pertanyaan, jenis_pertanyaan, id_soal sebagai *foreign key* dari tabel soal, dan gambar. Kode program untuk membuat tabel pertanyaan dapat dilihat pada Gambar 5.14.

```
CREATE TABLE `pertanyaan` (
`id_pertanyaan` integer(11) PRIMARY KEY NOT NULL
AUTO_INCREMENT,
`pertanyaan` longtext NOT NULL,
`jenis_pertanyaan` varchar(20) NOT NULL,
`id_soal` int(11) NOT NULL,
`gambar` text DEFAULT NULL,
CONSTRAINT `soal_pertanyaan` FOREIGN KEY(`id_soal`)
REFERENCES `soal` (`id_soal`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

Gambar 5.14 Kode Program Pembuatan Tabel Pertanyaan

14. Tabel Opsi Jawaban

Tabel opsi jawaban digunakan untuk menyimpan data opsi jawaban untuk pertanyan pada basisdata ujian. Tabel opsi jawaban terdiri dari id_opsi sebagai *primary key*, id_pertanyaan sebagai *foreign key* dari tabel pertanyaan, opsi, isi_opsi, jawaban, dan gambar. Kode program untuk membuat tabel opsi jawaban dapat dilihat pada Gambar 5.15.

```
CREATE TABLE `opsi_jawaban` (
`id_opsi` integer(11) PRIMARY KEY NOT NULL
AUTO_INCREMENT,
`id_pertanyaan` integer(11) NOT NULL,
`isi_opsi` longtext NOT NULL,
`jawaban` boolean,
CONSTRAINT `pertanyaan_opsi` FOREIGN
KEY(`id_pertanyaan`) REFERENCES `pertanyaan`
(`id_pertanyaan`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

Gambar 5.15 Kode Program Pembuatan Tabel Opsi Jawaban

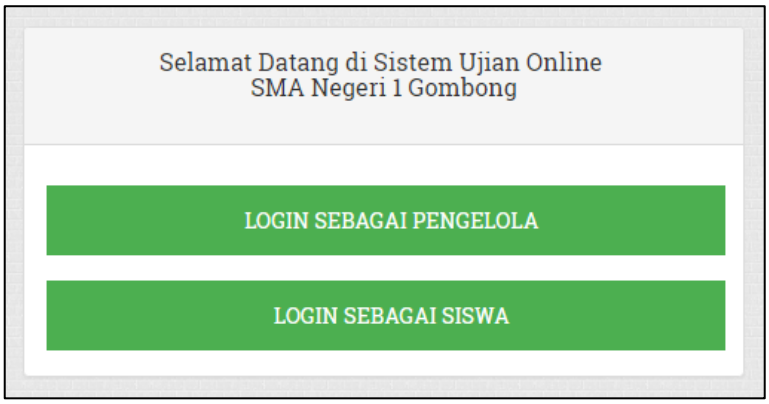
5.4 Implementasi Antarmuka

Implementasi antarmuka merupakan implementasi dari hasil rancangan antarmuka. Pada tahap ini akan menjelaskan implementasi antarmuka dan potongan kode program dari halaman sistem informasi ujian online berbasis web pada SMA Negeri Gombong. Kode program yang dijelaskan merupakan kode yang berhubungan dengan sistem yang telah dibangun.

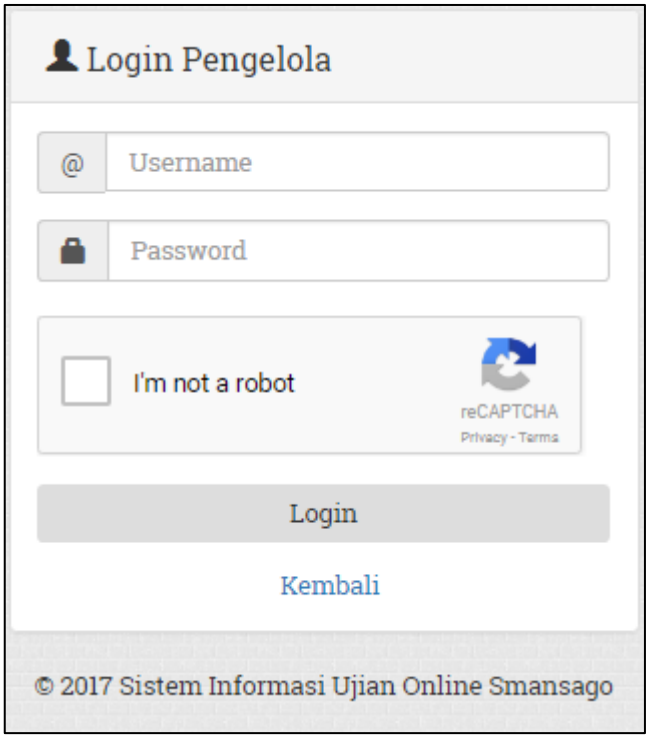
5.4.1 Halaman *Login*

Halaman *login* merupakan halaman yang pertama kali ditampilkan ketika pengguna membuka sistem. Disamping sebagai halaman pertama yang muncul, halaman *login* digunakan oleh pengguna untuk dapat masuk ke dalam sistem sesuai dengan hak aksesnya yang diberikan. Sebelum masuk ke halaman *login*, sistem menampilkan halaman utama *login* yang berisi *button* untuk memilih login sebagai pengelola atau siswa. Implementasi halaman utama *login* dapat dilihat pada Gambar 5.16. Setelah memilih salah satu dari button tersebut, pengguna akan dialihkan menuju halaman *login* pengelola atau halaman *login* siswa.

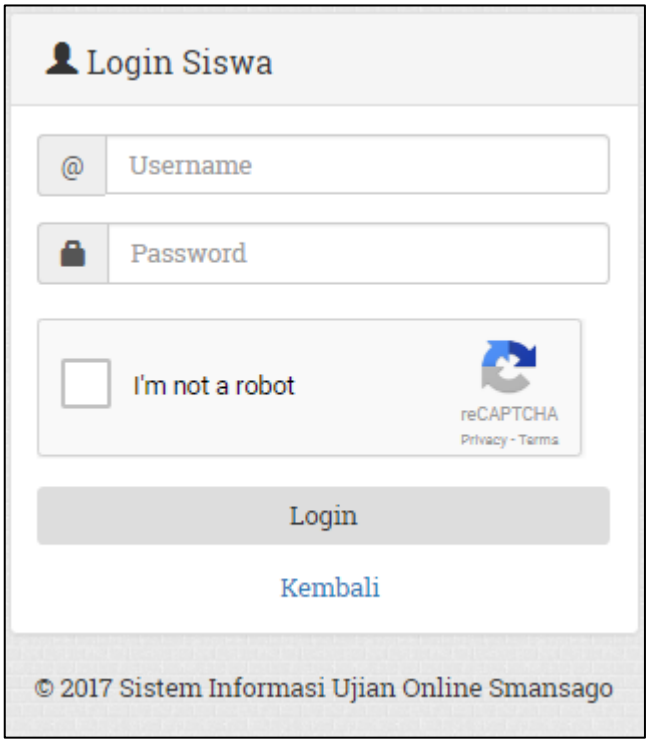
Implementasi halaman *login* pengelola ditunjukan pada Gambar 5.17. Implementasi halaman *login* siswa ditunjukan pada Gambar 5.18.



Gambar 5.16 Tampilan Halaman Utama Untuk *Login*



Gambar 5.17 Tampilan Halaman *Login* Pengelola



Gambar 5.18 Tampilan Halaman *Login* Siswa

Pada Halaman ini akan divalidasi dengan memasukkan username dan password. Kemudian pengguna diminta untuk memasukan *reCAPTCHA* untuk verifikasi apakah pengguna robot atau bukan. Apabila pengguna terdaftar sebagai pengguna, memiliki hak akses pada sistem ini dan berhasil isi *reCAPTCHA* maka sistem akan mengarahkannya ke halaman utama (Dashboard) sesuai hak akses yang diberikan. Kode program untuk implementasi halaman login pengelola dapat dilihat pada Gambar 5.19. Kode program untuk implementasi halaman login siswa dapat dilihat pada Gambar 5.20.

```
26      public function cek_login() {
27          $nama_pengguna = $this->input->post('nama_pengguna');
28          $kata_sandi = $this->input->post('kata_sandi');
29          $hasil = $this->m_login->cek_user($nama_pengguna, $kata_sandi);
30
31          if($hasil->num_rows()>0){
32              foreach($hasil->result() as $data){
33                  $sess_data['username'] = $data->username;
34                  $sess_data['password'] = $data->password;
35                  $sess_data['level'] = $data->level;
36                  $this->session->set_userdata($sess_data);
37              }
38               if($this->session->userdata('level') == 'admin')
39              {
40                  redirect('home');
41              }
42              else if($this->session->userdata('level') == 'guru')
43              {
44                  redirect('home/indexGuru');
45              }
46              else
47              {
48                  $this->session->set_flashdata('Pesan',"<div class='alert alert-danger alert-dismissible'
49                  role='alert'>Kombinasi <strong>Username</strong> dan <strong>Password</strong> tidak valid</div>");
50                  redirect('login/tampil_loginpengelola');
51              }
52          }
53          else {
54              $this->session->set_flashdata('Pesan',"<div class='alert alert-danger alert-dismissible'
55              role='alert'>Kombinasi <strong>Username</strong> dan <strong>Password</strong> tidak valid</div>");
56              redirect('login/tampil_loginpengelola');
57          }
58      }
```

Gambar 5.19 Kode Program Controller Halaman *Login* Pengelola

```
58      public function login_siswa() {
59          $nama_pengguna = $this->input->post('nama_pengguna');
60          $kata_sandi = md5($this->input->post('kata_sandi'));
61          $hasil = $this->m_login->cek_usersiswa($nama_pengguna, $kata_sandi);
62
63          if($hasil->num_rows()>0){
64              foreach($hasil->result() as $data){
65                  $sess_data['username'] = $data->username;
66                  $sess_data['password'] = $data->password;
67                  $sess_data['level'] = $data->level;
68                  $sess_data['nama_siswa'] = $data->nama_siswa;
69                  $sess_data['jenkel'] = $data->jenkel;
70                  $sess_data['nis'] = $data->nis;
71                  $sess_data['jenjang'] = $data->jenjang;
72                  $sess_data['jenis_kelas'] = $data->jenis_kelas;
73                  $sess_data['nama_kelas'] = $data->nama_kelas;
74                  $sess_data['nama_ujian'] = $data->nama_ujian;
75                  $sess_data['nama_mapel'] = $data->nama_mapel;
76
77                  $this->session->set_userdata($sess_data);
78              }
79               if($this->session->userdata('level') == 'siswa')
80              {
81                  redirect('home/indexSiswa');
82              }
83          }
84          else {
85              $this->session->set_flashdata('Pesan',"<div class='alert alert-danger alert-dismissible'
86              role='alert'>Kombinasi <strong>Username</strong> dan <strong>Password</strong> tidak valid</div>");
87              redirect('login/tampil_loginsiswa');
88          }
89      }
90
91
92      function logout()
93      {
94          $this->session->sess_destroy();
95          redirect('home');
96      }
```

Gambar 5.20 Kode Program *Controller* Halaman *Login* Siswa

Berdasarkan kode diatas, pada saat halaman dibuka maka sistem akan membuka halaman login. Setelah pengguna baik siswa, guru maupun admin memasukkan username dan password sistem akan melakukkan pemeriksaan

apakah *username* dan *password* yang dimasukkan sesuai dengan data di tabel user seperti pada baris kode 29 dan 61.

Berdasarkan pada kode 31 untuk function cek_login, jika data yang dimasukkan sesuai dengan data yang terdapat di dalam tabel user maka selanjutnya sistem akan menyimpan sesi data pengguna. Sesi ini berfungsi untuk menampilkan data berdasarkan pengguna yang login / masuk ke dalam sistem. Setelah itu sistem akan melakukan validasi untuk membedakan status pengguna tersebut sesuai dengan baris kode 38 sampai 51. Jika status pengguna sebagai admin maka sistem akan mengalihkan ke controler home(*dashboard admin*) dan sebaliknya jika status pengguna tersebut adalah guru maka sistem akan mengalihkan ke *controler* home dengan *function* indexGuru(dashboard guru).

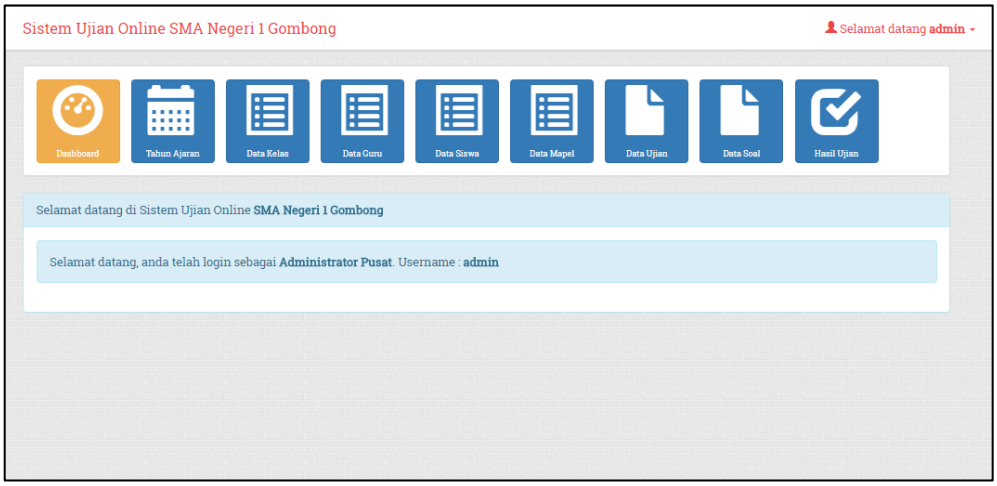
Berdasarkan pada kode 63 untuk *function* login_siswa, jika data yang dimasukkan sesuai dengan data yang terdapat di dalam tabel user *join* dengan siswa maka selanjutnya sistem akan menyimpan sesi data pengguna. Sesi ini berfungsi untuk menampilkan data siswa berdasarkan pengguna yang *login* / masuk ke dalam sistem. Jika status pengguna sebagai siswa maka sistem akan mengalihkan ke *controller* home dengan function indexSiswa (dashboard siswa).

Apabila data *username* dan *password* yang dimasukkan pengguna tidak terdapat di tabel guru, maka sistem akan menjalankan baris kode ke 54 dan 85. Pada baris tersebut merupakan pilihan lain jika data yang *username* dan *password* tidak ditemukan dalam tabel. Sistem akan mengembalikan ke *controller Login* untuk kembali melakukan *login* pengguna dengan menampilkan peringatan *alert* yang berisi “Username dan Password tidak valid”.

5.4.2 Halaman Utama (Dashboard) Admin

Apabila validasi pada proses di *Controller Login* berhasil, maka admin akan diarahkan menuju halaman utama admin yang berisi konten selamat datang. Pada bagian header bagian kanan halaman terdapat panel yang menampilkan *username* dari admin yang masuk ke dalam sistem. Seperti terlihat pada *Controller*, $this→session→userdata(‘username’) artinya mengambil data

username untuk ditampilkan di bagian *dashboard admin*. Implemetasi halaman utama admin ditunjukkan pada Gambar 5.21. Kode program untuk implementasi halaman utama admin dapat dilihat pada Gambar 5.22.



Gambar 5.21 Tampilan Halaman Utama Admin

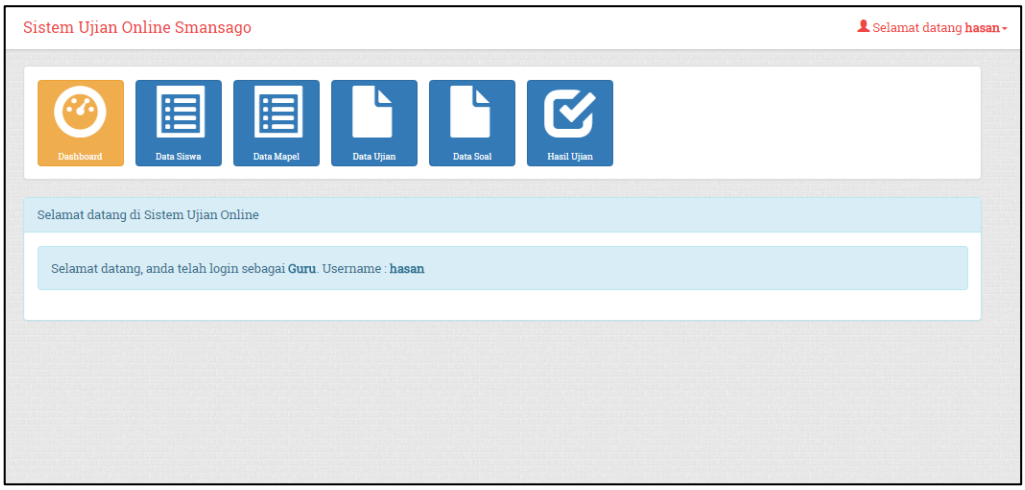
```
4   class Home extends CI_Controller
5   {
6
7       public function __construct()
8       {
9           parent::__construct();
10          if (!$this->session->userdata('username')) {
11              redirect('login');
12          }
13
14      }
15          public function index() {
16          $a['username'] = $this->session->userdata('username');
17          $a['p']        = "Admin/v_home";
18          $this->load->view('vwHeader',$a);
19
20          }
```

Gambar 5.22 Kode Program Controller Halaman Utama Admin

5.4.3 Halaman Utama (Dashboard) Guru

Sistem akan mengarahkan guru ke halaman utama guru jika status yang bersangkutan dalam basis data merupakkan guru. Pada dasarnya struktur dari *controller* home guru tidak jauh berbeda dengan *controller* home *admin*. Hanya saja perbedaannya pada halaman view yang dipanggil. Implementasi halaman

utama guru ditunjukan pada Gambar 5.23. Kode program implementasi halaman utama guru dapat dilihat pada Gambar 5.24.



Gambar 5.23 Tampilan Halaman Utama Guru

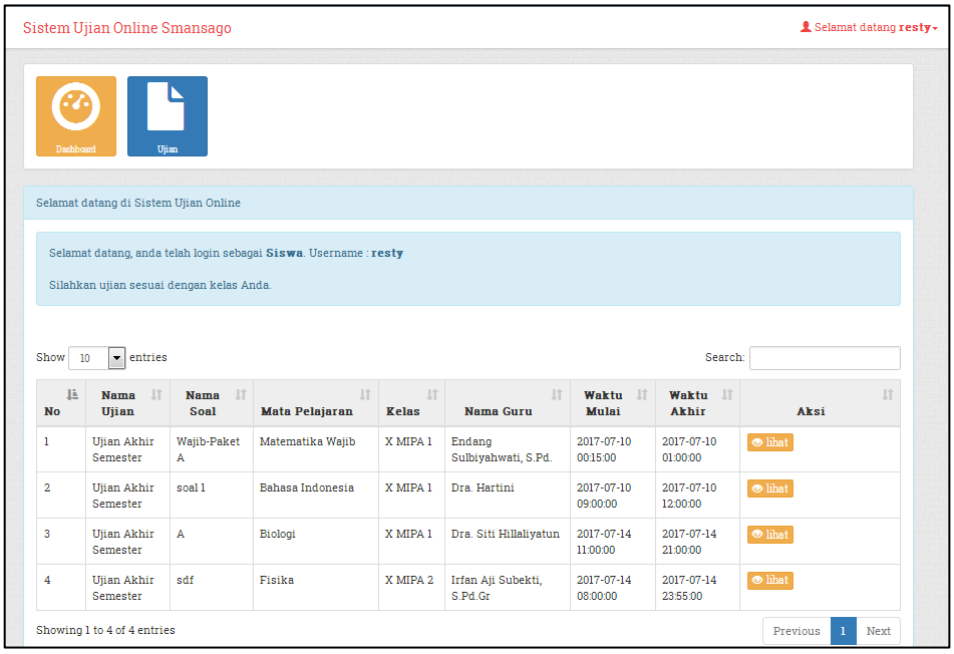
```
22        public function indexGuru() {
23        $a['username'] = $this->session->userdata('username');
24        $a['p']          = "Guru/vhome_g";
25        $this->load->view('Guru/template_guru',$a);
26
27        }
28
29        public function indexSiswa() {
30        $a['username'] = $this->session->userdata('username');
31        $a['p']          = "Siswa/vhome_s";
32        $this->load->view('Siswa/template_siswa',$a);
33
34        }
```

Gambar 5.24 Kode Program Controller Halaman Utama Guru

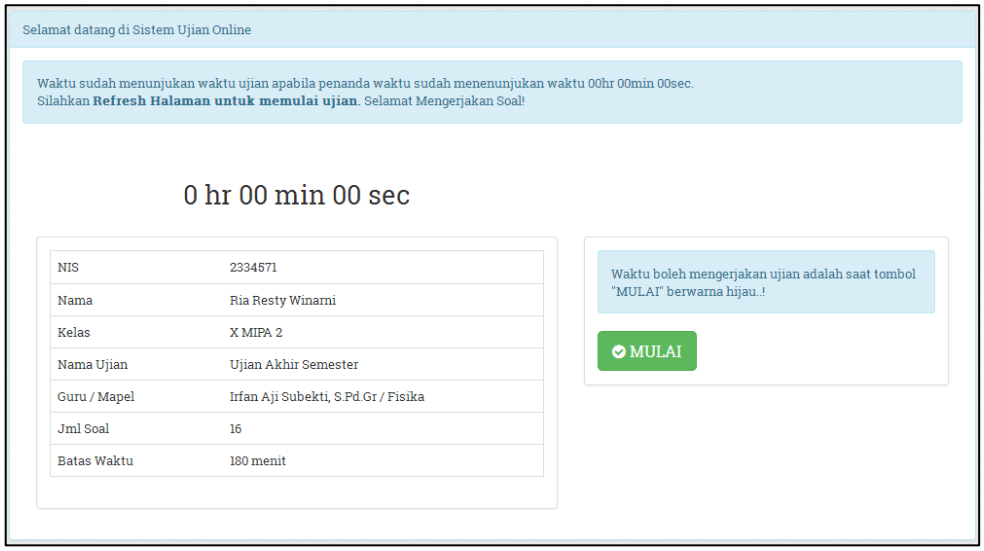
5.4.4 Halaman Utama (Dashboard) Siswa

Sistem akan mengarahkan siswa ke halaman utama siswa jika status yang bersangkutan dalam basisdata merupakan siswa. Di halaman utama siswa ini berbeda dengan halaman utama guru dan admin. Di halaman utama siswa ini ditampilkan daftar ujian yang sedang berlangsung. Siswa dapat memilih untuk mengikuti ujian sesuai dengan kelas nya masing-masing. Kemudian akan dialihkan menuju halaman utama ujian siswa. Di halaman utama ujian siswa ini menampilkan data siswa, ujian, guru pengajar, mapel, dan rincian soal yang aktif yang akan digunakan untuk memulai ujian. Implementasi halaman utama siswa

ditunjukan pada Gambar 5.25. Implementasi halaman utama ujian siswa ditunjukan pada Gambar 5.26. Adapun potongan kode program halaman utama ujian siswa dapat dilihat pada Gambar 5.27.



Gambar 5.25 Tampilan Halaman Utama Siswa



Gambar 5.26 Tampilan Halaman Utama Ujian Siswa

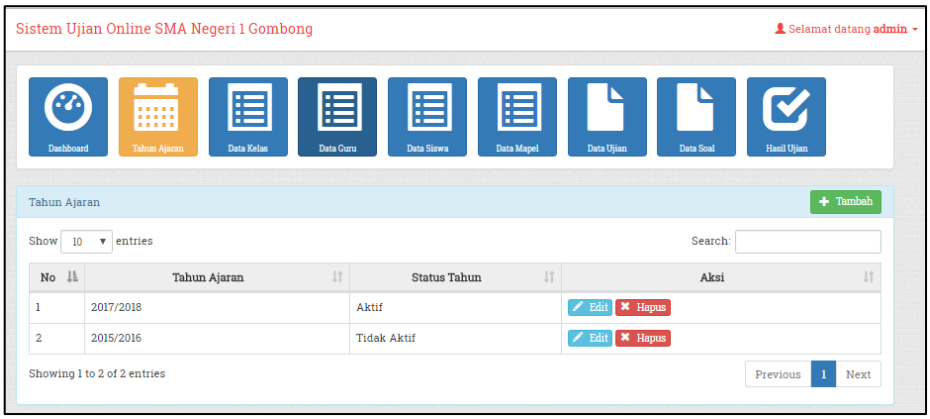
```
32          public function indexSiswa(){
33          $a['username'] = $this->session->userdata('username');
34          $a['p']        = "Siswa/v_dashboardsiswa";
35          $a['soal'] = $this->m_soal->getDaftarUjian();
36          $this->load->view('Siswa/template_siswa',$a);
37
38          }
39
40          public function detailindexSiswa($id_soal) {
41
42          $id = $this->m_soal->getIdkelas($this->session->userdata('username'),$this->session->userdata('password'));
43
44          foreach ($id as $key) {
45              $id_kelas = $key->id_kelas;
46
47          }
48
49          $a['username'] = $this->session->userdata('username');
50          $a['nis'] = $this->session->userdata('nis');
51          $a['nama_siswa'] = $this->session->userdata('nama_siswa');
52          $a['jenjang'] = $this->session->userdata('jenjang');
53          $a['jenis_kelas'] = $this->session->userdata('jenis_kelas');
54          $a['nama_kelas'] = $this->session->userdata('nama_kelas');
55          $a['waktu'] = $this->m_pertanyaan->tampil_waktu()->row_array();
56          $a['data_ujian'] = $this->m_soal->ujian(date("Y-m-d H:i:s",time()),$id_kelas)->result();
57          $a['jmlsoal'] = $this->m_pertanyaan->getJmlsoal(28);
58          $a['pertanyaan'] = $this->session->userdata('pertanyaan');
59          $a['soal']  = $this->m_soal->getSoal($this->session->userdata('id_ujian'),$this->session->userdata('id_kelas'), date("Y-m-d H:i:s",time())
60
61          $a['id_ujian'] = $this->session->userdata('id_ujian');
62          $a['id_kelas'] = $this->session->userdata('id_kelas');
63          $a['tgl'] = date("Y-m-d H:i:s",time());
64
65          $waktu = $this->m_pertanyaan->tampil_sisa($id_soal)->row_array();
66
67          $timestamp2 = strtotime($waktu['waktu_mulai']);
68
69          $a['waktuu'] = date("Y-m-d H:i:s", $timestamp2);
70
71          $a['p']        = "Siswa/vhome_s";
72          $this->load->view('Siswa/template_siswa',$a);
73
74          }
```

Gambar 5.27 Kode Program Controller Halaman Utama Ujian Siswa

5.4.5 Halaman Tahun Ajaran

1. Data Tahun Ajaran

Setelah memilih menu data tahun ajaran, admin akan dialihkan menuju halaman data kelas. Sistem akan menampilkan seluruh data tahun ajaran yang tersimpan. Berikut implementasi tampilan halaman data kelas dapat dilihat pada Gambar 5.28.



Gambar 5.28 Tampilan Halaman Data Tahun Ajaran

2. Tambah Tahun Ajaran

Setelah klik *button* tambah yang berada dibagian kanan atas halaman, admin akan dialihkan menuju halaman tambah data tahun ajaran. Disini admin

menambahkan tahun ajaran untuk diaktifkan. Apabila data berhasil ditambahkan akan tampil *alert* ‘Data berhasil ditambahkan’. Impelementasi halaman tambah data tahun ajaran ditunjukan pada Gambar 5.29. Adapun potongan kode program halaman tambah tahun ajaran dapat dilihat pada Gambar 5.30.

Gambar 5.29 Tampilan Halaman Tambah Tahun Ajaran

```
61      public function add(){
62              $this->form_validation->set_rules('tahun_ajaran','tahun_ajaran', 'trim|required');
63              $this->form_validation->set_rules('status_tahun','status_tahun', 'trim|required');
64              $this->form_validation->set_rules('trailer', 'Trailer', 'trim|is_numeric');
65
66              if($this->form_validation->run() === FALSE){
67                  $a['p']='Admin/tahun/v_addtahun';
68                  $this->load->view('vwHeader',$a);
69              }else{
70                      $tahun_ajaran = $this->input->post('tahun_ajaran');
71                      $status_tahun = $this->input->post('status_tahun');
72
73
74                      $object = array(
75                          "tahun_ajaran" => $tahun_ajaran,
76                          "status_tahun" => $status_tahun,
77              );
78                      $this->m_tahun->insert($object);
79                      $this->session->set_flashdata("Pesan", "<div class='alert alert-success' alert-dismissible' role='alert'>
80                          <button type='button' class='close' data-dismiss='alert' aria-label='Close'>
81                          <span aria-hidden='true'>&times;</span></button>Data Berhasil ditambahkan</div>");
82                      redirect('Admin/tahun');
83                  }
84      }
85
86
87      public function tampil(){
88          $this->load->library('form_validation');
89
90          $a['username'] = $this->session->userdata('username');
91          $a['p']='Admin/tahun/v_addtahun';
92          $a['tahun'] = $this->m_tahun->ModelTahun()->result_array();
93          $this->load->view('vwHeader',$a);
94      }
```

Gambar 5.30 Kode Program Controller Halaman Tambah Tahun

3. Ubah Tahun ajaran

Untuk mengubah data tahun ajaran, admin dapat memilih aksi *edit* pada kolom tabel aksi. Kemudian admin akan dialihkan menuju halaman ubah data tahun ajaran. Disini admin dapat mengubah status tahun ajaran menjadi tidak aktif yang sudah tidak digunakan. Impelementasi halaman ubah data tahun ajaran ditunjukan pada Gambar 5.31. Adapun potongan kode program halaman ubah data tahun ajaran dapat dilihat pada Gambar 5.32.

Gambar 5.31 Tampilan Halaman Ubah Tahun Ajaran

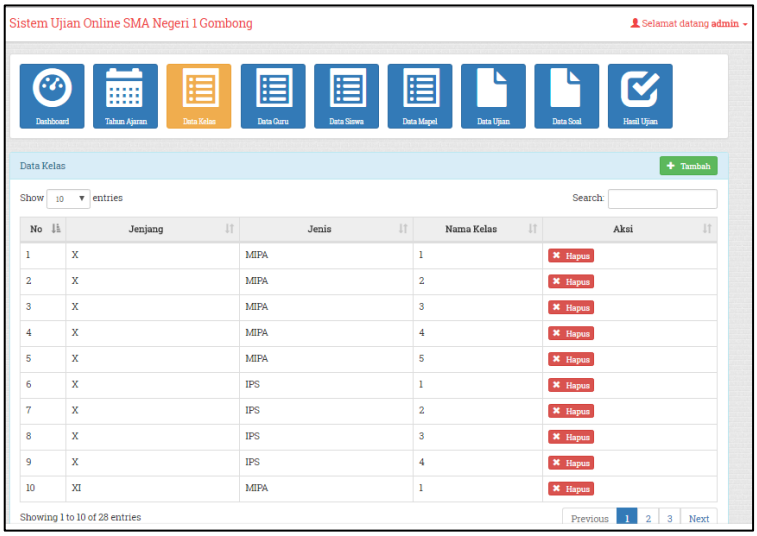
```
32      public function edit($id_tahun){
33
34          if($this->input->post('submit')){
35              $id_tahun1 = $this->input->post('id_tahun');
36              $tahun_ajaran = $this->input->post('tahun_ajaran');
37              $status_tahun = $this->input->post('status_tahun');
38
39              $object = array(
40                  "id_tahun" => $id_tahun1,
41                  "tahun_ajaran" => $tahun_ajaran,
42                  "status_tahun" => $status_tahun,
43              );
44
45
46          $this->db->where('id_tahun', $id_tahun);
47          $this->db->update('tahun',$object);
48          $this->session->set_flashdata("Pesan", "<div class='alert alert-success' alert-dismissible' role='alert'>
49              <button type='button' class='close' data-dismiss='alert' aria-label='Close'>
50              <span aria-hidden='true'>&times;</span></button>Data Berhasil disimpan</div>");
51          redirect('Admin/tahun');
52          }else{
53
54              $a['username'] = $this->session->userdata('username');
55              $a['p']='Admin/tahun/v_edittahun';
56              $a['editdata'] = $this->db->where('id_tahun',$id_tahun)->get('tahun')->result_array();
57              $this->load->view('vwHeader',$a);
58          }
59      }
```

Gambar 5.32 Kode Program Controller Halaman Ubah Tahun

5.4.6 Halaman Data Kelas

a. Data Kelas

Setelah memilih menu data kelas, admin akan dialihkan menuju halaman data kelas. Disini sistem akan menampilkan seluruh data kelas yang tersimpan dan terdaftar pada basis data kelas dalam bentuk tabel. Data kelas disini berisi data kelas-kelas yang terdapat di SMA Negeri 1 Gombong. Tabel yang digunakan pada data kelas menggunakan datatables sehingga kelas dapat diurutkan sesuai kolom yang dipilih. Berikut implementasi tampilan halaman data kelas dapat dilihat pada Gambar 5.33.



Gambar 5.33 Tampilan Halaman Data Kelas

b. Tambah Data Kelas

Setelah klik *button* tambah yang berada dibagian kanan atas halaman, admin akan dialihkan menuju halaman tambah data kelas. Disini admin dapat menginputkan Jenjang kelas, jenis kelas, dan nama kelas. Apabila data berhasil ditambahkan akan tampil *alert* 'Data berhasil ditambahkan'. Impelementasi halaman tambah data kelas ditunjukan pada Gambar 5.34. Adapun potongan kode program halaman tambah data kelas dapat dilihat pada Gambar 5.35.

Gambar 5.34 Tampilan Halaman Tambah Data Kelas

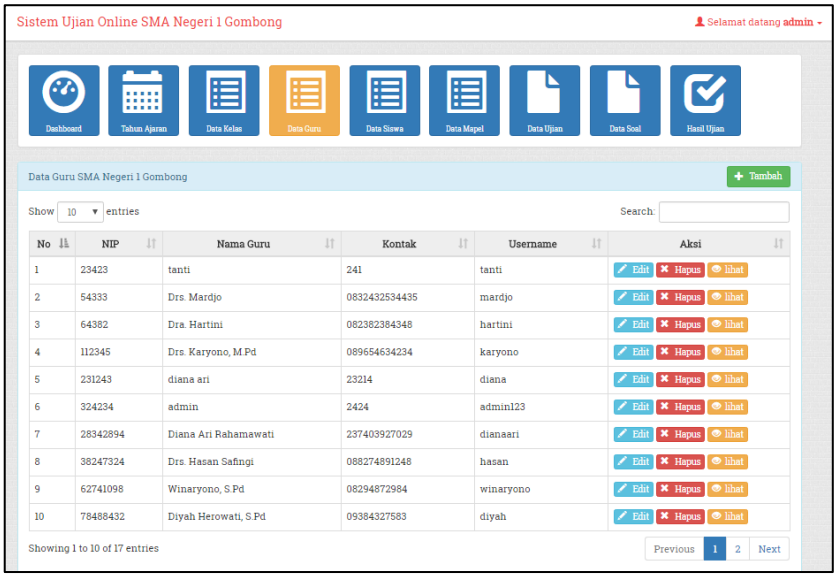
```
68      public function add(){
69              $this->form_validation->set_rules('jenjang','jenjang', 'trim|required');
70              $this->form_validation->set_rules('jenis_kelas','jenis_kelas', 'trim|required');
71              $this->form_validation->set_rules('nama_kelas','nama_kelas', 'trim|required');
72
73              if($this->form_validation->run() === FALSE){
74                  $a['p']='Admin/kelas/v_addkelas';
75                  $this->load->view('vwHeader',$a);
76              }else{
77                      $jenjang = $this->input->post('jenjang');
78                      $jenis_kelas = $this->input->post('jenis_kelas');
79                      $nama_kelas = $this->input->post('nama_kelas');
80
81                      $object = array(
82                          "jenjang" => $jenjang,
83                          "jenis_kelas" => $jenis_kelas,
84                          "nama_kelas" => $nama_kelas,
85
86                      );
87
88                      $this->m_kelas->insert($object);
89                      $this->session->set_flashdata("Pesan", "<div class='alert alert-success' alert-dismissible' role='alert'>
90                          <button type='button' class='close' data-dismiss='alert' aria-label='Close'>
91                          <span aria-hidden='true'>&times;</span></button>Data Berhasil ditambahkan</div>");
92                      redirect('Admin/kelas');
93              }
94      }
95
96      public function tampil(){
97          $this->load->library('form_validation');
98
99          $a['username'] = $this->session->userdata('username');
100         $a['p']='Admin/kelas/v_addkelas';
101         $a['kelas'] = $this->m_kelas->ModelKelas()->result_array();
102         $this->load->view('vwHeader',$a);
103     }
```

Gambar 5.35 Kode Program Controller Halaman Tambah Data Kelas

5.4.7 Halaman Data Guru

a. Data Guru

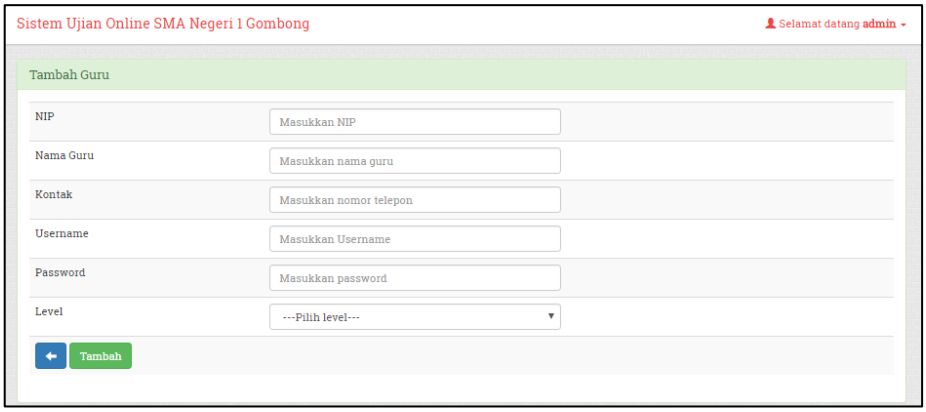
Halaman informasi data guru menampilkan seluruh data guru yang mengajar. Tabel yang digunakan pada data guru juga menggunakan *datatables* sehingga guru dapat diurutkan sesuai kolom yang dipilih. Berikut implementasi tampilan halaman data guru dapat dilihat pada Gambar 5.36.



Gambar 5.36 Tampilan Halaman Data Guru

b. Tambah Data Guru

Setelah klik *button* tambah yang berada dibagian kanan atas halaman, admin akan dialihkan menuju halaman tambah data guru. Pada saat melakukan penambahan data guru, maka sistem akan melakukan pengecekkan terhadap nip dan *username* yang dimasukkan. Kode program pada kondisi ini dapat dilihat pada baris ke 90 dan 97. Apabila nip ditemukan maka sistem akan menjalankan kondisi selanjutnya yaitu melakukan pemeriksaan apakah nip tersebut sudah dipakai. Jika ditemukan maka sistem akan memberikan pesan peringatan 'NIP sudah terdaftar sebagai guru'. Selain itu sistem juga akan melakukan pengecekan terhadap *username* yang diinputkan. Apabila *username* ditemukan maka sistem akan menjalankan kondisi selanjutnya yaitu melakukan pemeriksaan apakah *username* sudah dipakai. Jika ditemukan maka sistem akan memberikan pesan peringatan '*Username* sudah digunakan. Silahkan ganti dengan *username* baru!'. Hal ini sebab tidak diperbolehkan terdapat data guru dengan nip yang sama atau *username* yang sama. Jika tidak ditemukan yang sama pada basis data maka sistem akan menginpuitkan data guru baru dan sistem akan memberikan pesan 'Data berhasil ditambahkan', terlihat pada kode program baris ke 129. Pada saat Implementasi halaman tambah data guru ditunjukan pada Gambar 5.37. Kode program *Controller* tambah data guru dapat dilihat pada Gambar 5.38 dan 5.39.



Gambar 5.37 Tampilan Halaman Tambah Data Guru

```
77      public function add(){
78              $this->form_validation->set_rules('username','username', 'trim|required');
79              $this->form_validation->set_rules('password','password', 'trim|required');
80              $this->form_validation->set_rules('level','level', 'trim|required');
81              $this->form_validation->set_rules('nip','nip', 'trim|required');
82              $this->form_validation->set_rules('nama_guru','nama_guru', 'trim|required');
83              $this->form_validation->set_rules('no_telp','no_telp', 'trim|required');
84
85              if($this->form_validation->run() === TRUE){
86                  $nip = $this->input->post('nip');
87                  $username = $this->input->post('username');
88                  $check=$this->m_guru->check($nip);
89                  $check2=$this->m_guru->check2($username);
90                  if($check->num_rows()!=null){
91                      $a['username'] = $this->session->userdata('username');
92                      $a['message']="<div class='alert alert-danger' alert-dismissible' role='alert'><button type='button'
93                      class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span>
94                      </button>NIP sudah terdaftar sebagai Guru</div>";
95                      $a['p']='Admin/guru/v_addguru';
96                      $this->load->view('vwHeader',$a);
97                  }else if($check2->num_rows()!=null){
98                      $a['username'] = $this->session->userdata('username');
99                      $a['message']="<div class='alert alert-danger' alert-dismissible' role='alert'><button type='button'
100                     class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span>
101                     </button>Username sudah digunakan. Silahkan ganti dengan username baru!</div>";
102                     $a['p']='Admin/guru/v_addguru';
103                     $this->load->view('vwHeader',$a);
104                 }
105                 else{
106                     $username = $this->input->post('username');
107                     $password = md5($this->input->post('password'));
108                     $level = $this->input->post('level');
109                     $nip = $this->input->post('nip');
110                     $nama_guru = $this->input->post('nama_guru');
111                     $no_telp = $this->input->post('no_telp');
112
113                     $object_user = array(
114                         "username" => $username,
115                         "password" => $password,
116                         "level" => $level,
117                     );
```

Gambar 5.38 Kode Program *Controller* Halaman Tambah Data Guru

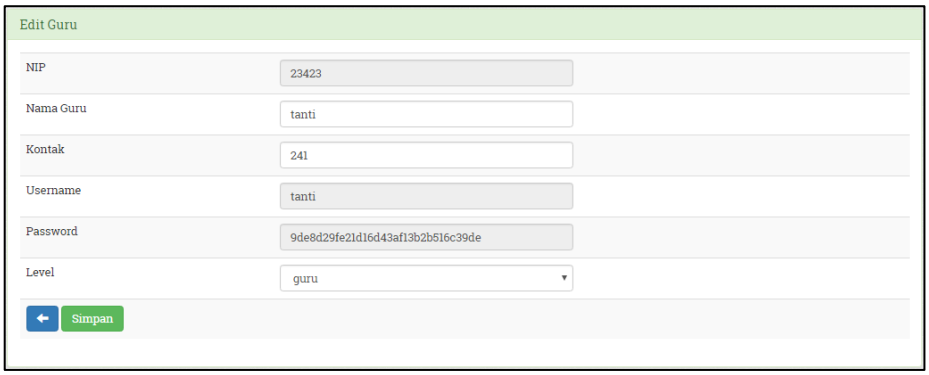
```
118
119                     $object = array(
120                         "nip" => $nip,
121                         "nama_guru" => $nama_guru,
122                         "no_telp" => $no_telp,
123                         "username" => $username,
124                     );
125
126                     $this->m_guru->insert_user($object_user);
127                     $this->m_guru->insert($object);
128
129                     $this->session->set_flashdata("Pesan", "<div class='alert alert-success' alert-dismissible' role='alert'>
130                         <button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span>
131                         </button>Data Berhasil ditambahkan</div>");
132                     redirect('Admin/guru');
133
134                 }
135             }else{
136                 $a['message']="";
137                 $a['p']='Admin/guru/v_addguru';
138                 $this->load->view('vwHeader',$a);
139             }
140     }
141
142     public function tampil(){
143         $this->load->library('form_validation');
144
145         $a['username'] = $this->session->userdata('username');
146         $a['p']='Admin/guru/v_addguru';
147         $a['guru'] = $this->m_guru->ModelGuru();
148         $a['message']=validation_errors();
149         $this->load->view('vwHeader',$a);
150     }
```

Gambar 5.39 Kode Program *Controller* Tambah Data Guru (Lanjutan)

c. Ubah Data Guru

Setelah admin memilih salah satu data guru, admin dapat memilih aksi edit. Admin dapat mengubah data guru berdasarkan nip. Pada form edit sistem akan mengunci form nip sehingga nip guru tidak dapat diubah. Berikut tampilan

antarmuka halaman ubah guru dapat dilihat pada Gambar 5.41. Implementasi Kode program Controller ubah data guru dapat dilihat pada Gambar 5.42.



Gambar 5.40 Tampilan Halaman Ubah Data Guru

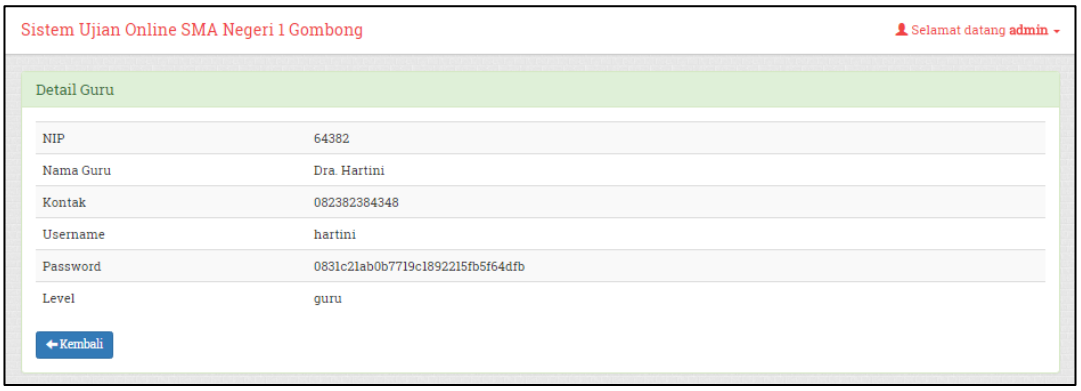
```
34      public function edit($nip){
35
36          if($this->input->post('submit')){
37              $username = $this->input->post('username');
38              $password = md5($this->input->post('password'));
39              $level = $this->input->post('level');
40              $nip1 = $this->input->post('nip');
41              $nama_guru = $this->input->post('nama_guru');
42              $no_telp = $this->input->post('no_telp');
43
44              $object_user = array(
45                  "username" => $username,
46                  "password" => $password,
47                  "level" => $level,
48              );
49
50              $object = array(
51                  "nip" => $nip1,
52                  "nama_guru" => $nama_guru,
53                  "no_telp" => $no_telp,
54                  "username" => $username,
55              );
56
57          $this->db->where('username', $username);
58          $this->db->update('user',$object_user);
59
60          $this->db->where('nip', $nip);
61          $this->db->update('guru',$object);
62
63          $this->session->set_flashdata("Pesan", "<div class='alert alert-success' alert-dismissible' role='alert'><button type='button'
64              class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span></button>Data Berhasil disimpan</div>");
65          redirect('Admin/guru');
66          }else{
67
68              $a['username'] = $this->session->userdata('username');
69              $a['p']='Admin/guru/v_editguru';
70              $a['editdata'] = $this->db->where('nip',$nip)->join('user','user.username=guru.username')->get('guru')->result_array();
71              $this->load->view('vwHeader',$a);
72          }
73
74      }
```

Gambar 5.41 Kode Program *Controller* Halaman Ubah Data Guru

d. Lihat Detail Guru

Setelah admin memilih aksi lihat pada kolom tabel aksi, admin akan dialihkan menuju halaman detail data guru. Sistem akan menampilkan NIP, nama guru, No hp, Kelas, Username, Password, dan Level. Pada form detail data guru, sistem hanya menampilkan data lengkap guru sesuai nip dan tidak bisa diubah-ubah. Implementasi antarmuka halaman lihat detail guru dapat dilihat pada

Gambar 5.42. Implementasi Kode program Controller lihat detail guru dapat dilihat pada Gambar 5.43.



Gambar 5.42 Tampilan Halaman Lihat Detail Guru

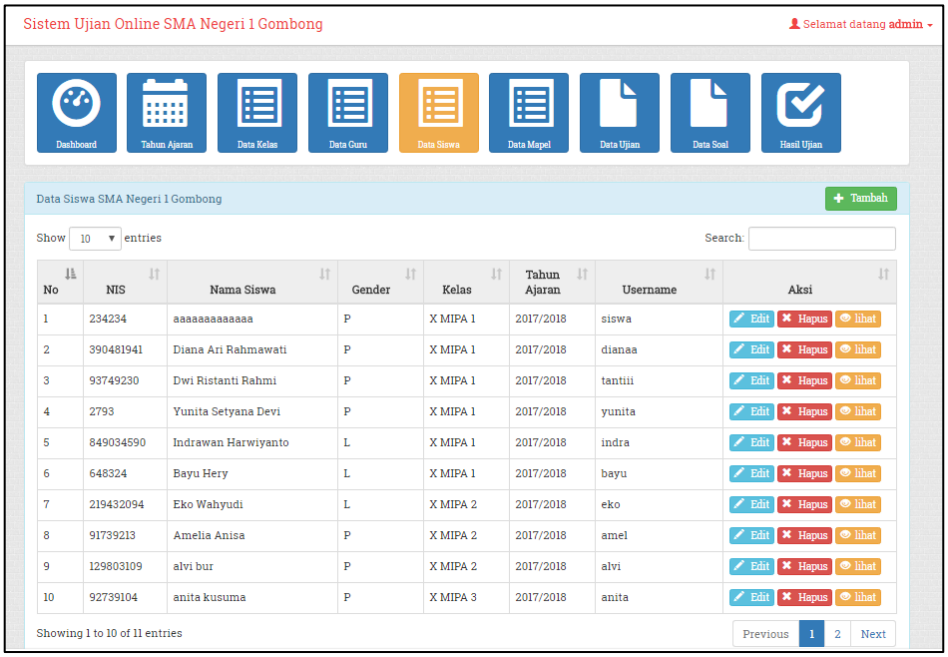
```
140     public function detail($nip) {
141         $this->load->library('form_validation');
142
143         $a['username'] = $this->session->userdata('username');
144         $a['p']='Admin/guru/v_detailguru';
145         $a['guru'] = $this->m_guru->get_detail($nip)->row_array();
146         $this->load->view('vwHeader',$a);
147
148     }
```

Gambar 5.43 Kode Program *Controller* Halaman Lihat Detail Guru

5.4.8 Halaman Data Siswa untuk Admin

1. Data Siswa

Setelah memilih menu Data Siswa pada bagian *sidebar*, admin akan dialihkan menuju halaman data siswa. Di halaman ini sistem akan menampilkan seluruh data siswa yang telah didaftarkan dan tersimpan di basis data siswa dalam bentuk tabel. Data yang ditampilkan diurutkan berdasarkan nis terbesar. Selain itu pada masing-masing kolom siswa terdapat aksi yang bisa digunakan oleh *Admin* dalam mengelola data siswa. Berikut Implementasi tampilan halaman data siswa dapat dilihat pada Gambar 5.44.



Gambar 5.44 Tmapilan Halaman Data Siwa

2. Tambah Data Siswa

Pada saat melakukan penambahan data siswa, maka sistem akan melakukan pengecekkan terhadap nis dan *username* yang dimasukkan. Kode program pada kondisi ini dapat dilihat pada baris ke 105 dan 112. Apabila nis ditemukan maka sistem akan menjalankan kondisi selanjutnya yaitu melakukan pemeriksaan apakah nis tersebut sudah dipakai. Jika ditemukan maka sistem akan memberikan pesan peringatan 'NIS sudah terdaftar sebagai siswa'. Selain itu sistem juga akan melakukan pengecekan terhadap *username* yang diinputkan. Apabila username ditemukan maka sistem akan menjalankan kondisi selanjutnya yaitu melakukan pemeriksaan apakah username sudah dipakai. Jika ditemukan maka sistem akan memberikan pesan peringatan '*Username* sudah digunakan. Silahkan ganti dengan *username* baru!'. Hal ini sebab tidak diperbolehkan terdapat data siswa dengan nis yang sama atau *username* yang sama. Jika tidak ditemukan yang sama pada basis data maka sistem akan menginpuitkan data guru baru dan sistem akan memberikan pesan 'Data berhasil ditambahkan', terlihat pada kode program baris ke 152. Implementasi halaman tambah data siswa

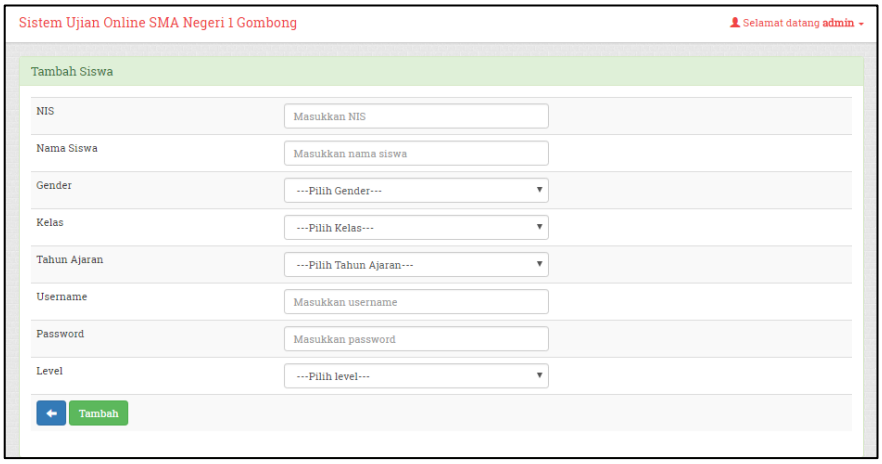
ditunjukan pada Gambar 5.47. Kode program *Controller* tambah data siswa dapat dilihat pada Gambar 5.45 dan 5.46.

```
90      public function add(){
91              $this->form_validation->set_rules('nis','nis', 'trim|required');
92              $this->form_validation->set_rules('nama_siswa','nama_siswa', 'trim|required');
93              $this->form_validation->set_rules('jenkel','jenkel', 'trim|required');
94              $this->form_validation->set_rules('kelas','kelas', 'trim|required');
95              $this->form_validation->set_rules('tahun','tahun', 'trim|required');
96              $this->form_validation->set_rules('username','username', 'trim|required');
97              $this->form_validation->set_rules('password','password', 'trim|required');
98              $this->form_validation->set_rules('level','level', 'trim|required');
99
100             if($this->form_validation->run() === TRUE){
101                 $nis = $this->input->post('nis');
102                 $username = $this->input->post('username');
103                 $check=$this->m_siswa->check($nis);
104                 $check2=$this->m_siswa->check2($username);
105                 if($check->num_rows()!=null){
106                     $a['username'] = $this->session->userdata('username');
107                     $a['message']="<div class='alert alert-danger' alert-dismissible' role='alert'><button type='button'
108                     class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span>
109                     </button>NIS sudah terdaftar sebagai Siswa</div>";
110                     $a['p']='Admin/siswa/v_addsiswa';
111                     $this->load->view('vwHeader',$a);
112                 }else if ($check2->num_rows()!=null){
113                     $a['username'] = $this->session->userdata('username');
114                     $a['message']="<div class='alert alert-danger' alert-dismissible' role='alert'><button type='button'
115                     class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span>
116                     </button>Username sudah digunakan. Silahkan ganti dengan username baru!</div>";
117                     $a['p']='Admin/siswa/v_addsiswa';
118                     $this->load->view('vwHeader',$a);
119                 }else{
120                     $nis = $this->input->post('nis');
121                     $nama_siswa = $this->input->post('nama_siswa');
122                     $jenkel = $this->input->post('jenkel');
123                     $kelas = $this->input->post('kelas');
124                     $tahun = $this->input->post('tahun');
125                     $username = $this->input->post('username');
126                     $password = md5($this->input->post('password'));
127                     $level = $this->input->post('level');
128
129                     $object_user = array(
130                         "username" => $username,
131                         "password" => $password,
132                         "level" => $level,
133                     );
```

Gambar 5.45 Kode Program *Controller* Halaman Tambah Siswa

```
135                     $object_siswa = array(
136                         "nis" => $nis,
137                         "nama_siswa" => $nama_siswa,
138                         "jenkel" => $jenkel,
139                         "username" => $username,
140                     );
141
142                     $object_kls = array(
143                         "nis" => $nis,
144                         "id_kelas" => $kelas,
145                         "id_tahun" => $tahun,
146                     );
147
148                     $this->m_siswa->insert_user($object_user);
149                     $this->m_siswa->insert($object_siswa);
150                     $this->m_kelas->insert_kls($object_kls);
151
152                     $this->session->set_flashdata("Pesan", "<div class='alert alert-success' alert-dismissible' role='alert'><button type='button'
153                         class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span>
154                         </button>Data Berhasil ditambahkan</div>");
155                     redirect('Admin/siswa');
156
157                 }
158             }else{
159                 $a['message']="";
160                 $a['p']='Admin/siswa/v_addsiswa';
161                 $this->load->view('vwHeader',$a);
162             }
163     }
164
165     public function tampil(){
166         $this->load->library('form_validation');
167
168         $a['username'] = $this->session->userdata('username');
169         $a['p']='Admin/siswa/v_addsiswa';
170         $a['siswa'] = $this->m_siswa->ModelSiswa();
171         $a['kelas'] = $this->m_kelas->kelas_siswa();
172         $a['tahun'] = $this->m_siswa->getTahun()->result();
173         $a['message']=validation_errors();
174
175         $this->load->view('vwHeader',$a);
176     }
```

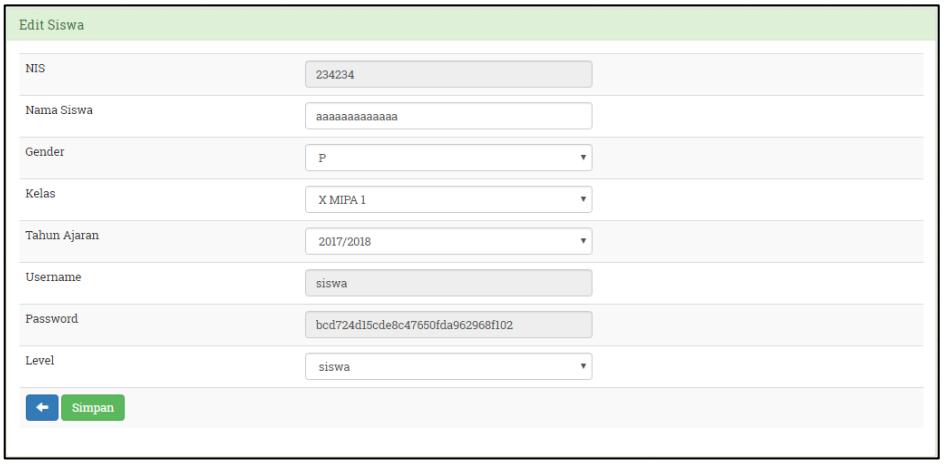
Gambar 5.46 Kode Program *Controller* Halaman Tambah Siswa(Lanjutan)



Gambar 5.47 Tampilan Halaman Tambah Data Siswa

3. Ubah Data Siswa

Setelah *admin* memilih salah satu data siswa, admin dapat memilih aksi *edit*. *Admin* dapat mengubah data siswa berdasarkan nis. Pada form edit sistem akan mengunci form nis sehingga nis siswa tidak dapat diubah. Berikut tampilan antarmuka halaman ubah data siswa dapat dilihat pada Gambar 5.48. Implementasi Kode program *Controller* ubah data siswa dapat dilihat pada Gambar 5.49 dan 5.50.



Gambar 5.48 Tampilan Halaman Ubah Data Siswa

```
34      public function edit($nis){
35
36          if($this->input->post('submit')){
37              $nis1 = $this->input->post('nis');
38              $nama_siswa = $this->input->post('nama_siswa');
39              $jenkel = $this->input->post('jenkel');
40              $kelas = $this->input->post('kelas');
41              $tahun = $this->input->post('tahun');
42              $username = $this->input->post('username');
43              $password = md5($this->input->post('password'));
44              $level = $this->input->post('level');
45
46              $object_user = array(
47                          "username" => $username,
48                          "password" => $password,
49                          "level" => $level,
50              );
51
52              $object_siswa = array(
53                          "nis" => $nis1,
54                          "nama_siswa" => $nama_siswa,
55                          "jenkel" => $jenkel,
56                          "username" => $username,
57              );
58
59              $object_kls = array(
60                          "nis" => $nis,
61                          "id_kelas" => $kelas,
62                          "id_tahun" => $tahun,
63              );
64
65          $this->db->where('username', $username);
66          $this->db->update('user',$object_user);
67
68          $this->db->where('nis', $nis);
69          $this->db->update('siswa',$object_siswa);
70
71          $this->db->where('nis', $nis);
72          $this->db->update('kelas_siswa',$object_kls);
73
74          $this->session->set_flashdata("Pesan", "<div class='alert alert-success' alert-dismissible' role='alert'>
75              <button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>
76              &times;</span></button>Data Berhasil disimpan</div>");
77          redirect('Admin/siswa');
78          }else{
```

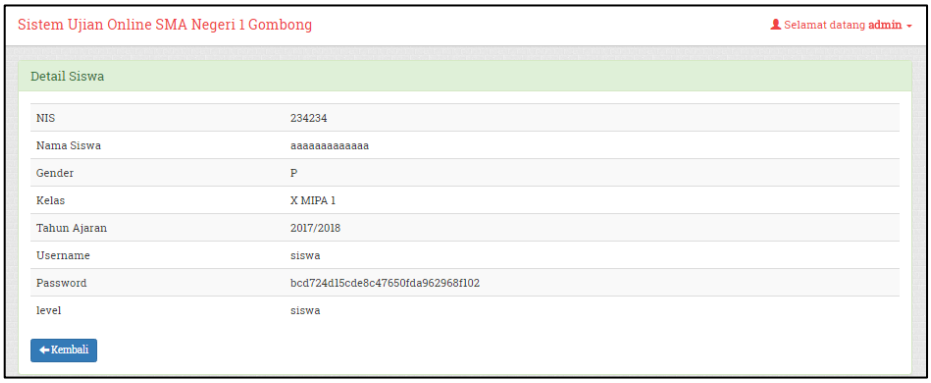
Gambar 5.49 Kode Program *Controller* Halaman Ubah Siswa

```
79
80              $a['username'] = $this->session->userdata('username');
81              $a['p']='Admin/siswa/v_editsiswa';
82              $a['editdata'] = $this->db->where('siswa.nis',$nis)->join('user','user.username=siswa.username')->join('kelas_siswa',
83              $a['kelas'] = $this->m_kelas->kelas_siswa();
84              $a['tahun'] = $this->m_siswa->getTahun()->result();
85              $this->load->view('vwHeader',$a);
86          }
87      }
```

Gambar 5.50 Kode Program *Controller* Halaman Ubah Siswa(Lanjutan)

4. Lihat Detail Data Siswa

Setelah admin memilih aksi lihat pada kolom tabel aksi, admin akan dialihkan menuju halaman detail data siswa. Sistem akan menampilkan NIS, nama siswa, Gender, Kelas, Tahun Ajaran, *Username*, *Password*, dan Level. Pada form detail data siswa, sistem hanya menampilkan data lengkap siswa sesuai nis dan tidak bisa diubah-ubah. Implementasi antarmuka halaman tambah data siswa dapat dilihat pada Gambar 5.51. Implementasi Kode program Controller detail data siswa dapat dilihat pada Gambar 5.52.



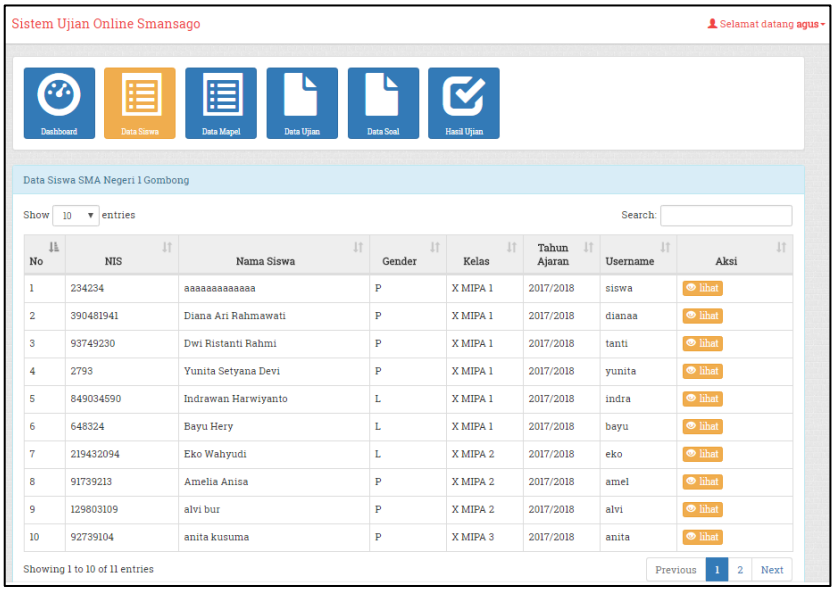
Gambar 5.51 Tampilan Halaman Lihat Detail Siswa

```
175     public function detail($nis) {
176         $this->load->library('form_validation');
177
178         $a['username'] = $this->session->userdata('username');
179         $a['p']='Admin/siswa/v_detailsiswa';
180         $a['siswa'] = $this->m_siswa->get_detail($nis)->row_array();
181
182         $this->load->view('vwHeader',$a);
183
184     }
```

Gambar 5.52 Kode Program Controller Halaman Lihat Detail Siswa

5.4.9 Halaman Data Siswa untuk Guru

Halaman data siswa untuk guru hampir sama dengan halaman siswa untuk admin. Perbedaannya terletak pada hak aksesnya. Guru hanya bisa melihat data siswa tidak dapat menambah, merubah, dan menghapus. Guru hanya bisa melakukan aksi lihat detail siswa. Implementasi antarmuka halaman data siswa untuk guru dapat dilihat pada Gambar 5.53.

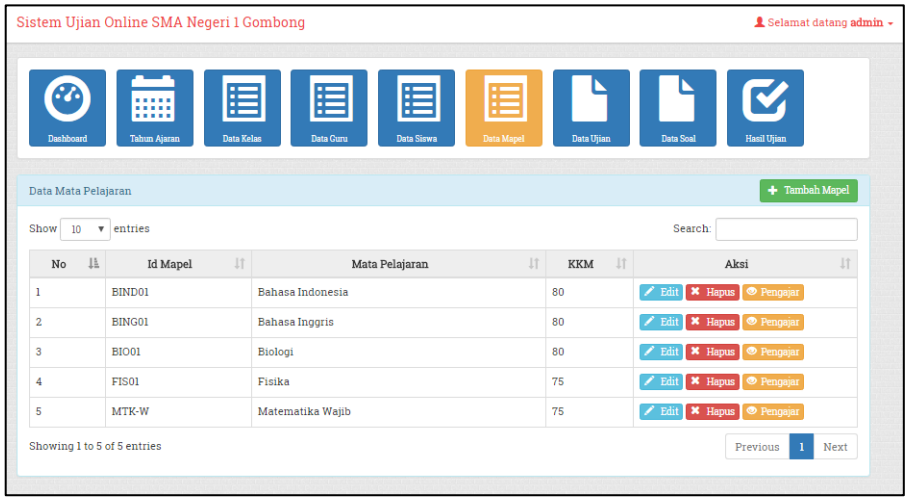


Gambar 5.53 Tampilan Halaman Data Siswa Untuk Guru

5.4.10 Halaman Data Mapel untuk Admin

1. Data Mapel

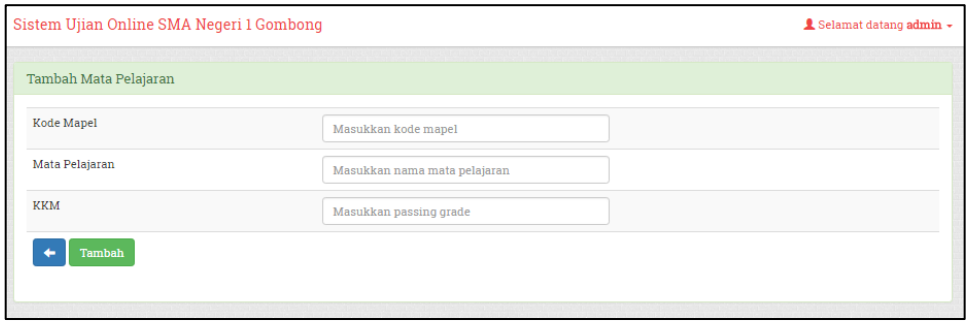
Halaman Data mapel menampilkan seluruh data mata pelajaran yang akan digunakan untuk pelaksanaan ujian. Setiap pelajaran memiliki kkm dan pengajar pada masing-masing kelas. Nilai kkm dapat dirubah sesuai dengan nilai yang ditetapkan oleh sekolah untuk setiap mata pelajaran. Implementasi halaman data mapel dapat dilihat pada Gambar 5.54.



Gambar 5.54 Tampilan Halaman Data Mapel

2. Tambah Data Mapel

Setelah admin klik button tambah pada bagian pojok kanan halaman, admin akan dialihkan pada halaman tambah mata pelajaran. Pada halaman tambah data mapel admin akan menginputkan kode mapel, nama mata pelajaran, dan kkm. Implementasi halaman tambah data mapel ditunjukan pada Gambar 5.55.



Gambar 5.55 Tampilan Halaman Tambah Data Mapel

Pada saat melakukan penambahan data mapel, maka sistem akan melakukan pengecekkan terhadap id mapel yang dimasukkan. Kode program pada kondisi ini dapat dilihat pada baris ke 69. Apabila id mapel ditemukan maka sistem akan menjalankan kondisi selanjutnya yaitu melakukan pemeriksaan apakah id mapel tersebut sudah dipakai. Jika ditemukan maka sistem akan memberikan pesan peringatan ‘Id mapel sudah terdaftar sebagai mata pelajaran’. Hal ini sebab tidak diperbolehkan terdapat data siswa dengan id mapel yang sama. Jika tidak maka sistem akan menginputkan data mapel baru dan sistem akan memberikan pesan ‘Data berhasil ditambahkan’, terlihat pada kode program baris ke 88. Kode program *Controller* tambah data mapel dapat dilihat pada Gambar 5.56.

```
61      public function add_mapel(){
62              $this->form_validation->set_rules('id_mapel','id_mapel', 'trim|required');
63              $this->form_validation->set_rules('nama_mapel','nama_mapel', 'trim|required');
64              $this->form_validation->set_rules('kkm','kkm', 'trim|required');
65
66              if($this->form_validation->run() === TRUE){
67                  $id_mapel = $this->input->post('id_mapel');
68                  $check=$this->m_mapel->check($id_mapel);
69                  if($check->num_rows()!=null){
70                      $a['username'] = $this->session->userdata('username');
71                      $a['message']="<div class='alert alert-danger' alert-dismissible' role='alert'><button type='button' class='close'
72                      data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span>
73                      </button>Id Mapel sudah terdaftar sebagai Mata Pelajaran</div>";
74                      $a['p']='Admin/mapel/v_addmapel';
75                      $this->load->view('vwHeader',$a);
76                  }else{
77                      $id_mapel = $this->input->post('id_mapel');
78                      $nama_mapel = $this->input->post('nama_mapel');
79                      $kkm = $this->input->post('kkm');
80
81                      $object_mapel = array(
82                          "id_mapel" => $id_mapel,
83                          "nama_mapel" => $nama_mapel,
84                          "kkm" => $kkm,
85                      );
86
87                      $this->m_mapel->insert($object_mapel);
88                      $this->session->set_flashdata("Pesan", "<div class='alert alert-success' alert-dismissible' role='alert'>
89                          <button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span>
90                          </button>Data Berhasil ditambahkan</div>");
91                      redirect('Admin/mapel');
92                  }
93              }else{
94                  $a['message']="";
95                  $a['p']='Admin/mapel/v_addmapel';
96                  $this->load->view('vwHeader',$a);
97              }
98      }
99
100     public function tampil_addmapel(){
101         $a['username'] = $this->session->userdata('username');
102         $a['p']='Admin/mapel/v_addmapel';
103         $a['mapel'] = $this->m_mapel->ModelMapel()->result_array();
104         $a['message']=validation_errors();
105         $this->load->view('vwHeader',$a);
106     }
```

Gambar 5.56 Kode Program Controller Halaman Tambah Mapel

3. Ubah Data Mapel

Setelah admin memilih aksi edit pada salah satu mata pelajaran, admin akan dialihkan menuju halaman edit mata pelajaran. Admin dapat mengubah data mapel berdasarkan kode mapel. Pada form edit sistem akan mengunci form kode mapel sehingga kode mapel tidak dapat diubah. Kode program *Controller* ubah data mapel dapat dilihat pada Gambar 5.57.

```
34      public function edit($id_mapel){
35
36          if($this->input->post('submit')){
37              $id_mapel1 = $this->input->post('id_mapel');
38              $nama_mapel = $this->input->post('nama_mapel');
39              $kkm = $this->input->post('kkm');
40
41              $object = array(
42                          "id_mapel" => $id_mapel1,
43                          "nama_mapel" => $nama_mapel,
44                          "kkm" => $kkm,
45              );
46
47          $this->db->where('id_mapel', $id_mapel);
48          $this->db->update('mapel',$object);
49          $this->session->set_flashdata("Pesan", "<div class='alert alert-success' alert-dismissible' role='alert'>
50              <button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span>
51              </button>Data Berhasil disimpan</div>");
52          redirect('Admin/mapel');
53          }else{
54              $a['username'] = $this->session->userdata('username');
55              $a['p']='Admin/mapel/v_editmapel';
56              $a['editdata'] = $this->db->where('id_mapel',$id_mapel)->get('mapel')->result_array();
57              $this->load->view('vwHeader',$a);
58          }
59      }
```

Gambar 5.57 Kode Program Controller Halaman Ubah Data Mapel

4. Lihat Pengajar Mapel

Setelah admin memilih aksi pengajar, maka akan dialihkan menuju halaman lihat pengajar sesuai dengan id mapel yang dipilih. Dihalaman ini akan ditampilkan pengajar mata pelajaran di semua kelas sesuai dengan id mapel yang dipilih. Berikut kode program *Controller* lihat pengajar mapel dapat dilihat pada Gambar 5.58.

```
187     public function detail($id_mapel) { //tampil halaman lihat pengajar atau detail mapel
188         $this->load->library('form_validation');
189
190         $a['username'] = $this->session->userdata('username');
191         $a['p']='Admin/mapel/v_detailmapel';
192         $a['pengajar'] = $this->m_mapel->get_detail($id_mapel);
193         $this->load->view('vwHeader',$a);
194     }
```

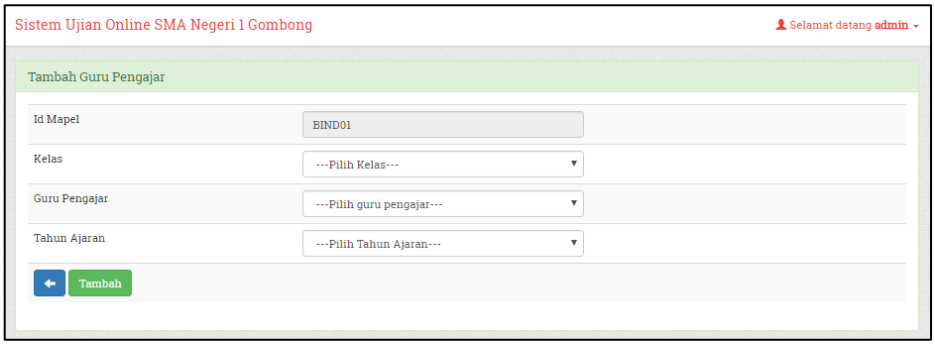
Gambar 5.58 Kode Program Controller Halaman Lihat Pengajar Mapel

5. Tambah Pengajar Mapel

Pada saat tambah pengajar, admin dapat mengubah kelas, nama pengajar, dan tahun ajaran. Admin tidak dapat mengganti dengan kode mapel lain, sehingga form input kode mapel hanya bisa dibaca. Berikut tampilan antarmuka halaman tambah pengajar mapel dapat dilihat pada Gambar 5.60. Implementasi Kode program *Controller* tambah pengajar mapel dapat dilihat pada Gambar 5.59.

```
108 ▼   public function add_pengajar(){
109             $this->form_validation->set_rules('nip','nip', 'trim|required');
110             $this->form_validation->set_rules('mapel','mapel', 'trim|required');
111             $this->form_validation->set_rules('kelas','kelas', 'trim|required');
112             $this->form_validation->set_rules('tahun','tahun', 'trim|required');
113
114 ▼           if($this->form_validation->run() === FALSE){
115                 $a['username'] = $this->session->userdata('username');
116                 $a['p']='Admin/mapel/v_detailmapel';
117                 $this->load->view('vwHeader',$a);
118 ▼           }else{
119                     $mapel = $this->input->post('mapel');
120                     $nip = $this->input->post('nip');
121                     $kelas = $this->input->post('kelas');
122                     $tahun = $this->input->post('tahun');
123
124 ▼                   $object_pengajar = array(
125                         "id_mapel" => $mapel,
126                         "nip" => $nip,
127                         "id_kelas" => $kelas,
128                         "id_tahun" => $tahun,
129 ▼           );
130                     $this->m_mapel->insert_pengajar($object_pengajar);
131
132 ▼                   $this->session->set_flashdata("Pesan", "<div class='alert alert-success' alert-dismissible' role='alert'>
133                         <button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>
134                         &times;</span></button>Data Berhasil ditambahkan</div>");
135                     $id_mapel=$this->agent->referrer();
136                     redirect($id_mapel);
137             }
138     }
139
140 ▼   public function tampil_addpengajar($id_mapel){
141         $this->load->library('form_validation');
142
143         $a['username'] = $this->session->userdata('username');
144         $a['p']='Admin/mapel/v_addpengajar';
145         $a['pengajar'] = $this->m_mapel->get_detail($id_mapel);
146         $a['guru'] = $this->m_mapel->getGuru()->result();
147         $a['kelas'] = $this->m_kelas->kelas_siswa();
148         $a['tahun'] = $this->m_mapel->getTahun()->result();
149
150         $this->load->view('vwHeader',$a);
151
152     }
```

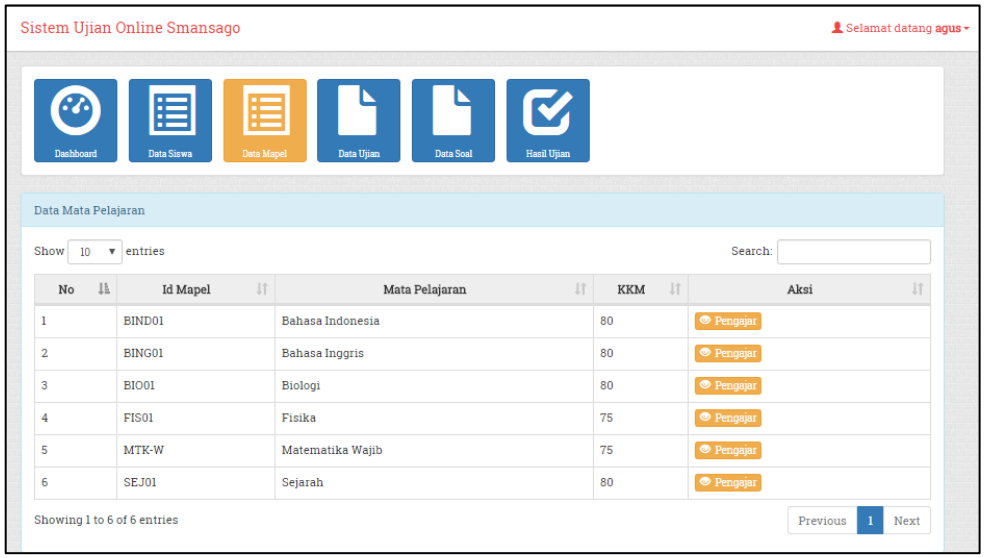
Gambar 5.59 Kode Program Controller Halaman Tambah Pengajar Mapel



Gambar 5.60 Tampilan Halaman Tambah Pengajar Mapel

5.4.11 Halaman Data Mapel Untuk Guru

Halaman data mapel untuk guru hampir sama dengan halaman mapel untuk admin. Perbedaannya terletak pada hak aksesnya. Guru hanya bisa melihat data mapel dan data pengajar untuk setiap mapel, tidak dapat menambah, merubah, dan menghapus. Implementasi antarmuka halaman data siswa untuk guru dapat dilihat pada Gambar 5.61.

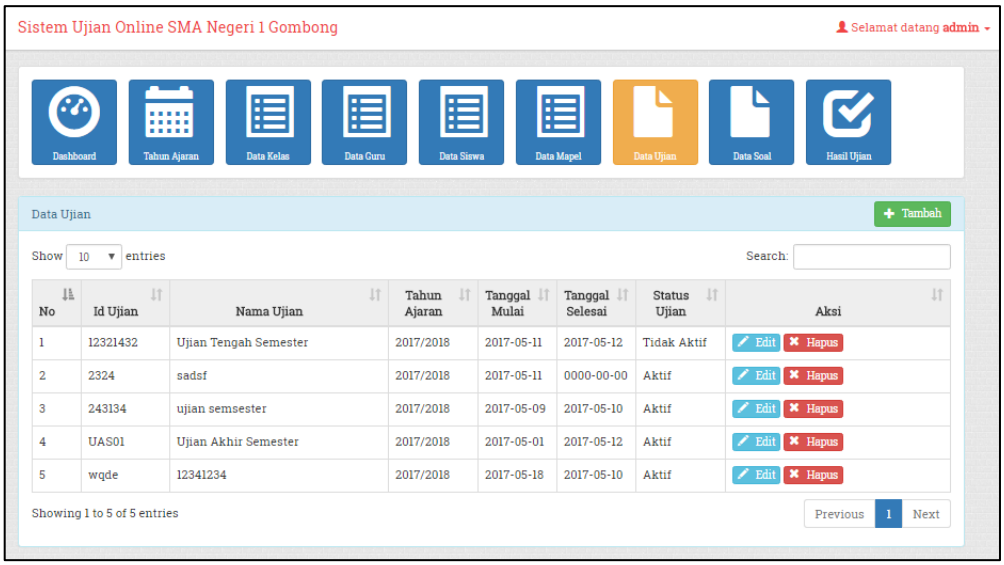


Gambar 5.61 Tampilan Halaman Data Mapel untuk Guru

5.4.12 Halaman Data Ujian untuk Admin

1. Data Ujian

Pada halaman ujian ini meneampilkan data tabel ujian yaitu jadwal dilaksanakannya ujian yang berelasi dengan tabel soal. Admin dapat melakukukan aksi edit dan hapus pada data yang telah ditambahkan. Implemetasi antarmuka halaman informasi data ujian dapat dilihat pada Gambar 5.62.



Gambar 5.62 Tampilan Halaman Data Ujian

2. Tambah Data Ujian

Sama seperti halaman tambah mapel, pada saat melakukan penambahan data di menu ujian, sistem akan melakukan pengecekkan terhadap id ujian yang dimasukkan. Apabila id ujian ditemukan maka sistem akan menjalankan kondisi selanjutnya yaitu melakukan pemeriksaan apakah id ujian tersebut sudah dipakai. Jika ditemukan maka sistem akan memberikan pesan peringatan ‘Id Ujian sudah terdaftar. Silahkan ganti dengan Id yang lain !’. Hal ini sebab tidak diperbolehkan terdapat data ujian dengan id ujian yang sama. Jika tidak maka sistem akan menginputkan data ujian baru dan sistem akan memberikan pesan ‘Data berhasil ditambahkan’. Kode program *Controller* tambah data ujian dapat dilihat pada Gambar 5.63 dan 5.64.

```
68      public function add(){
69              $this->form_validation->set_rules('id_ujian','id_ujian', 'trim|required');
70              $this->form_validation->set_rules('nama_ujian','nama_ujian', 'trim|required');
71              $this->form_validation->set_rules('tahun','tahun', 'trim|required');
72              $this->form_validation->set_rules('tgl_mulai','tgl_mulai', 'required');
73              $this->form_validation->set_rules('tgl_selesai','tgl_selesai', 'required');
74              //$this->form_validation->set_rules('status_ujian','status_ujian', 'trim|required');
75
76              if($this->form_validation->run() === TRUE){
77                  $id_ujian = $this->input->post('id_ujian');
78                  $check=$this->m_ujian->check($id_ujian);
79                  if($check->num_rows()!=null){
80                      $a['username'] = $this->session->userdata('username');
81                      $a['message']="<div class='alert alert-danger' alert-dismissible' role='alert'>
82                      <button type='button' class='close' data-dismiss='alert' aria-label='Close'>
83                      <span aria-hidden='true'>&times;</span></button>Id Ujian sudah terdaftar. Silahkan ganti dengan Id yang lain !</div>";
84                      $a['p']='Admin/ujian/v_addujian';
85                      $this->load->view('vwHeader',$a);
86                  }else{
87                      $id_ujian = $this->input->post('id_ujian');
88                      $nama_ujian = $this->input->post('nama_ujian');
89                      $tahun = $this->input->post('tahun');
90                      $tgl_mulai = $this->input->post('tgl_mulai');
91                      $tgl_selesai = $this->input->post('tgl_selesai');
92
93                          $object = array(
94                              "id_ujian" => $id_ujian,
95                              "nama_ujian" => $nama_ujian,
96                              "id_tahun" => $tahun,
97                              "tgl_mulai" => $tgl_mulai,
98                              "tgl_selesai" => $tgl_selesai,
99                  );
100
101                     $this->m_ujian->insert($object);
102                     $this->session->set_flashdata("Pesan", "<div class='alert alert-success' alert-dismissible' role='alert'>
103                         <button type='button' class='close' data-dismiss='alert' aria-label='Close'>
104                         <span aria-hidden='true'>&times;</span></button>Data Berhasil ditambahkan</div>");
105                     redirect('Admin/ujian');
106                 }
107             }else{
108                 $a['message']="";
109                 $a['p']='Admin/ujian/v_addujian';
110                 $this->load->view('vwHeader',$a);
111             }
112     }
```

Gambar 5.63 Kode Program *Controller* Halaman Data Ujian

```
114     public function tampil(){
115         $this->load->library('form_validation');
116
117         $a['username'] = $this->session->userdata('username');
118         $a['tahun'] = $this->m_ujian->getTahun()->result();
119         $a['ujian'] = $this->m_ujian->ModelUjian()->result_array();
120         $a['message']=validation_errors();
121         $a['p']='Admin/ujian/v_addujian';
122         $this->load->view('vwHeader',$a);
123         // print_r($a);
124     }
```

Gambar 5.64 Kode Program *Controller* Halaman Data Ujian (Lanjutan)

3. Ubah Data Ujian

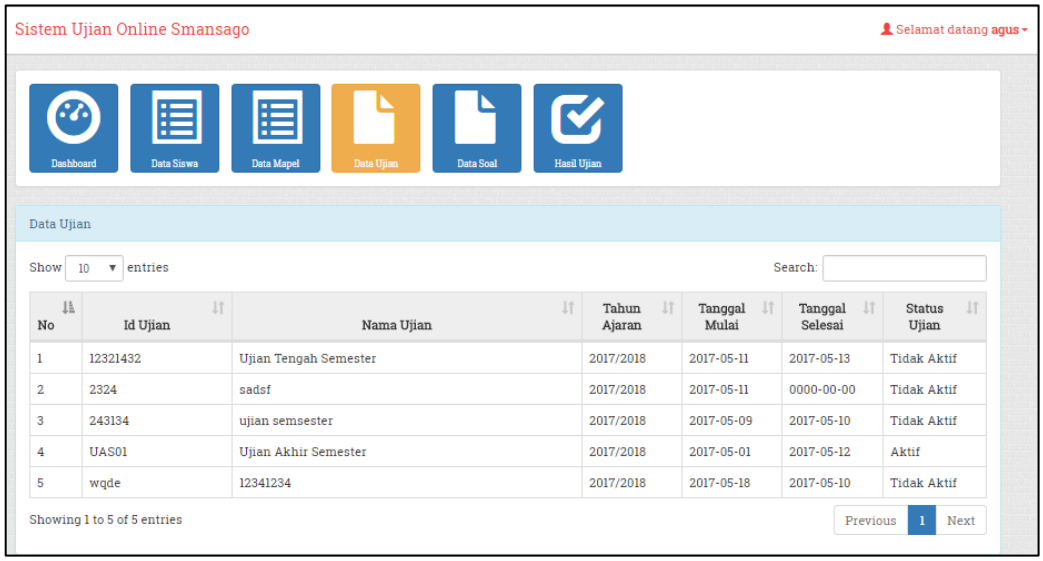
Setelah admin memilih aksi edit pada salah satu data ujian, admin akan dialihkan menuju halaman edit ujian. Admin dapat mengubah data ujian berdasarkan kode ujian. Pada form edit sistem akan mengunci form kode ujian sehingga kode ujian tidak dapat diubah. Kode program *Controller* ubah data ujian dapat dilihat pada Gambar 5.65.

```
33      public function edit($id_ujian){
34
35          if($this->input->post('submit')){
36              $id_ujian1 = $this->input->post('id_ujian');
37              $nama_ujian = $this->input->post('nama_ujian');
38              $tahun = $this->input->post('tahun');
39              $tgl_mulai = $this->input->post('tgl_mulai');
40              $tgl_selesai = $this->input->post('tgl_selesai');
41              $status_ujian = $this->input->post('status_ujian');
42
43              $object = array(
44                          "id_ujian" => $id_ujian1,
45                          "nama_ujian" => $nama_ujian,
46                          "id_tahun" => $tahun,
47                          "tgl_mulai" => $tgl_mulai,
48                          "tgl_selesai" => $tgl_selesai,
49                          "status_ujian" => $status_ujian,
50              );
51
52          $this->db->where('id_ujian', $id_ujian);
53          $this->db->update('ujian',$object);
54          $this->session->set_flashdata("Pesan", "<div class='alert alert-success' alert-dismissible' role='alert'>
55              <button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;
56              </span></button>Data Berhasil disimpan</div>");
57          redirect('Admin/ujian');
58          }else{
59
60              $a['username'] = $this->session->userdata('username');
61              $a['p']='Admin/ujian/v_editujian';
62              $a['editdata'] = $this->db->where('id_ujian',$id_ujian)->get('ujian')->result_array();
63              $a['tahun'] = $this->m_ujian->getTahun()->result();
64
65              $this->load->view('vwHeader',$a);
66          }
67      }
```

Gambar 5.65 Kode Program *Controller* Halaman Ubah Data UJian

5.4.13 Halaman Data Ujian untuk Guru

Halaman data ujian untuk guru hampir sama dengan halaman ujian untuk admin. Perbedaannya terletak pada hak aksesnya. Guru hanya bisa melihat data ujian tidak dapat menambah, merubah, dan menghapus. Implementasi antarmuka halaman data ujian untuk guru dapat dilihat pada Gambar 5.66.

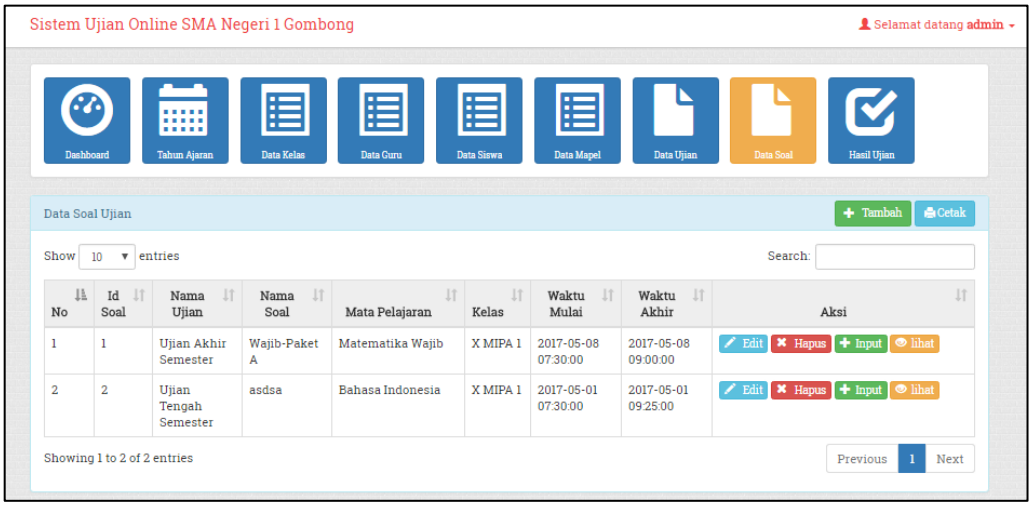


Gambar 5.66 Tampilan Halaman Data Ujian untuk Guru

5.4.14 Halaman Data Soal

1. Data Soal

Setelah memilih menu data soal, admin akan dialihkan menuju halaman data soal. Disini sistem akan menampilkan data soal yang tersimpan dan terdaftar pada basis data soal dalam bentuk tabel. Tabel yang digunakan pada data kelas menggunakan datatables sehingga kelas dapat diurutkan sesuai kolom yang dipilih. Berikut implementasi tampilan halaman data soal dapat dilihat pada Gambar 5.67.



Gambar 5.67 Tampilan Halaman Data Soal

2. Tambah Soal

Setelah klik button tambah yang berada dibagian kanan atas halaman, admin akan dialihkan menuju halaman tambah soal. Disini admin dapat menginputkan Id ujian, nama soal, mata pelajaran, kelas, waktu mulai dan waktu akhir. Apabila data berhasil ditambahkan akan tampil alert 'Data berhasil ditambahkan'. Adapun potongan kode program halaman tambah data kelas dapat dilihat pada Gambar 5.68.

```
77      public function add(){
78              $this->form_validation->set_rules('nama_soal','nama_soal', 'trim|required');
79              $this->form_validation->set_rules('mapel','mapel', 'trim|required');
80              $this->form_validation->set_rules('kelas','kelas', 'trim|required');
81              $this->form_validation->set_rules('ujian','ujian', 'trim|required');
82              $this->form_validation->set_rules('waktu_mulai','waktu_mulai', 'trim|required');
83              $this->form_validation->set_rules('waktu_akhir','waktu_akhir', 'trim|required');
84
85              if($this->form_validation->run() === FALSE){
86                  $a['p']='Admin/soal/v_addsoal';
87                  $this->load->view('vwHeader',$a);
88              }else{
89                      $nama_soal = $this->input->post('nama_soal');
90                      $mapel = $this->input->post('mapel');
91                      $kelas = $this->input->post('kelas');
92                      $ujian = $this->input->post('ujian');
93                      $waktu_mulai = $this->input->post('waktu_mulai');
94                      $waktu_akhir = $this->input->post('waktu_akhir');
95
96                      $object = array(
97                          "nama_soal" => $nama_soal,
98                          "id_mapel" => $mapel,
99                          "id_kelas" => $kelas,
100                         "id_ujian" => $ujian,
101                         "waktu_mulai" => $waktu_mulai,
102                         "waktu_akhir" => $waktu_akhir,
103
104             );
105                     $this->m_soal->insert($object);
106
107                     $this->session->set_flashdata("Pesan", "<div class='alert alert-success' alert-dismissible' role='alert'>
108                         <button type='button' class='close' data-dismiss='alert' aria-label='Close'>
109                         <span aria-hidden='true'>&times;</span></button>Data Berhasil ditambahkan</div>");
110                     redirect('Admin/soal');
111             }
112     }
113
114     public function tampil(){
115         $this->load->library('form_validation');
116         $a['username'] = $this->session->userdata('username');
117         $a['p']='Admin/soal/v_addsoal';
118         $a['soal'] = $this->m_soal->ModelSoal();
119         $a['mapel'] = $this->m_soal->getMapel()->result();
120         $a['ujian'] = $this->m_soal->getIdujian()->result();
121         $a['kelas'] = $this->m_kelas->kelas_siswa();
122         $this->load->view('vwHeader',$a);
123     }
```

Gambar 5.68 Kode Program *Controller* Halaman Tambah Data Soal

3. Ubah Soal

Setelah admin memilih aksi edit pada salah satu data soal, admin akan dialihkan menuju halaman ubah soal. Admin dapat mengubah data soal berdasarkan id soal. Kode program *Controller* ubah data soal dapat dilihat pada Gambar 5.69.

```
34      public function edit($id_soal){
35
36          if($this->input->post('submit')){
37                  $id_soal1 = $this->input->post('id_soal');
38                  $nama_soal = $this->input->post('nama_soal');
39                  $ujian = $this->input->post('ujian');
40                  $mapel = $this->input->post('mapel');
41                  $kelas = $this->input->post('kelas');
42                  $waktu_mulai = $this->input->post('waktu_mulai');
43                  $waktu_akhir = $this->input->post('waktu_akhir');
44                  $jml_pertanyaan = $this->input->post('jml_pertanyaan');
45
46              $object = array(
47                      "id_soal" => $id_soal1,
48                      "nama_soal" => $nama_soal,
49                      "id_ujian" => $ujian,
50                      "id_mapel" => $mapel,
51                      "id_kelas" => $kelas,
52                      "waktu_mulai" => $waktu_mulai,
53                      "waktu_akhir" => $waktu_akhir,
54                      "jml_pertanyaan" => $jml_pertanyaan,
55              );
56
57          $this->db->where('id_soal', $id_soal);
58          $this->db->update('soal',$object);
59          $this->session->set_flashdata("Pesan", "<div class='alert alert-success' alert-dismissible' role='alert'>
60              <button type='button' class='close' data-dismiss='alert' aria-label='Close'>
61              <span aria-hidden='true'>&times;</span></button>Data Berhasil disimpan</div>");
62          redirect('Admin/soal');
63          }else{
64
65              $a['username'] = $this->session->userdata('username');
66              $a['p']='Admin/soal/v_editsoal';
67              $a['editdata'] = $this->db->where('id_soal', $id_soal)->get('soal')->result_array();
68              $a['mapel'] = $this->m_soal->getMapel()->result();
69              $a['ujian'] = $this->m_soal->getIdujian()->result();
70              $a['kelas'] = $this->m_kelas->kelas_siswa();
71
72              $this->load->view('vwHeader',$a);
73          }
74      }
```

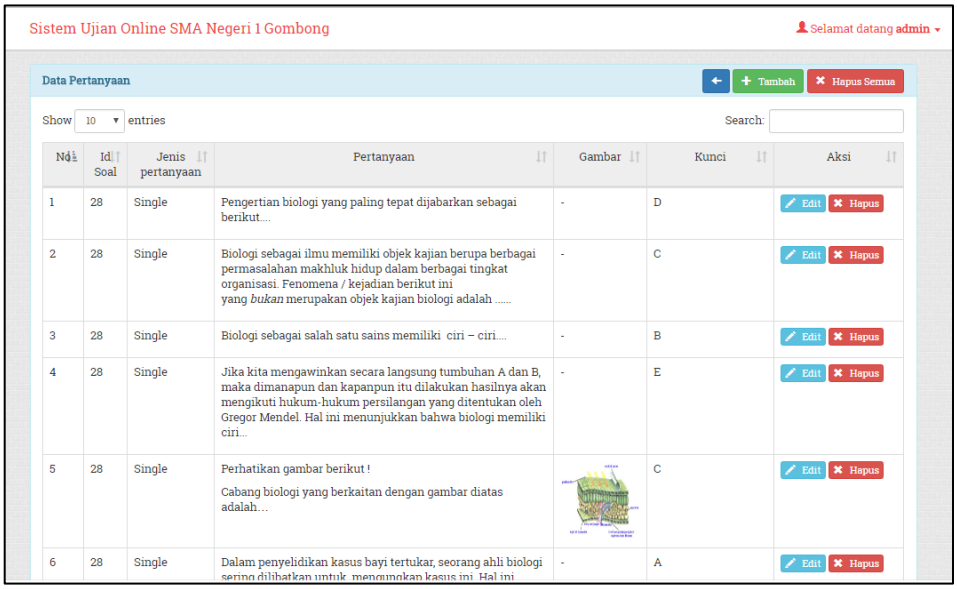
Gambar 5.69 Kode Program *Controller* Halaman Ubah Data Soal

4. Daftar Pertanyaan

Untuk melihat daftar pertanyaan, admin memilih button aksi input. Aksi input ini digunakan sebagai halaman untuk menginputkan pertanyaan pada masing-masing soal ujian berdasarkan id soal. Pada halaman ini ditampilkan id soal, jenis pertanyaan, pertanyaan, gambar, dan kunci jawaban. Berikut implementasi tampilan halaman daftar pertanyaan dapat dilihat pada Gambar 5.71. Kode program *Controller* ubah data soal dapat dilihat pada Gambar 5.70.

```
4   class pertanyaan extends CI_Controller{
5
6           public function __construct()
7       {
8           parent::__construct();
9           $this->load->library('form_validation');
10          $this->load->model('m_pertanyaan');
11
12      }
13
14      public function index(){
15          $a['username'] = $this->session->userdata('username');
16          $a['p']='Admin/soal/v_input';
17          $a['pertanyaan'] = $this->m_pertanyaan->ModelPertanyaan()->result_array();
18          $this->load->view('vwHeader',$a);
19          }
```

Gambar 5.70 Kode Program *Controller* Halaman Daftar Pertanyaan



Gambar 5.71 Halaman Daftar Pertanyaan

5. Tambah Pertanyaan

Halaman tambah pertanyaan ini digunakan untuk menambahkan pertanyaan pada setiap masing-masing soal. Pada halaman ini ditampilkan cek editor untuk menginputkan teks pertanyaan dan opsi jawaban. Terdapat *button file choose* untuk menambahkan gambar pada teks pertanyaan. Berikut kode program *Controller* tambah pertanyaan dapat dilihat pada Gambar 5.72 dan 5.73.

```
44      public function input($id_soal){
45              $this->form_validation->set_rules('id_soal','id_soal', 'trim|required');
46              $this->form_validation->set_rules('jenis_pertanyaan','jenis_pertanyaan', 'trim|required');
47              $this->form_validation->set_rules('pertanyaan','pertanyaan', 'trim|required');
48              $this->form_validation->set_rules('gambar','gambar', 'trim|');
49              $this->form_validation->set_rules('jawaban','jawaban', 'trim|required');
50
51              if($this->form_validation->run() === FALSE){
52                  $a['p']='Admin/soal/v_inputpertanyaan';
53                  $a['id_soal'] = $id_soal;
54                  $a['data'] = $this->input->post();
55                  $this->load->view('vwHeader',$a);
56              }else{
57
58                      $config['upload_path']          = 'assets/img/pertanyaan';
59                      $config['allowed_types']        = 'gif|jpg|png';
60                      $config['max_size']             = 2000;
61                      $config['max_width']            = 1024;
62                      $config['max_height']           = 1024;
63
64                      $this->upload->initialize($config);
65
66                      if ( ! $this->upload->do_upload('gambar')){
67                          // $error = array('error' => $this->upload->display_errors());
68                          // $this->load->view('v_upload', $error);
69                          $gambar="";
70                      }else{
71                          // $data = array('upload_data' => $this->upload->data());
72                          // $this->load->view('v_upload_sukses', $data);
73                          $gambar=$this->upload->file_name;
74                      }
75
76                      $id_soal = $this->input->post('id_soal');
77                      $id_pertanyaan = $this->input->post('id_pertanyaan');
78                      $jenis_pertanyaan = $this->input->post('jenis_pertanyaan');
79                      $pertanyaan = $this->input->post('pertanyaan');
80                      //$gambar = $this->upload->data();
81                      // $gambar = $this->input->post('gambar[]');
82                      // $gambar->$this->upload->file_name;
83                      $opsi = $this->input->post('opsi[]');
84                      $isi_opsi = $this->input->post('isi_opsi[]');
85                      $jawaban = $this->input->post('jawaban');
```

Gambar 5.72 Kode Program *Controller* Halaman Tambah Pertanyaan

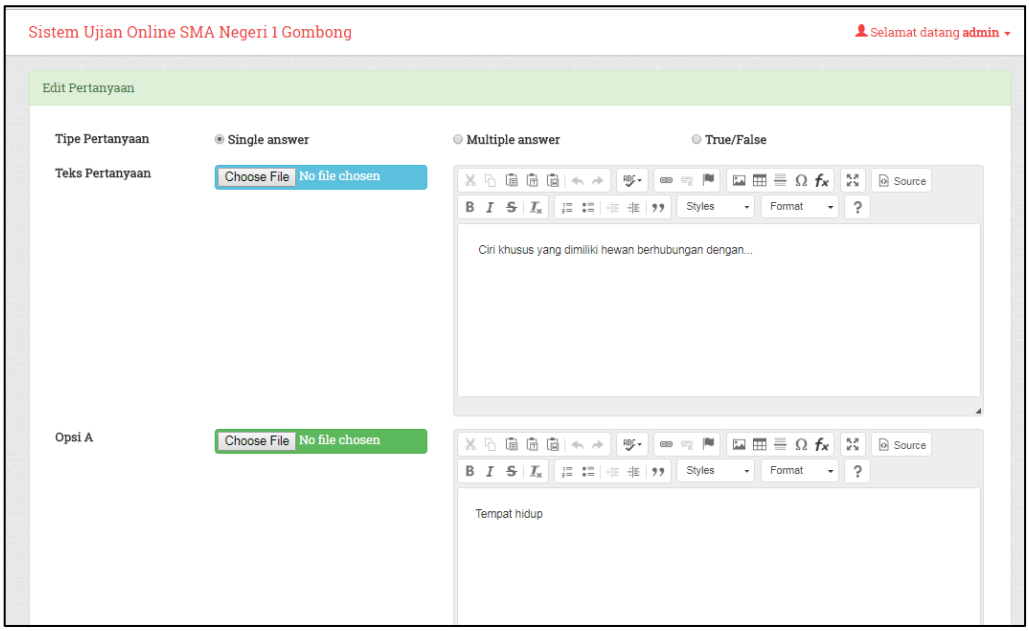
```
89                      $object = array(
90                          'id_soal' => $id_soal,
91                          'jenis_pertanyaan' => $jenis_pertanyaan,
92                          'pertanyaan' => $pertanyaan,
93                          'gambar' => $gambar
94                      );
95
96                      $id_pertanyaan = $this->m_pertanyaan->insert($object);
97                      $i = 0;
98                      foreach ($isi_opsi as $key => $row) {
99
100                         $object_opsi = array(
101                         "id_pertanyaan" => $id_pertanyaan,
102                         "opsi" => $opsi[$i],
103                         "isi_opsi" => $row,
104                         "jawaban" => ($jawaban==$i++)?1:0,
105                         );
106
107                         $this->m_pertanyaan->insert_opsi($object_opsi);
108                     }
109
110                     $this->session->set_flashdata("Pesan", "<div class='alert alert-success' alert-dismissible' role='alert'>
111                         <button type='button' class='close' data-dismiss='alert' aria-label='Close'>
112                         <span aria-hidden='true'>&times;</span></button>Data Berhasil ditambahkan</div>");
113                     redirect('Admin/pertanyaan');
114             }
115     }
116
117     public function tampil($id_soal){
118         $this->load->library('form_validation');
119
120         $a['username'] = $this->session->userdata('username');
121         $a['p']='Admin/soal/v_inputpertanyaan';
122         $a['pertanyaan'] = $this->m_pertanyaan->get_input($id_soal);
123         $a['id_soal'] = $id_soal;
124
125         $this->load->view('vwHeader',$a);
126     }
```

Gambar 5.73 Kode Program *Controller* Halaman Tambah Pertanyaan

6. Ubah Pertanyaan

Untuk merubah data pertanyaan, pengguna akan dibawa ke halaman ubah pertanyaan. Perubahan data pertanyaan dapat dilakukan dengan memilih tautan

aksi edit. Implemetasi antarmuka halaman ubah pertanyaan dapat dilihat pada Gambar 5.74.



Gambar 5.74 Halaman Ubah Pertanyaan

7. Lihat Detail Soal dan Pertanyaan

Pada halaman ini menampilkan detail soal dan pertanyaan yang kemudian dapat dicetak oleh pengelola. Pada halaman ini terdapat menu cetak yang digunakan untuk mengexpor halaman detail soal dan pertanyaan menjadi lembar soal dengan format pdf. Berikut kode program *Controller* lihat detail dan pertanyaan dapat dilihat pada Gambar 5.75.

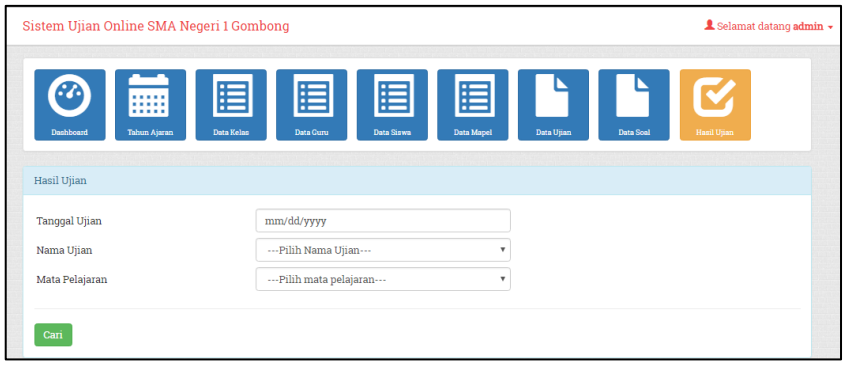
```
170
171     public function detail($id_soal) {
172         $this->load->library('form_validation');
173
174         $a['username'] = $this->session->userdata('username');
175         $a['p']='Admin/soal/v_detailsoal';
176         $a['soal'] = $this->m_soal->get_detail($id_soal)->row_array();
177         $a['pertanyaan'] = $this->m_pertanyaan->tampil_soal($id_soal);
178         $a['kelas'] = $this->m_kelas->kelas_siswa();
179         $a['id_soal']    = $id_soal;
180
181         $this->load->view('vwHeader',$a);
182     }
```

Gambar 5.75 Kode Program *Controller* Halaman Lihat Detail Soal

5.4.15 Halaman Hasil Ujian

1. Data Hasil Ujian

Pada halaman hasil ujian ini menampilkan form tanggal ujian, nama ujian, dan mata pelajaran yang digunakan untuk mencari hasil ujian berdasarkan data tersebut. Admin dan guru juga dapat melakukan cetak hasil ujian. Setelah klik cari pengelola diarahkan menuju ke halaman cetak data hasil ujian. Implemetasi antarmuka halaman informasi data hasil ujian dapat dilihat pada Gambar 5.76. Kode program *Controller* ubah data soal dapat dilihat pada Gambar 5.77.



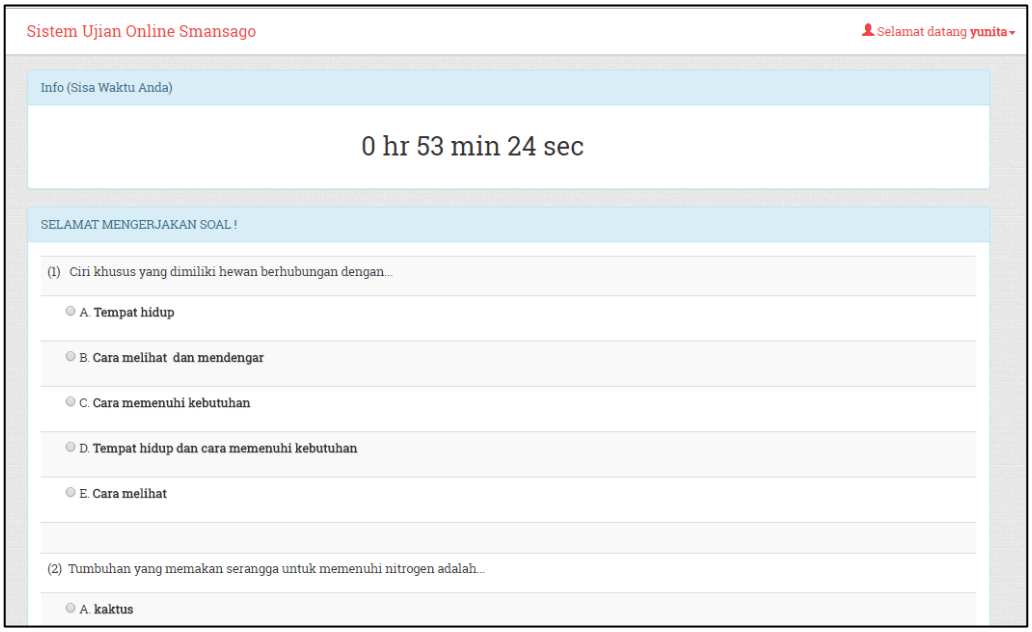
Gambar 5.76 Halaman Hasil Ujian

```
4  class Hasil extends CI_Controller{
5
6          public function __construct()
7      {
8          parent::__construct();
9          $this->load->library('form_validation');
10         $this->load->model('m_hasil');
11     }
12
13     public function index(){
14         $a['username'] = $this->session->userdata('username');
15
16             $a['p']='Admin/ujian/v_hasilujian';
17             $a['mapel'] = $this->m_hasil->mapel();
18             $a['ujian'] = $this->m_hasil->ujian();
19
20             $this->load->view('vwHeader',$a);
21     }
22
23     public function carihasil(){
24         $a['tgl'] = $this->input->post('tgl_ujian');
25         $a['ujian'] = $this->input->post('ujian');
26         $a['mapel'] = $this->input->post('mapel');
27
28         $a['username'] = $this->session->userdata('username');
29         $a['p']='Admin/ujian/v_carihasil';
30         $a['hasil'] = $this->m_hasil->search($a['tgl'],$a['ujian'],$a['mapel']);
31         $this->load->view('vwHeader',$a);
32
33     }
```

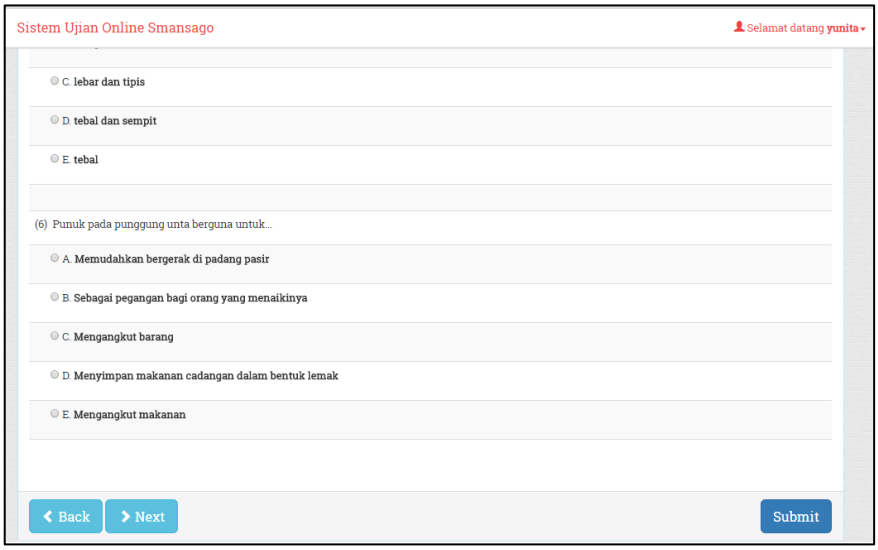
Gambar 5.77 Kode Program *Controller* Halaman Hasil Ujian

5.4.16 Halaman Soal Ujian Siswa

Halaman Soal Ujian Siswa ini adalah halaman yang digunakan oleh user siswa. Pada halaman ini siswa mengerjakan soal sesuai dengan waktu mundur yang ditampilkan di halaman. Setelah siswa klik button mulai ujian pada dashboard siswa baru siswa dapat mengerjakan soal. Terdapat button submit digunakan apabila siswa telah selesai melaksanakan ujian. Adapun tampilan halaman soal ujian siswa dapat dilihat pada Gambar 5.78 dan 5.79. Kode program *Controller* halaman soal ujian siswa dapat dilihat pada Gambar 5.80.



Gambar 5.78 Tampilan Halaman Soal Ujian Siswa

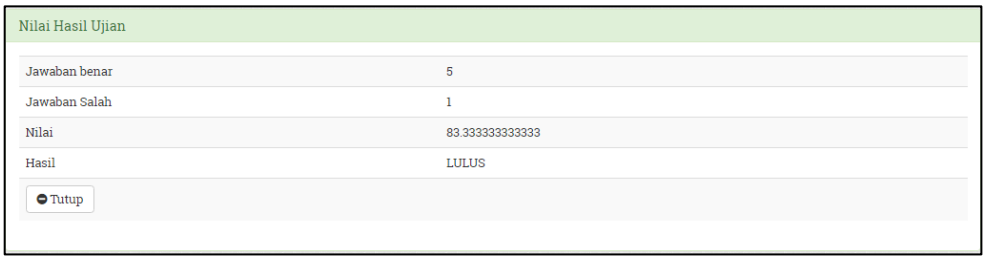


Gambar 5.79 Tampilan Halaman Soal Ujian Siswa (Lanjutan)

```
15      public function index($id_soal)
16      {
17          $a['username']   = $this->session->userdata('username');
18          $a['p']          = 'Siswa/v_tampilsoal';
19          $a['pertanyaan'] = $this->m_pertanyaan->tampil_soal($id_soal);
20          $a['id_soal']    = $id_soal;
21
22          $waktu = $this->m_pertanyaan->tampil_jam($id_soal)->row_array();
23
24          $timestamp2 = strtotime($waktu['waktu_akhir']);
25
26          $a['waktu'] = date("Y-m-d H:m", $timestamp2);
27
28          $this->load->view('Siswa/template_siswa', $a);
29          // print_r($a);
30
31      }
```

Gambar 5.80 Kode Program *Controller* Halaman Soal Ujian Siswa

Setelah submit, siswa akan dialihkan menuju halaman nilai ujian. Di halaman ujian siswa ini, siswa dapat melihat jawaban benar, jawaban salah, nilai, dan hasil lulus atau tidaknya. Adapun halaman nilai ujian dapat dilihat pada Gambar 5.81. Kode program *Controller* halaman nilai ujian dapat dilihat pada Gambar 5.82.



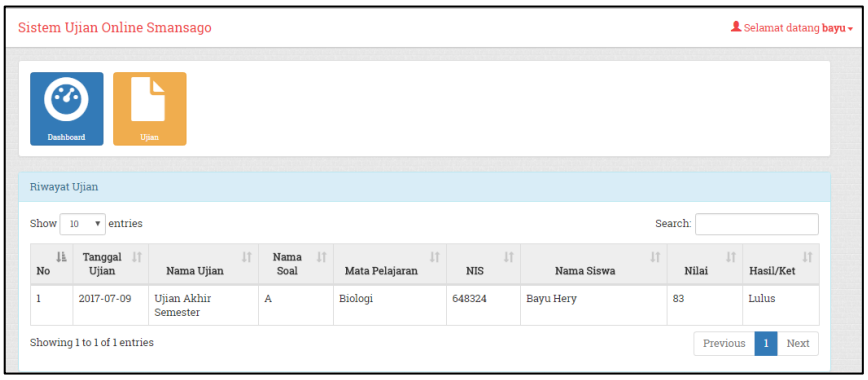
Gambar 5.81 Tampilan Halaman Nilai Ujian

```
33      public function nilai()
34      {
35          $id = $this->input->post('id_soal');
36
37          $kkmm = $this->m_hasil->getKkm($id);
38
39          foreach ($kkmm as $key) {
40              $kkm = $key->kkm;
41          }
42
43          if ($this->input->post('opsi') == null) {
44              $a['benar'] = 0;
45          } else {
46              $a['benar'] = $this->m_pertanyaan->get_nilai($this->input->post('opsi'));
47          }
48          $a['salah']    = $this->input->post('jml_soal') - $a['benar'];
49          $a['nilai']    = $a['benar'] * (100 / $this->input->post('jml_soal'));
50          $a['username'] = $this->session->userdata('username');
51
52          if ($a['nilai']  >= $kkm) {
53              $keterangan = "LULUS";
54          }elseif ($a['nilai']  < $kkm) {
55              $keterangan = "TIDAK LULUS";
56          }
57
58          $object = [
59              'id_soal'     => $this->input->post('id_soal'),
60              'nis'         => $this->session->userdata('nis'),
61              'nilai_ujian' => $a['nilai'],
62              'keterangan'  => $keterangan,
63          ];
64
65          $this->m_hasil->insert($object);
66
67          $a['ket'] = $keterangan;
68
69          $a['p'] = "Siswa/v_nilai";
70          $this->load->view('Siswa/template_siswa', $a);
71      }
72  }
```

Gambar 5.82 Kode Program *Controller* Halaman Nilai Ujian

5.4.17 Halaman Riwayat Ujian Siswa

Pada halaman riwayat ujian, siswa dapat melihat semua hasil nilai dari riwayat ujian yang telah dilaksanakan sebelumnya. Adapun tampilan halaman riwayat ujian dapat dilihat pada Gambar 5.83.



Gambar 5.83 Halaman Nilai Ujian

BAB VI
PENGUJIAN DAN PEMBAHASAN SISTEM

Tahap pengujian dan pembahasan adalah tahap dimana sistem diuji kelayakannya. Tahap ini bertujuan untuk mengavaluasi sistem, memeriksanya adanya kesalahan dan memastikan seluruh sistem berfungsi. Selain itu, tahap ini juga memastikan apakah sistem dapat digunakan dengan optimal. Pengujian sistem pada penelitian ini berfokus pada pengujian fungsional sistem. Pengujian sistem dibagi menjadi tiga bagian, yaotu skenario pengujian sistem, uji kasus hasil pengujian sistem dan pengujian pengguna.

5.1 Skenario Pengujian Sistem

Skenario pengujian merupakan proses yang menjelaskan urutan atau langkah-langkah dan hal-hal yang menjadi materi pengujian pada sistem ujian online berbasis web. Adapun skenario pengujian yang akan dilakukan dapat dilihat pada Tabel 6.1.

Tabel 6.1 Skenario Pengujian Sistem

Pengujian	Materi	Hasil
Melakukan proses *login*	Mengisi *username* dan *password*	Sistem dapat menampilkan halaman utama jika berhasil
Pengelolaan data tahun	Isi data masukkan tahun ajaran	Sistem dapat menambah, merubah(mengaktif dan nonaktifkan), dan menghapus tahun ajaran.
Pengelolaan data kelas siswa	Isi data masukan kelas	Sistem dapat menambah dan menghapus data kelas
Pengelolaan data guru	Isi data masukan guru	Sistem dapat menambah, merubah, menghapus, dan melihat detail data guru
Pengelolaan data siswa	Isi data masukan siswa	Sistem dapat menambah, merubah, menghapus, dan melihat detail data siswa
Pengelolaan data mapel	Isi data masukkan mapel, Isi data masukkan	Sistem dapat menambah, mengubah, menghapus

	pengajar	mapel serta melihat detail pengajar, tambah pengajar, edit, dan hapus pengajar.
Pengelolaan data ujian	Isi data masukan ujian	Sistem dapat menambah, mengubah, dan menghapus data ujian
Pengelolaan data soal	Isi masukkan data soal, Isi masukkan data pertanyaan	Sistem dapat menambah. merubah, menghapus data soal serta menambah, merubah, dan ,menghapus data pertanyaan.
Cetak soal dan pertanyaan dalam bentuk pdf	Memilih aksi cetak	Sistem dapat mengekspor soal menjadi bentuk pdf.
Pelaksanaan ujian siswa	Memilih data ujian di dashboard siswa, klik tombol mulai ujian	Sistem dapat menampilkan waktu mundur dan soal ujian, serta menampilkan hasil ujian setelah klik tombol *submit*.
Pengelolaan hasil ujian	Cari berdasarkan tanggal, nama ujian, dan mata pelajaran	Sistem dapat menampilkan hasil ujian siswa
Cetak hasil ujian dalam bentuk pdf	Memilih aksi cetak	Sistem dapat mengekspor hasil ujian menjadi bentuk pdf.
Pengelolaan *password* pengguna	Mengisi *password* lama dan *password* baru	Sistem dapat mengubah password lama menjadi password baru.

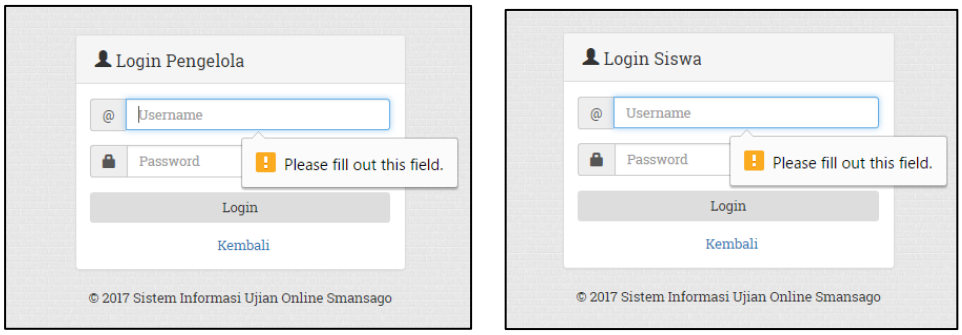
5.2 Hasil Pengujian Sistem

Hasil pengujian sistem merupakan tahap setelah rencana pengujian yang disusun pada skenario pengujian. Pengujian ini berfokus pada fungsional sistem, yaitu hanya memperhatikan masukan ke dalam sistem dan keluaran dari masukan tersebut. Pengujian ini akan menggambarkan uji kasus dan hasil pengujian masukkan data yang benar dan data yang salah.

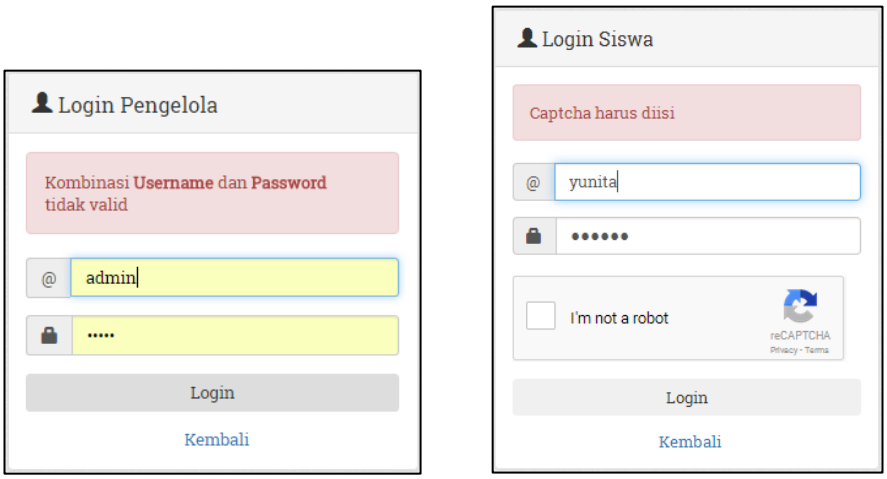
5.2.1 Proses Login

Proses login bertujuan untuk menyaring pengguna yang dapat masuk ke sistem. Pengguna yang bisa masuk ke sistem adalah pengguna yang sudah

terdaftar dalam basis data sistem. Dengan melakukkan login ke dalam sistem maka sudah berlaku aturan yang berjalan di sistem ini. Yakni pengguna dibedakan berdasarkan hak aksesnya masing-masing dan mendapatkan fitur dan layanan yang disesuaikan dengan hak akses tersebut. Apabila pengguna berhasil masuk ke sistem maka sistem akan menampilkan menu halaman utama. Namun jika gagal masuk ke sistem akan kembali menampilkan halaman login dengan memberikan pesan peringatan jika username dan password yang dimasukkan tidak sesuai. Apabila pengguna belum mengisi *reCAPTCHA* maka akan muncul peringatan "Captcha harus diisi". Tampilan halaman login dapat dilihat pada Gambar 6.1 dan tampilan pemberitahuan saat login gagal dapat dilihat pada Gambar 6.2.



Gambar 6.1 Halaman Login

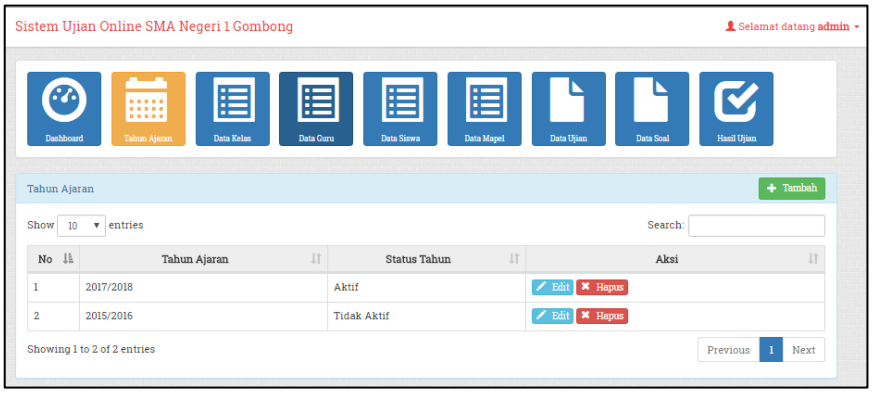


Gambar 6.2 Pemberitahuan *Login* Gagal

5.2.2 Pengolahan Data Tahun Ajaran

1. Halaman Data Tahun Ajaran

Halaman data tahun ajaran berupa daftar tabel yang berisi tahun ajaran pelaksanaan Ujian di SMA Negeri 1 Gombong. Pada halaman ini terdapat tautan tambah, ubah dan hapus tahun ajaran. Masing-masing tautan dan tombol ini berperan dalam proses pengelolaan data tahun ajaran. Halaman data tahun ajaran terdapat pada Gambar 6.3.

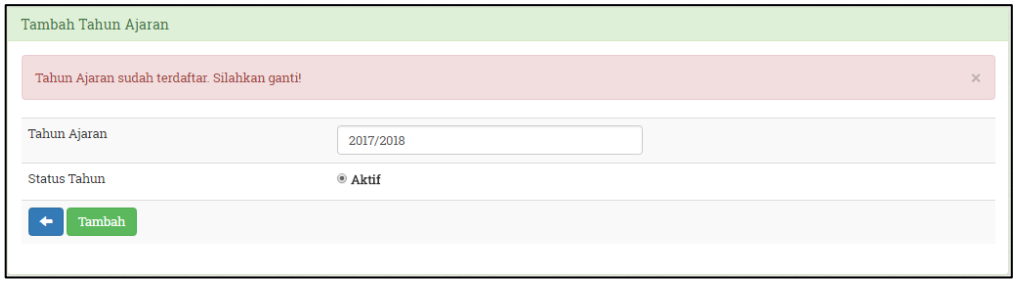


Gambar 6.3 Halaman Data Tahun Ajaran

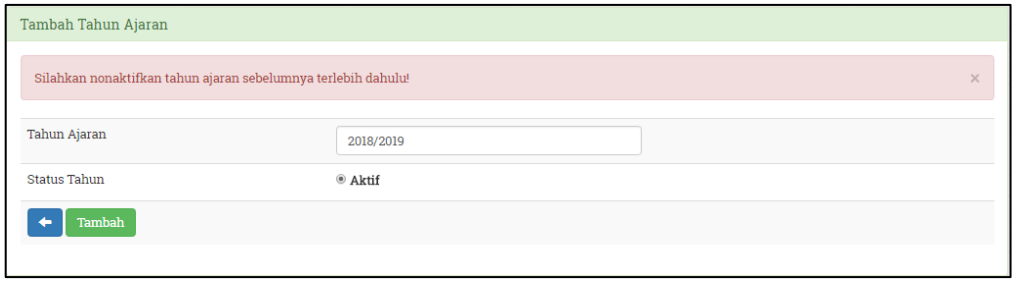
2. Halaman Tambah Tahun Ajaran

Halaman tambah tahun ajaran digunakan untuk menambahkan data tahun ajaran baru yang merupakan tahun pelaksanaan ujian. Pada halaman ini admin menginputkan tahun ajaran dan otomatis tahun ajaran sudah aktif. Adapun halaman tambah tahun ajaran terdapat pada Gambar 6.4. Apabila menginputkan dengan tahun ajaran yang sudah ada sebelumnya maka akan muncul peringatan " Tahun Ajaran sudah terdaftar". Adapun peringatan tahun ajaran sudah terdaftar dapat dilihat pada Gambar 6.5. Kemudian apabila saat menginputkan pengguna lupa belum menonaktifkan tahun ajaran, maka akan muncul peringatan " Silahkan nonaktifkan tahun ajaran sebelumnya terlebih dahulu". Adapun gambar peringatan untuk menonaktifkan tahun ajaran dapat dilihat pada Gambar 6.6.

Gambar 6.4 Halaman Tambah Tahun Ajaran



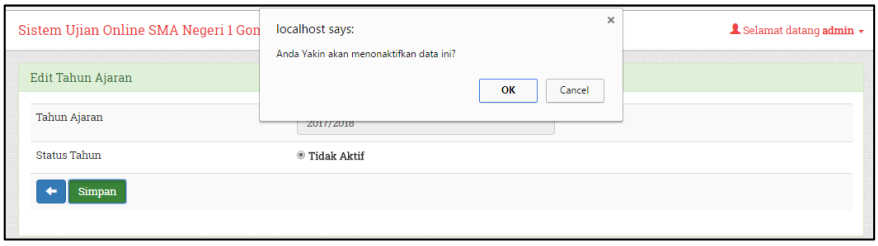
Gambar 6.5 Peringatan Tahun Ajaran Sudah Terdaftar



Gambar 6.6 Peringatan untuk Menonaktifkan Tahun Ajaran Sebelumnya

3. Halaman Ubah Tahun

Halaman ubah tahun ajaran digunakan untuk menonaktifkan tahun ajaran yang sudah berlalu. Pada halaman ini admin dialihkan menuju halaman ubah tahun ajaran, kemudian klik simpan. Sistem akan menampilkan peringatan ‘Anda yakin akan menonaktifkan data ini ?’ untuk mengkonfirmasi ulang aktifitas yang dilakukan pengguna. Apabila klik OK maka data akan tersimpan dengan status tidak aktif. Adapun peringatan menonkatifkan tahun ajaran terdapat pada Gambar 6.7.



Gambar 6.7 Peringatan Konfirmasi Menonaktifkan Tahun Ajaran

4. Hapus Tahun Ajaran

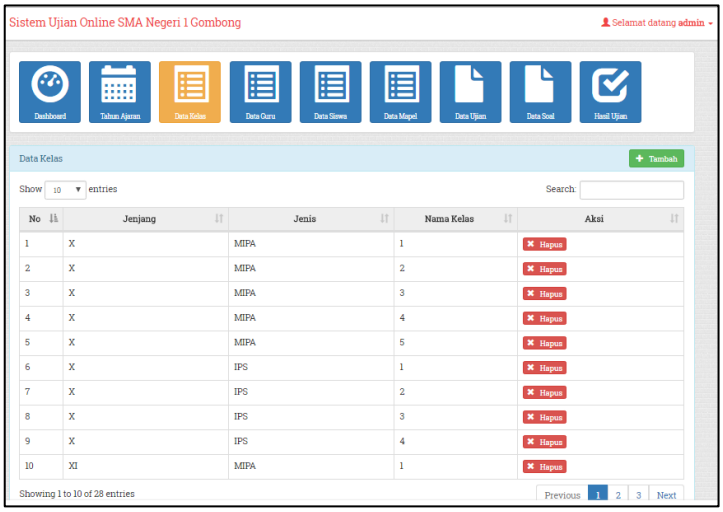
Pada saat hapus data tahun ajaran, pengguna tidak dapat menghapus tahun ajaran yang sudah digunakan sebelumnya. Akan muncul peringatan "Tahun Ajaran yang telah digunakan tidak dapat dihapus!". Hal ini dikarenakan didalamnya terdapat data-data yang saling terkait dengan ujian yang dilaksanakan pada tahun ajaran tersebut misalnya data paket soal dan hasil ujian. Peringatan Tahun Ajaran tidak dapat dihapus dapat dilihat pada Gambar 6.8.

Gambar 6.8 Peringatan Tahun Ajaran Tidak Dapat Dihapus

5.2.3 Pengolahan Data Kelas

1. Halaman Data Kelas

Halaman data kelas berupa daftar tabel seluruh data kelas yang terdaftar di dalam sistem. Pada halaman ini terdapat tautan tambah dan hapus kelas. Masing-masing tautan dan tombol ini berperan dalam proses pengelolaan data kelas. Halaman data kelas terdapat pada Gambar 6.9.



Gambar 6.9 Halaman Data Kelas

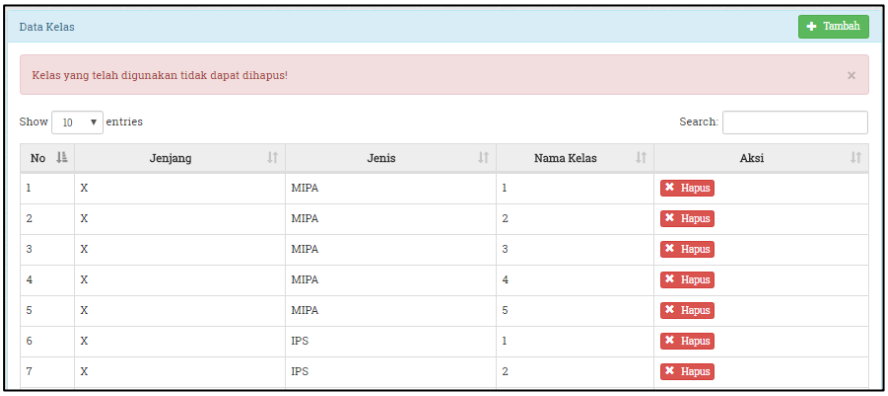
2. Halaman Tambah Kelas

Halaman tambah kelas digunakan untuk menambahkan data kelas baru yang akan disertakan dalam pelaksanakan ujian. Pada halaman ini admin menginputkan jenjang, jenis kelas, dan nama kelas. Adapaun halaman tambah kelas terdapat pada Gambar 6.10.

Gambar 6.10 Halaman Data Kelas

3. Hapus Kelas

Pada saat hapus kelas, pengguna tidak dapat menghapus kelas yang sudah digunakan sebelumnya. Akan muncul peringatan “Kelas yang telah digunakan tidak dapat dihapus!”. Hal ini dikarenakan didalamnya terdapat data-data yang saling terkait dengan siswa yang terdaftar pada kelas tersebut. Peringatan Kelas tidak dapat dihapus dapat dilihat pada Gambar 6.11.

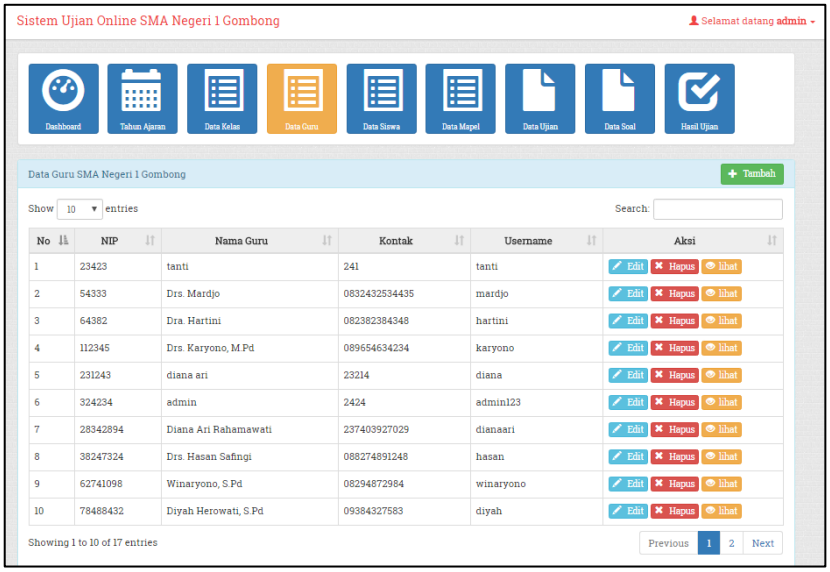


Gambar 6.11 Peringatan Kelas Tidak Dapat Dihapus

5.2.4 Pengolahan Data Guru

1. Halaman Data Guru

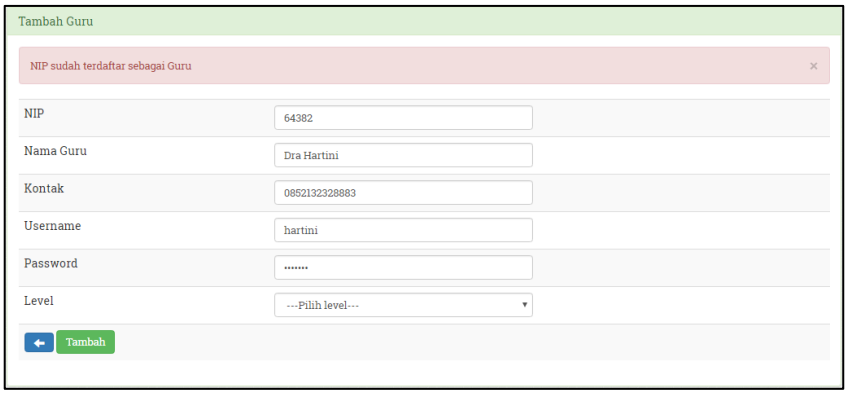
Halaman utama data guru berupa daftar tabel seluruh data guru yang terdaftar di dalam sistem. Pada halaman ini terdapat tautan lihat data guru, tambah data guru, ubah data guru dan hapus data guru dan lihat detail guru. Masing-masing tautan dan tombol ini berperan dalam proses pengelolaan data guru. Terdapat pula kolom pencarian yang dapat digunakan pengguna untuk mencari data guru berdasarkan kata kunci tertentu. Halaman utama data guru terdapat pada Gambar 6.12.



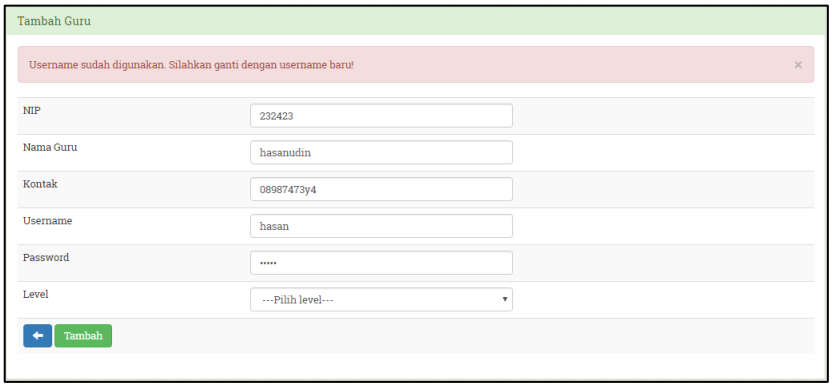
Gambar 6.12 Halaman Data Guru

2. Halaman Tambah Guru

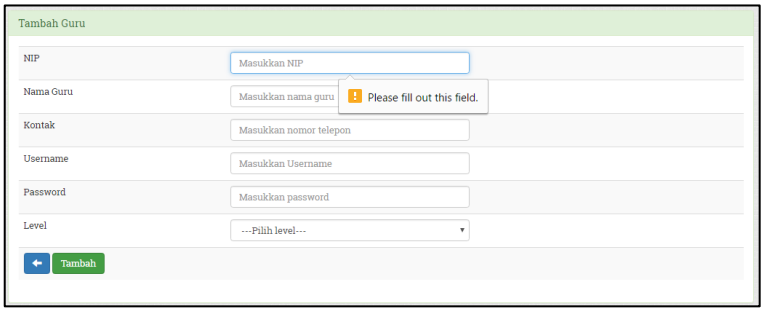
Halaman tambah guru digunakan untuk menambahkan data guru yang mengajar di SMA Negeri 1 Gombong. Pada saat proses penginputan data guru maka sistem akan memeriksa nip guru dan *username* . Apabila sistem menemukan nip yang sama dalam basis data maka sistem akan menampilkan pemberitahuan "Nip sudah terdaftar sebagai guru". Apabila sistem menemukan *username* yang sama dalam basis data maka sistem akan menampilkan pemberitahuan "*Username* sudah digunakan. Silahkan ganti dengan *username* baru!". Jika nip ataupun username tidak ditemukan sistem secara otomatis menyimpan data guru baru. Pemberitahuan Nip sudah terdaftar sebagai guru dapat dilihat pada Gambar 6.13. Pemberitahuan *username* sudah digunakan dapat dilihat pada Gambar 6.14. Setelah itu sistem akan memeriksa apakah semua form telah diisi, jika belum maka akan muncul peringatan seperti pada Gambar 6.15.



Gambar 6.13 Pemberitahuan Nip Sudah Terdaftar Sebagai Guru



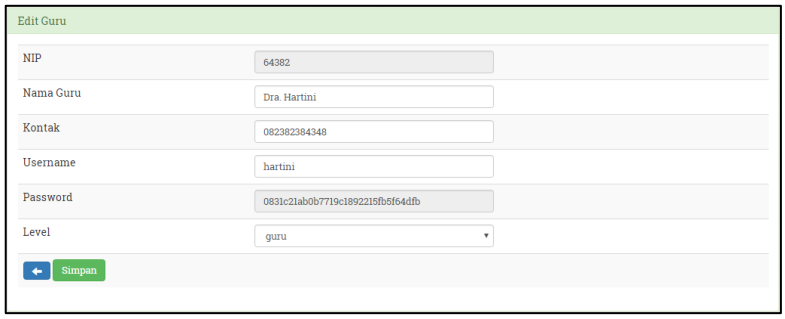
Gambar 6.14 Pemberitahuan Username Sudah Digunakan



Gambar 6.15 Pemberitahuan Kolom Masih Kosong

3. Halaman Ubah Guru

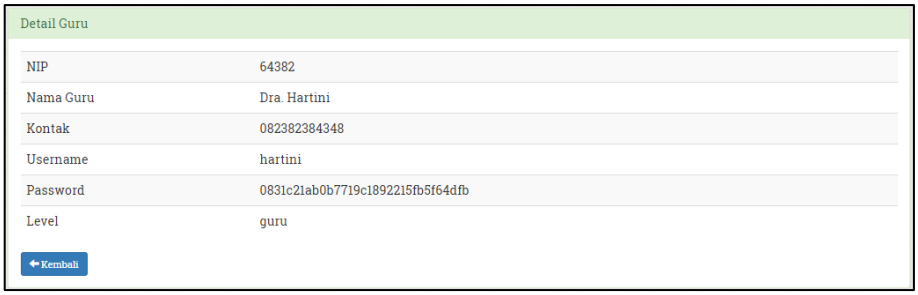
Untuk merubah data guru, pengguna akan dibawa ke halaman ubah data guru. Perubahan data guru dapat dilakukan dengan memilih tautan aksi edit. Adapun halaman ubah data guru ini dapat dilihat pada Gambar 6.16.



Gambar 6.16 Halaman Ubah Guru

4. Halaman Detail Guru

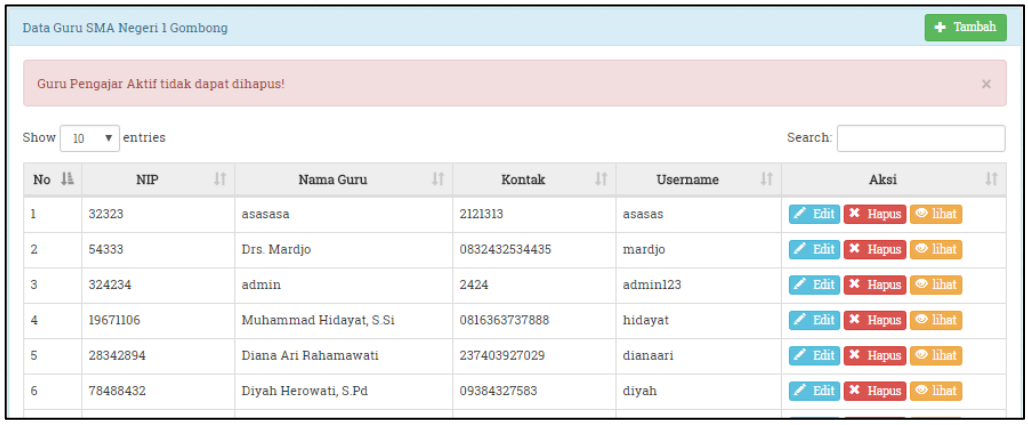
Untuk melihat detail guru, pengguna akan dibawa ke halaman detail guru. Melihat detail guru dapat dilakukan dengan memilih tautan aksi lihat detail. Adapun halaman detail guru ini dapat dilihat pada Gambar 6.17.



Gambar 6.17 Halaman Detail Guru

5. Hapus Guru

Pada saat hapus data guru, pengguna tidak dapat menghapus guru yang sudah terdaftar sebagai pengajar mata pelajaran. Akan muncul peringatan "Guru Pengajar Aktif tidak dapat dihapus!". Hal ini dikarenakan didalamnya terdapat data-data yang saling terkait dengan Guru pengajar mata pelajaran. Peringatan guru tidak dapat dihapus dapat dilihat pada Gambar 6.18.

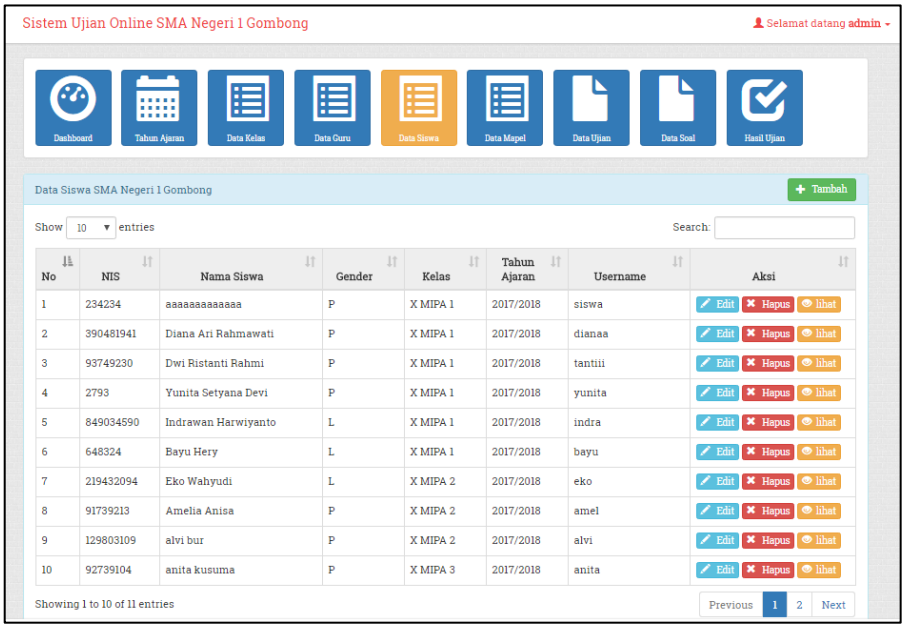


Gambar 6.18 Peringatan guru tidak dapat dihapus

5.2.5 Pengolahan Data Siswa

1. Halaman Data Siswa

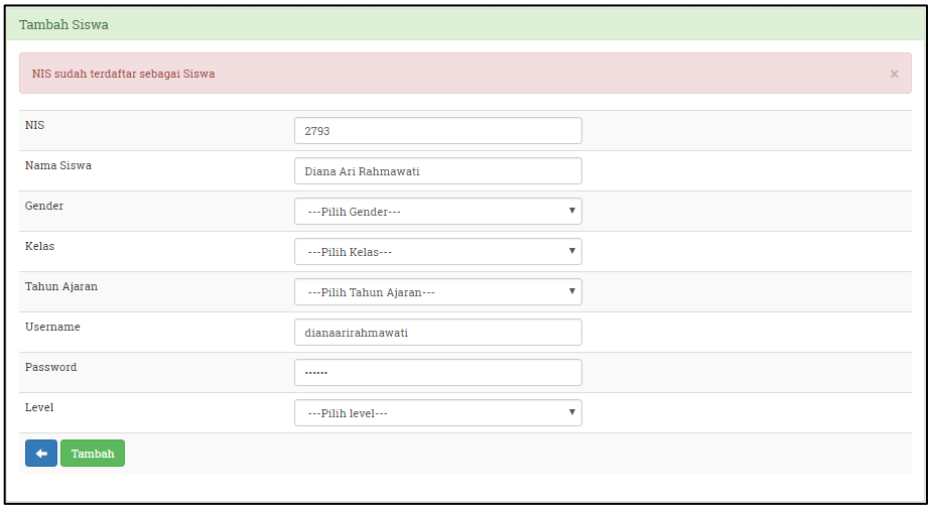
Halaman utama data siswa berupa daftar tabel seluruh data siswa yang terdaftar di dalam sistem. Pada halaman ini terdapat tautan lihat data siswa, tambah data siswa, ubah data siswa dan hapus data siswa. Masing-masing tautan dan tombol ini berperan dalam proses pengelolaan data siswa. Terdapat pula kolom pencarian yang dapat digunakan pengguna untuk mencari data siswa berdasarkan kata kunci tertentu. Untuk kemudian sistem melakukan pencarian siswa melalui kata kunci tersebut. Selain itu, terdapat pula kolom inputan *combo box* untuk menampilkan data siswa dalam berapa baris halaman. Nilai 10 pada kolom *combo box* memperlihatkan data siswa dilihat berdarkan tiap sepuluh baris data. Adapun gambar halaman utama data siswa terdapat pada Gambar 6.19.



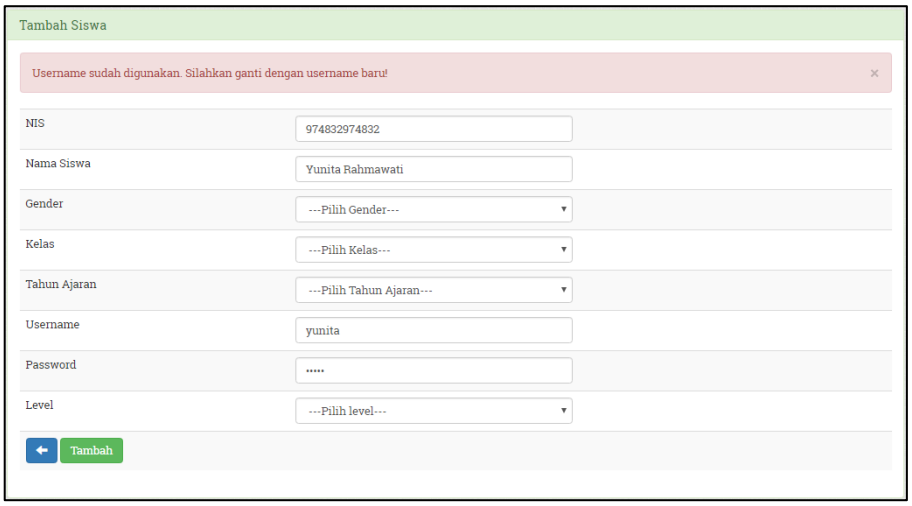
Gambar 6.19 Halaman Data Siswa

2. Halaman Tambah Siswa

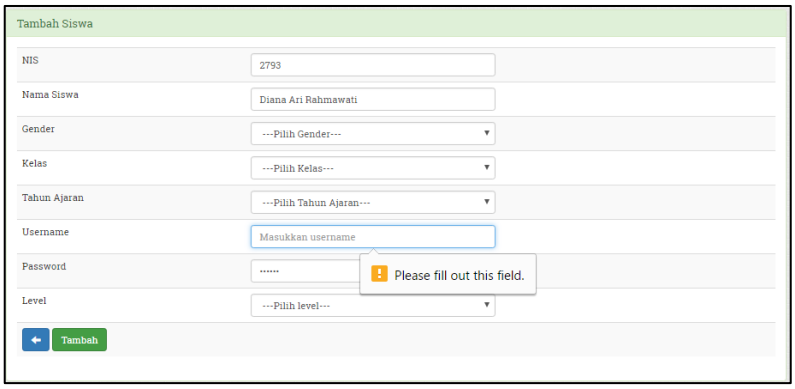
Pada saat pemasukan data siswa maka akan dilakukan pengecekan terhadap nis siswa. Data tidak sah apabila pengguna memasukkan data siswa dengan nis ataupun *username* yang telah ada sebelumnya. Apabila nis sudah ada sistem akan memunculkan pesan peringatan seperti pada Gambar 6.20. Apabila username sudah ada sistem akan memunculkan pesan peringatan seperti pada Gambar 6.21. Setelah itu sistem akan memeriksa apakah semua form telah diisi, jika belum maka akan muncul peringatan seperti pada Gambar 6.22.



Gambar 6.20 Pemberitahuan Nis Sudah Terdaftar Sebagai Siswa



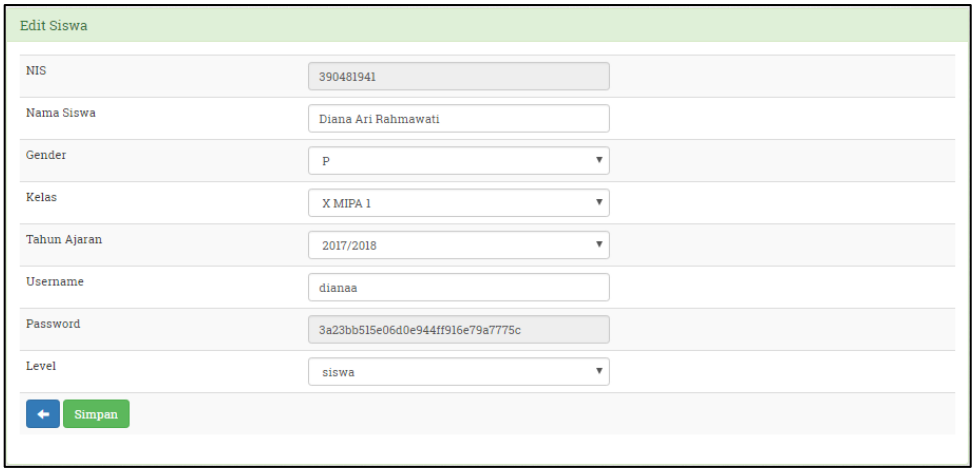
Gambar 6.21 Pemberitahuan Username Sudah Digunakan



Gambar 6.22 Pemberitahuan Kolom Masih Kosong

3. Halaman Ubah Siswa

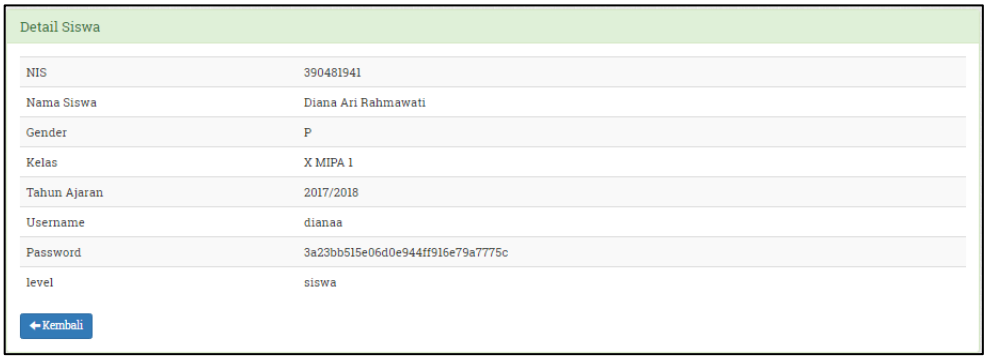
Untuk merubah data siswa, pengguna akan dibawa ke halaman ubah data siswa. Perubahan data siswa dapat dilakukan dengan memilih tautan aksi *edit*. Adapun halaman ubah data siswa ini dapat dilihat pada Gambar 6.23.



Gambar 6.23 Halaman Ubah Data Siswa

4. Halaman Detail Siswa

Untuk melihat detail siswa, pengguna akan dibawa ke halaman detail siswa. Melihat detail siswa dapat dilakukan dengan memilih tautan aksi lihat detail. Adapun halaman detail siswa ini dapat dilihat pada Gambar 6.24.

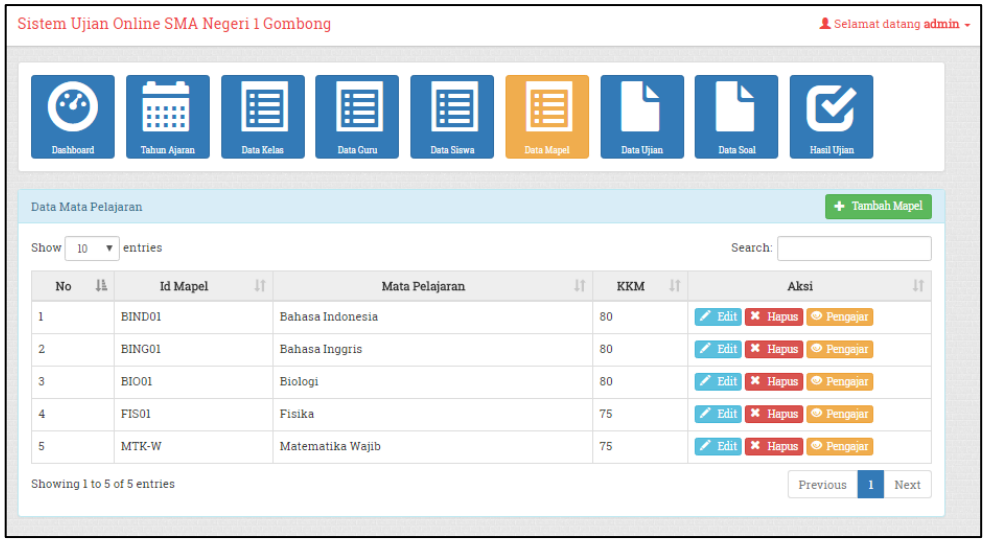


Gambar 6.24 Halaman Detail Siswa

5.2.6 Pengolahan Data Mapel

1. Informasi Data Mapel

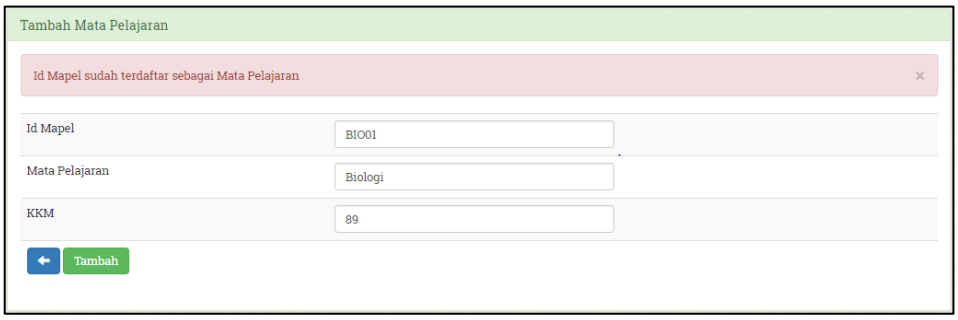
Halaman mapel menampilkan data mata pelajaran, dan kkm. Selain itu pengguna juga dapat melihat pengajar di setiap mata pelajarannya dengan memilih aksi pengajar. Setiap mata pelajaran yang sama setiap kelas bisa memiliki guru pengajar yang berbeda antara kelas satu dengan kelas yang lainnya dan memiliki nilai kkm yang dapat berubah. Adapun Tampilan halaman mapel dapat dilihat pada Gambar 6.25.



Gambar 6.25 Halaman Data Mapel

2. Halaman Tambah Mapel

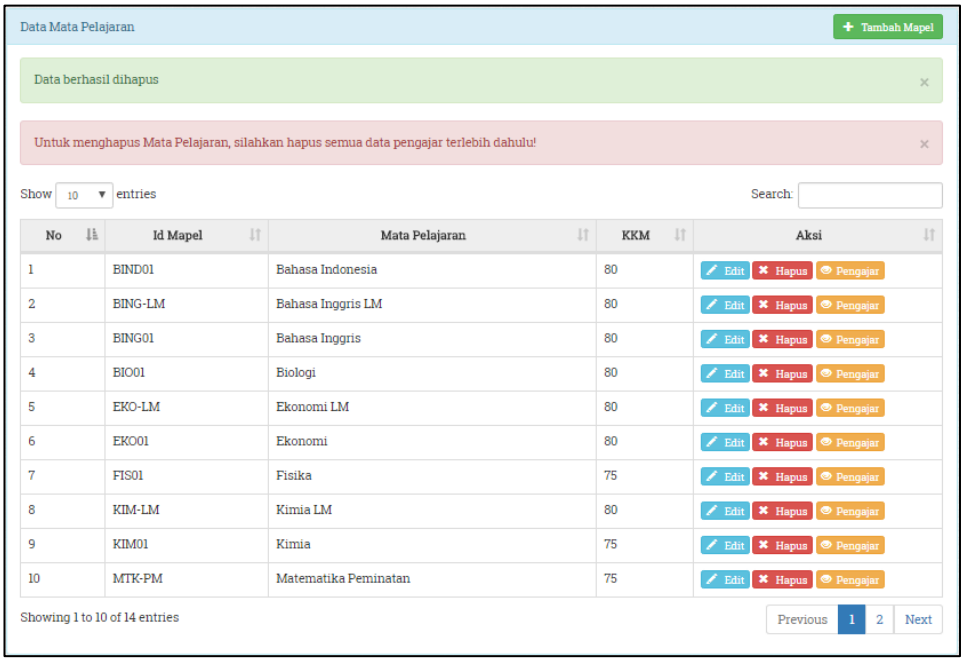
Pada saat pemasukan data mapel maka akan dilakukan pengecekan terhadap id mapel siswa. Data tidak sah apabila pengguna memasukkan data mapel dengan id mapel yang telah ada sebelumnya. Apabila id mapel sudah ada, maka sistem akan memunculkan pesan peringatan seperti pada Gambar 6.26.



Gambar 6.26 Pemberitahuan Id Mapel Sudah Terdaftar Sebagai Mapel

3. Hapus Mapel

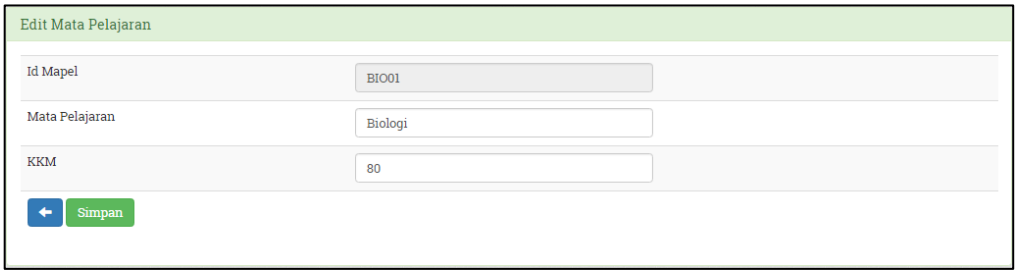
Pada saat hapus mapel, pengguna tidak dapat menghapus mapel yang sudah terdapat Guru pengajar. Akan muncul peringatan “Untuk menghapus Mata Pelajaran, silahkan hapus semua data pengajar terlebih dahulu!”. Hal ini dikarenakan didalamnya terdapat data-data yang saling terkait dengan pengajar pada mata pelajaran tersebut. Peringatan Mapel tidak dapat dihapus dapat dilihat pada Gambar 6.27.



Gambar 6.27 Peringatan saat Hapus Mata Pelajaran

4. Halaman Ubah Mapel

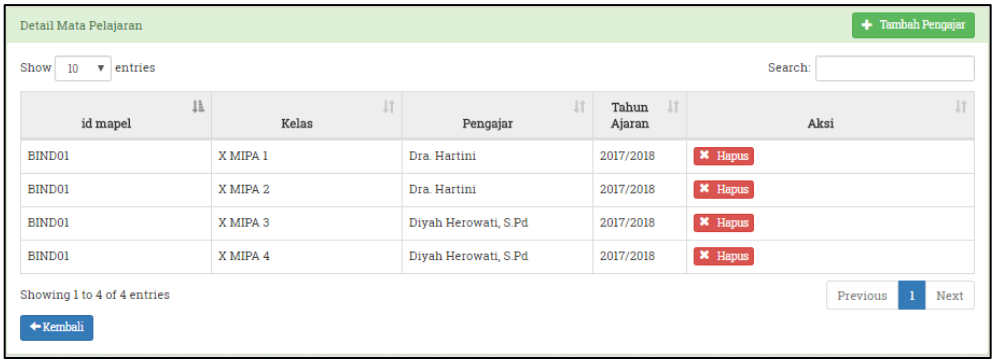
Untuk merubah data mapel, pengguna akan dibawa ke halaman ubah data mapel. Perubahan data mapel dapat dilakukan dengan memilih tautan aksi edit. Adapun halaman ubah data mapel ini dapat dilihat pada Gambar 6.28.



Gambar 6.28 Halaman Ubah Mapel

5. Halaman Lihat Pengajar

Pada halaman lihat pengajar sistem menampilkan daftar guru pengajar yang mengampu mata pelajaran. Setiap Mata Pelajaran dapat diampu oleh lebih dari satu guru, sehingga guru pengajar antara kelas satu dengan yang lainnya bisa sama atau bisa juga berbeda. Implementasi antarmuka halaman pengajar dapat dilihat pada Gambar 6.29.



Gambar 6.29 Halaman Lihat Pengajar

6. Halaman Tambah Pengajar

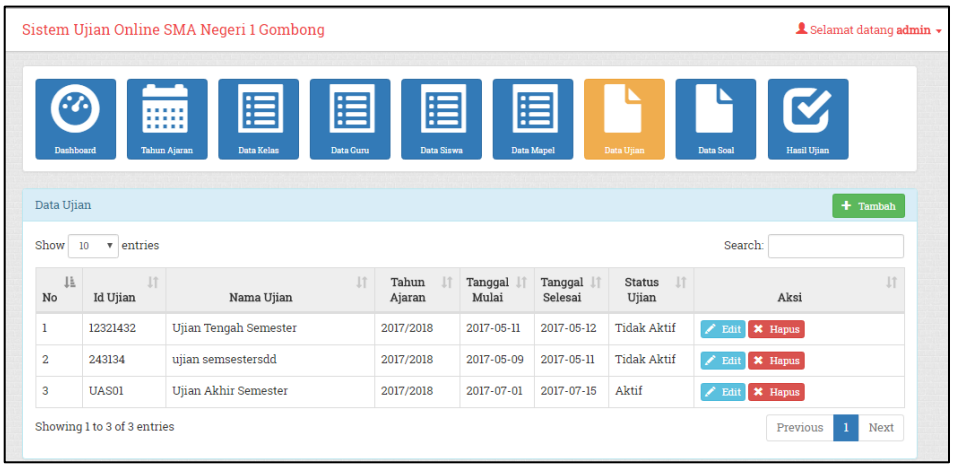
Halaman tambah pengajar ini berfungsi untuk menambahkan guru pengajar di setiap mata pelajaran sesuai kelas dan tahun ajaran. Setelah klik button tambah sistem akan menampilkan *alert* 'Data Berhasil Ditambahkan' pada halaman tambah Pengajar. Inplementasi antarmuka halaman tambah pengajar dapat dilihat pada Gambar 6.30.

Gambar 6.30 Halaman Tambah Pengajar

5.2.7 Pengolahan Data Ujian

1. Halaman Data Ujian

Halaman utama data ujian berupa daftar tabel seluruh data ujian yang terdaftar di dalam sistem. Pada halaman ini terdapat aksi tambah data ujian, ubah data ujian dan hapus data ujian. Masing-masing button ini berperan dalam proses pengelolaan data ujian. Terdapat pula kolom pencarian yang dapat digunakan pengguna untuk mencari data guru berdasarkan kata kunci tertentu. Halaman utama data ujian terdapat pada Gambar 6.31.

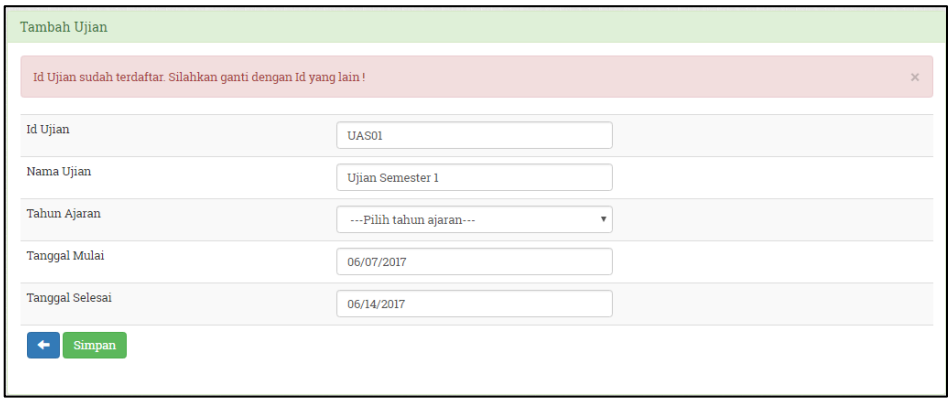


Gambar 6.31 Halaman Data Ujian

2. Halaman Tambah Ujian

Pada saat penambahkan data ujian maka akan dilakukan pengecekan terhadap id ujian yang diinputkan. Data tidak sah apabila pengguna memasukkan

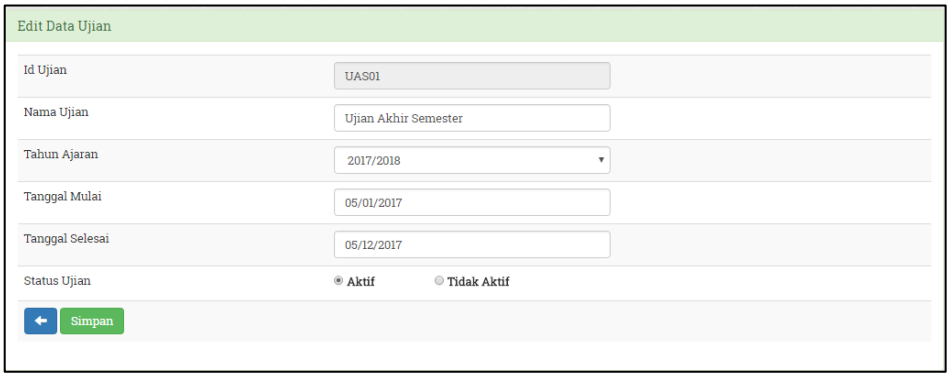
data ujian dengan id ujian yang telah ada sebelumnya. Apabila id ujian sudah ada, maka sistem akan memunculkan pesan peringatan seperti pada Gambar 6.32.



Gambar 6.32 Pemberitahuan Id Ujian Sudah Digunakan

3. Halaman Ubah Ujian

Untuk merubah data ujian, pengguna akan dibawa ke halaman ubah data ujian. Perubahan data ujian dapat dilakukan dengan memilih tautan aksi edit. Adapun halaman ubah data ujian ini dapat dilihat pada Gambar 6.33.

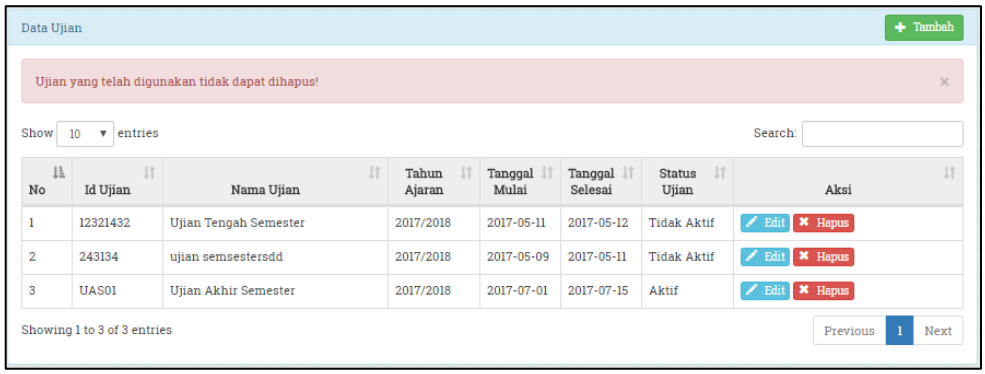


Gambar 6.33 Halaman Ubah Ujian

4. Hapus Ujian

Pada saat hapus data ujian, pengguna tidak dapat menghapus data yang sudah digunakan sebelumnya. Akan muncul peringatan "Ujian yang telah digunakan tidak dapat dihapus!". Hal ini dikarenakan didalamnya terdapat data-data yang saling terkait dengan ujian yang dilaksanakan pada tahun ajaran tersebut

misalnya data paket soal dan hasil ujian. Peringatan Ujian tidak dapat dihapus dapat dilihat pada Gambar 6.34.

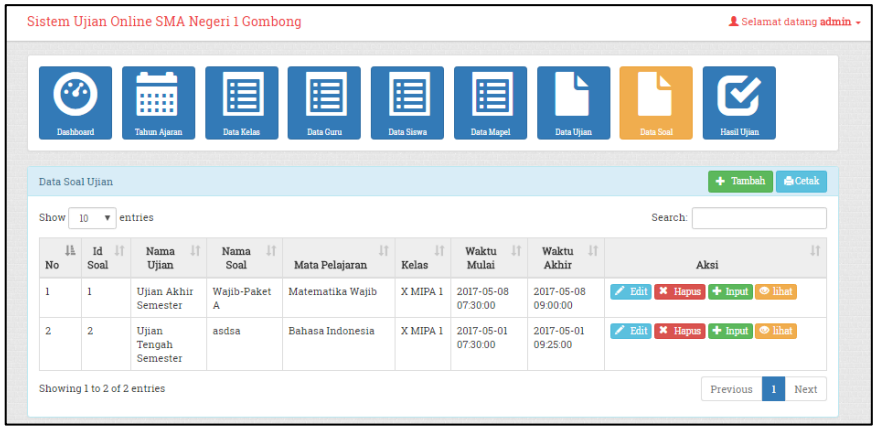


Gambar 6.34 Peringatan Ujian Tidak Dapat Dihapus

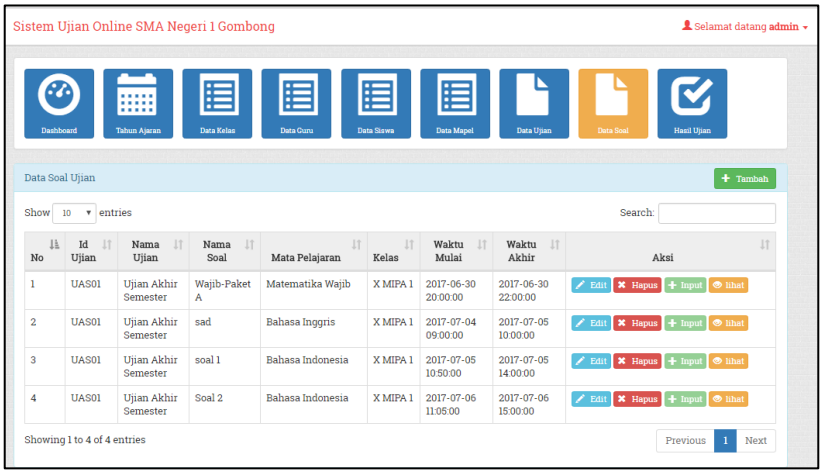
5.2.8 Pengolahan Data Soal

1. Halaman Data Soal

Halaman data soal menampilkan data soal yang terdiri dari id ujian, nama ujian, nama soal, mata pelajaran, kelas, waktu mulai, dan waktu akhir. Adapun Tampilan halaman soal dapat dilihat pada Gambar 6.35. Pada halaman data soal terdapat aksi input dimana apabila data paket soal sudah digunakan untuk pelaksanaan ujian maka pengguna tidak dapat menginputkan pertanyaan baru. Tombol dibedakan dengan button input *disable* apabila paket soal sudah digunakan untuk pelaksanaan ujian. Hal ini dikarenakan akan membuang-buang memori server. Adapun halaman data soal dengan aksi input *disable* dapat dilihat pada Gambar 6.36.



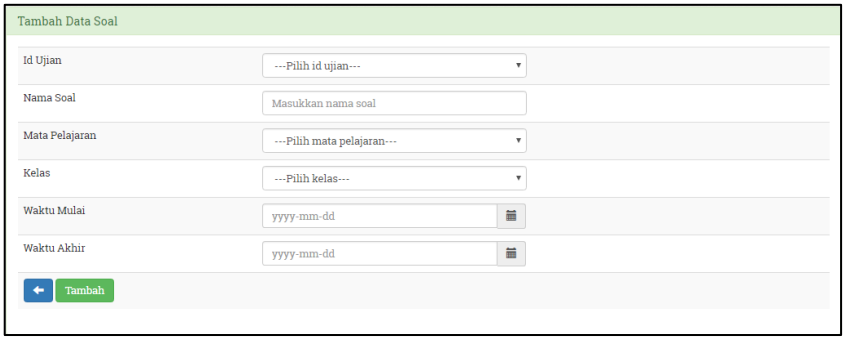
Gambar 6.35 Halaman Data Soal



Gambar 6.36 Tampilan Aksi Input *Disable* Pada Halaman Data Soal

2. Tambah Data Soal

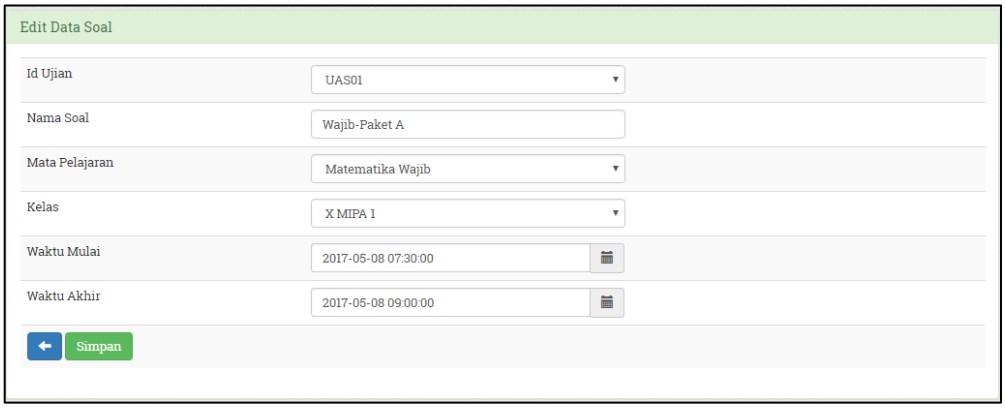
Halaman tambah data soal digunakan untuk menambahkan data soal baru yang merupakan data informasi soal untuk menginputkan pertanyaan. Adapun halaman tambah data soal terdapat pada Gambar 5.37.



Gambar 6.37 Halaman Tambah Data Soal

3. Ubah Data Soal

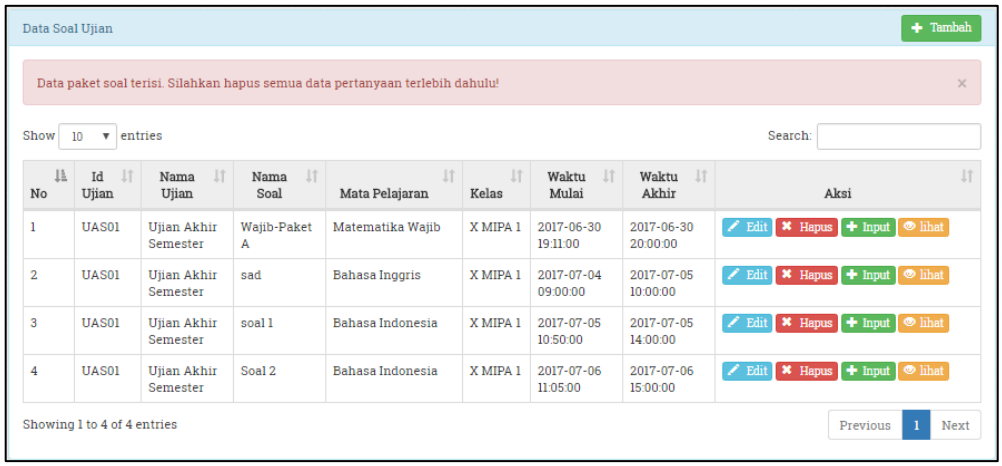
Untuk merubah data soal, pengguna akan dibawa ke halaman ubah data soal. Perubahan data soal dapat dilakukan dengan memilih tautan aksi edit. Adapun halaman ubah data soal ini dapat dilihat pada Gambar 6.38.



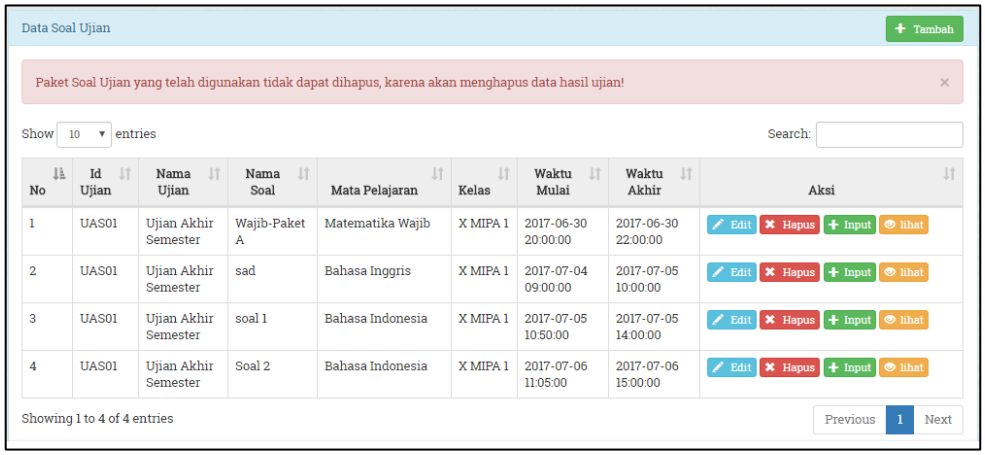
Gambar 6.38 Halaman Ubah Data Soal

4. Hapus Paket Soal

Pada saat hapus data soal, pengguna tidak dapat menghapus data soal secara langsung, harus menghapus data pertanyaan terlebih dahulu. Akan muncul peringatan "Data paket soal terisi. Silahkan hapus semua data pertanyaan terlebih dahulu!". Peringatan hapus data soal dapat dilihat pada Gambar 6.39. Selain itu pengguna juga tidak dapat menghapus data paket soal yang sudah digunakan untuk pelaksanaan ujian. Hal ini dikarenakan akan menghapus data lain yang saling berhubungan seperti hasil ujian siswa pada paket soal tersebut. Adapun Peringatan tidak dapat menghapus data soal dapat dilihat pada Gambar 6.40.



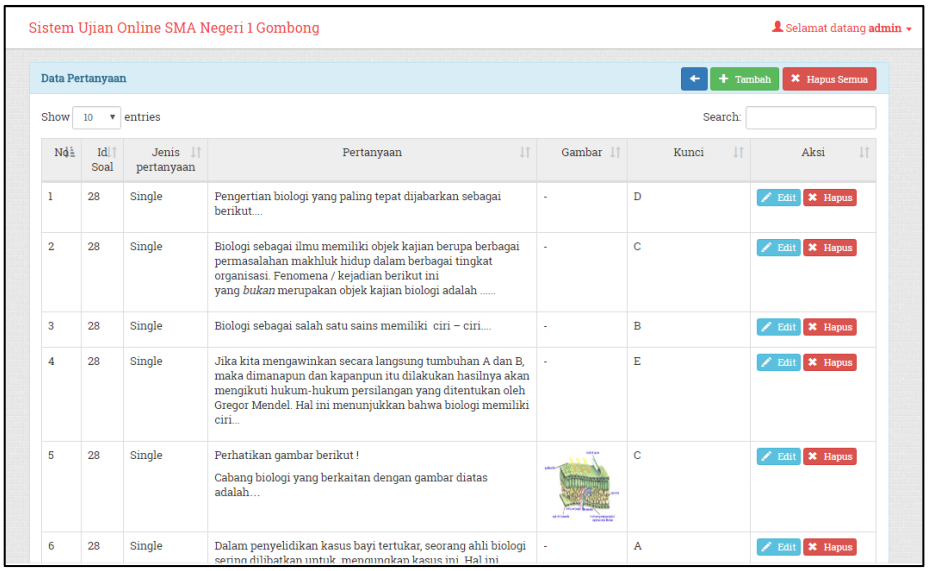
Gambar 6.39 Peringatan Hapus Data Soal



Gambar 6.40 Peringatan Tidak Dapat Menghapus Data Soal

5. Daftar Pertanyaan

Halaman daftar pertanyaan ini digunakan untuk melihat daftar pertanyaan yang telah ditambahkan sebelumnya. Pada halaman ini ditampilkan id soal, jenis pertanyaan, pertanyaan, gambar, dan kunci jawaban. Adapun halaman daftar pertanyaan ini dapat dilihat pada Gambar 6.41.

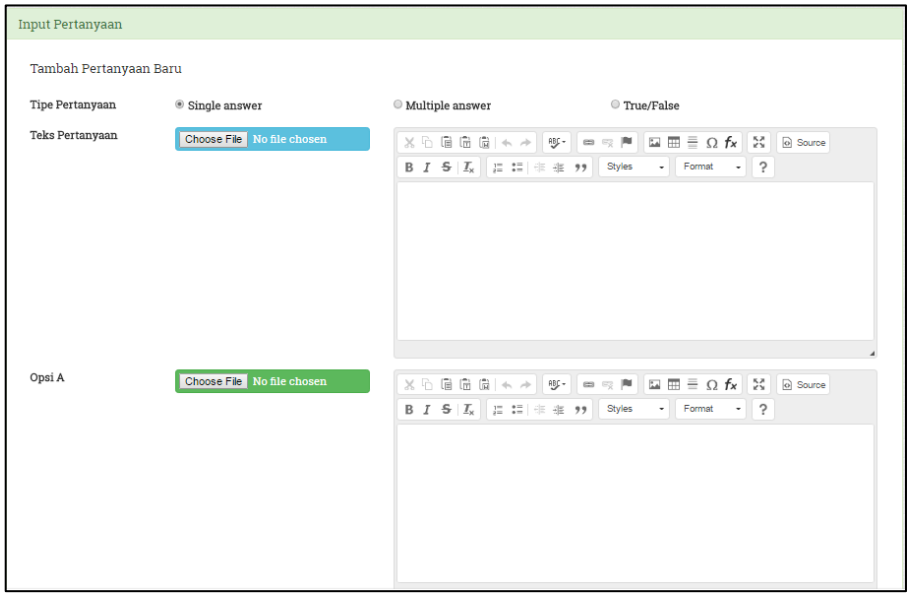


Gambar 6.41 Halaman Daftar Pertanyaan

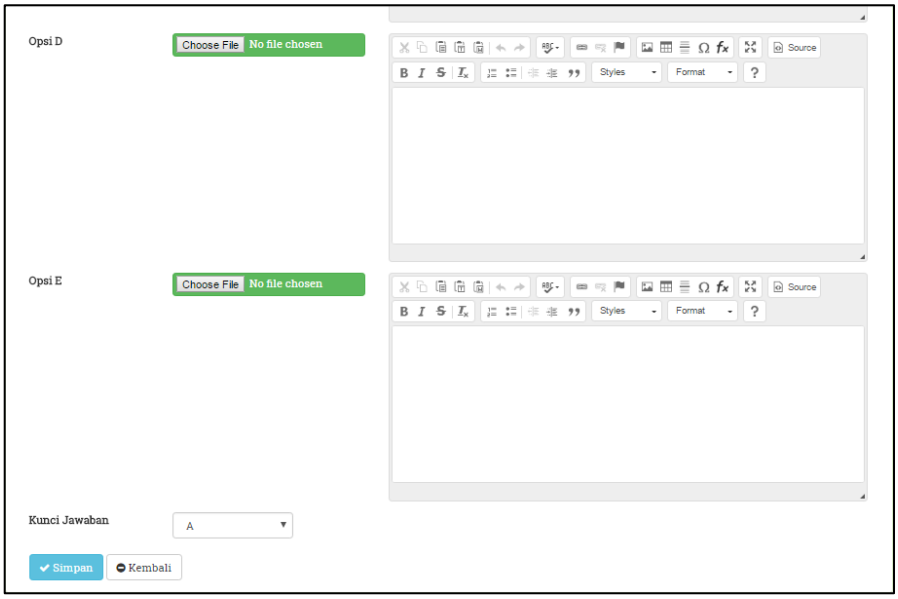
6. Tambah Pertanyaan

Halaman tambah pertanyaan ini digunakan untuk menambahkan pertanyaan pada setiap masing-masing soal. Pada halaman ini ditampilkan cek

editor untuk menginputkan teks pertanyaan dan opsi jawaban. Terdapat *button file choose* untuk menambahkan gambar pada teks pertanyaan. Adapun halaman ubah data soal ini dapat dilihat pada Gambar 6.42 dan 6.43.



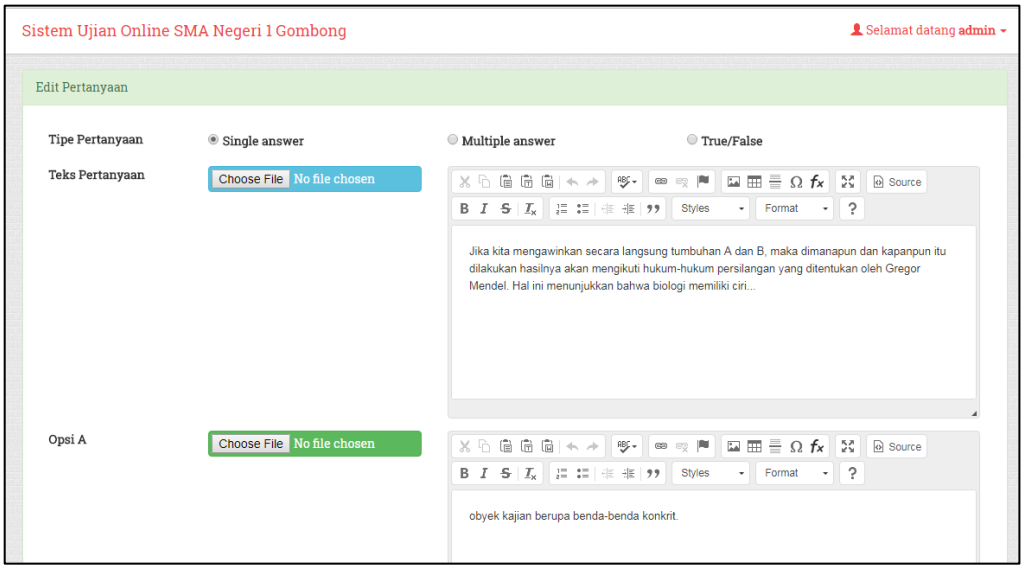
Gambar 6.42 Halaman Tambah Pertanyaan



Gambar 6.43 Halaman Tambah Pertanyaan (Lanjutan)

7. Ubah Pertanyaan

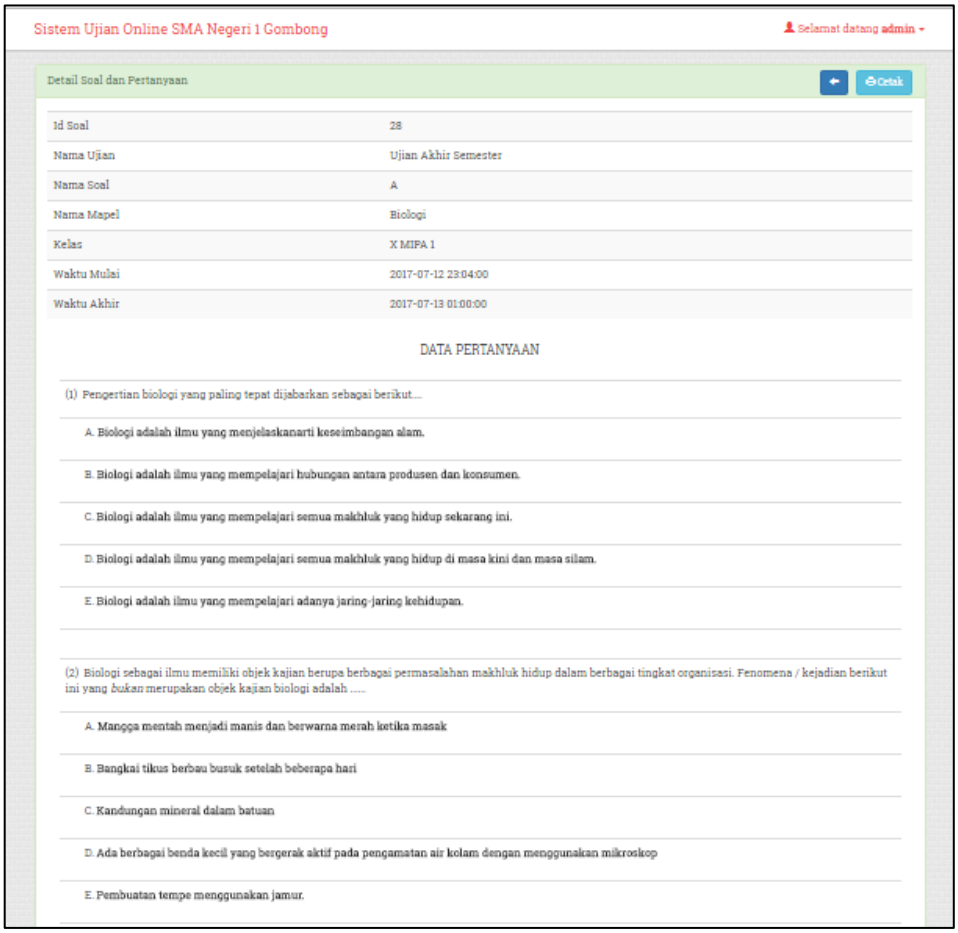
Untuk merubah data pertanyaan, pengguna akan dibawa ke halaman ubah pertanyaan. Perubahan data pertanyaan dapat dilakukan dengan memilih tautan aksi edit. Adapun halaman ubah pertanyaan ini dapat dilihat pada Gambar 6.44.



Gambar 6.44 Halaman Ubah Pertanyaan

8. Lihat Detail Soal dan Pertanyaan

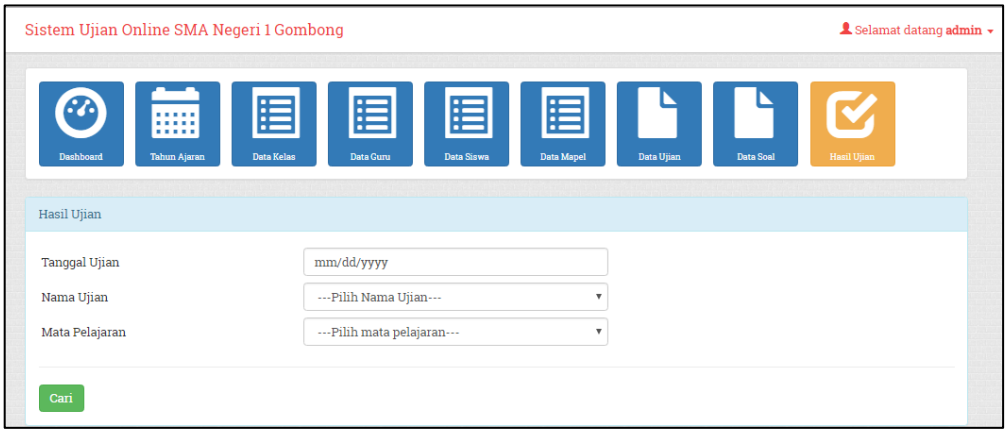
Pada halaman ini menampilkan detail soal dan pertanyaan yang kemudian dapat dicetak oleh pengelola. Pada halaman ini terdapat menu cetak yang digunakan untuk mengexpor halaman detail soal dan pertanyaan menjadi lembar soal dengan format pdf. Adapun halaman lihat detail soal dan pertanyaan dapat dilihat pada Gambar 6.45.



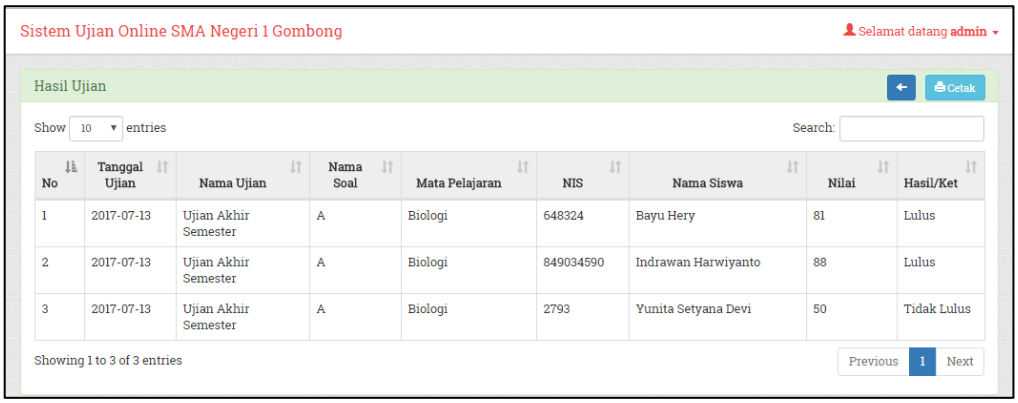
Gambar 6.45 Halaman Lihat Detail Soal dan Pertanyaan

5.2.9 Pengolahan Data Hasil Ujian

Pada halaman hasil ujian ini menampilkan form tanggal ujian, nama ujian, dan mata pelajaran yang digunakan untuk mencari hasil ujian berdasarkan data tersebut. Adapun halaman hasil ujian dapat dilihat pada Gambar 6.46. *Admin* dan guru juga dapat melakukan cetak hasil ujian. Setelah klik cari pengelola diarahkan menuju ke halaman cetak data hasil ujian. Adapun halaman cetak data hasil ujian ini dapat dilihat pada Gambar 6.47.



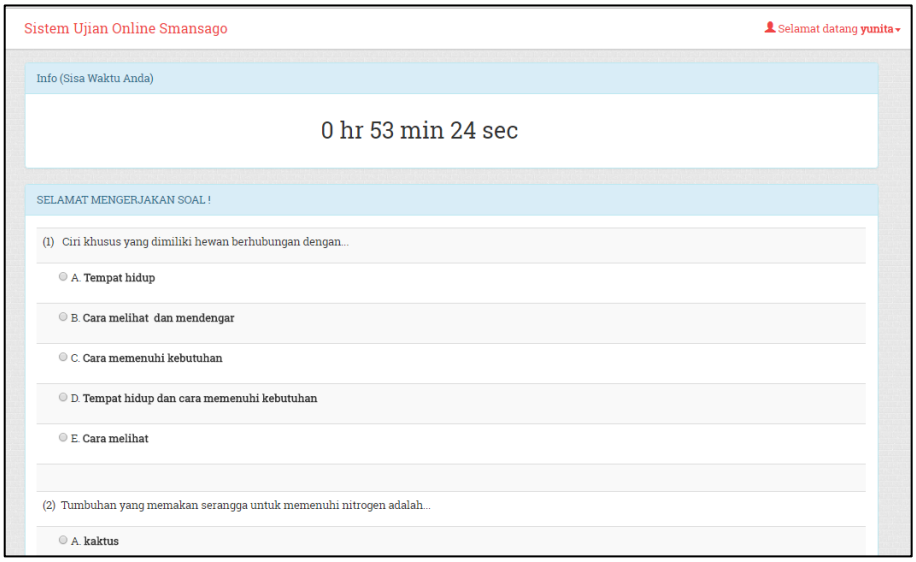
Gambar 6.46 Halaman Data Hasil Ujian



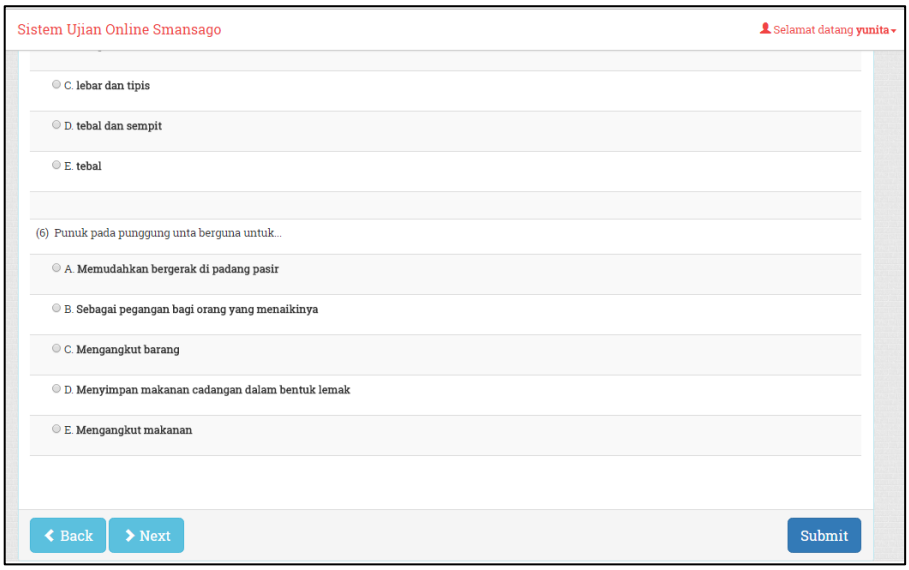
Gambar 6.47 Halaman Cetak Data Hasil Ujian

5.2.10 Pengelolaan Soal Ujian Siswa

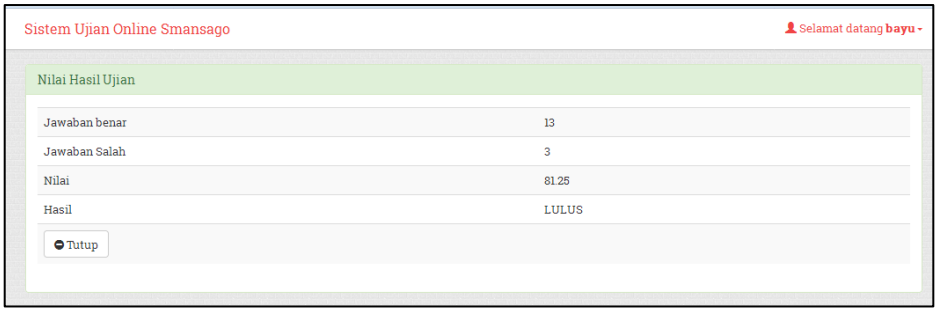
Halaman Soal Ujian Siswa ini adalah halaman yang digunakan oleh *user* siswa. Pada halaman ini siswa mengerjakan soal sesuai dengan waktu mundur yang ditampilkan di halaman. Setelah siswa klik *button* mulai ujian pada dashboard siswa baru siswa dapat mengerjakan soal. Terdapat button submit digunakan apabila siswa telah selesai melaksanakan ujian. Adapun halaman soal ujian siswa dapat dilihat pada Gambar 6.48 dan 6.49. Setelah siswa submit siswa akan melihat hasil ujiannya. Adapun Halaman nilai ujian siswa dapat dilihat pada Gambar 6.50.



Gambar 6.48. Halaman Soal Ujian Siswa



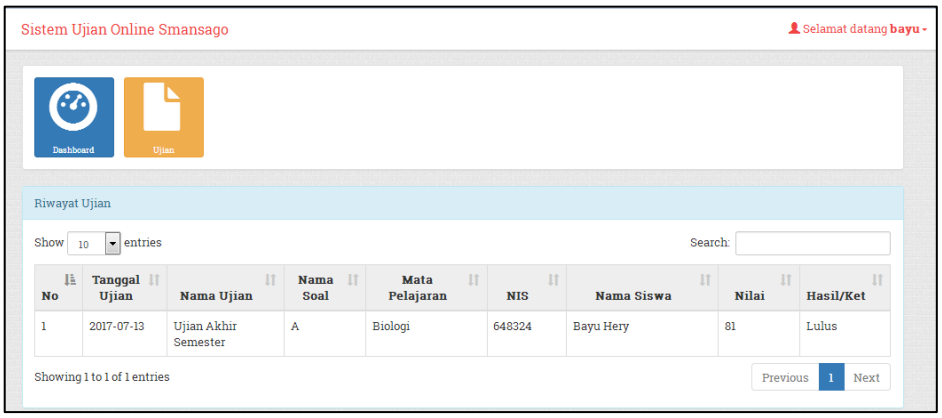
Gambar 6.49. Halaman Soal Ujian Siswa (Lanjutan)



Gambar 6.50 Halaman Nilai Ujian Siswa

5.2.11 Pengelolaan Data Riwayat Ujian Siswa

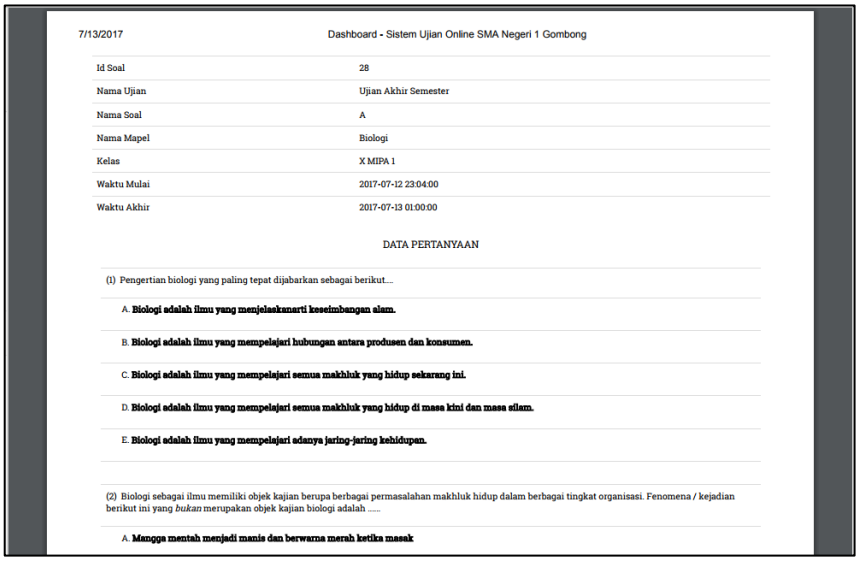
Pada halaman riwayat ujian, siswa dapat melihat semua hasil nilai dari riwayat ujian yang telah dilaksanakan sebelumnya. Adapun halaman riwayat ujian siswa dapat dilihat pada Gambar 6.51.



Gambar 6.51 Halaman Riwayat Ujian Siswa

5.2.12 Pengelolaan Cetak Soal

Cetak soal dapat dilakukan setelah admin maupun guru masuk ke halaman soal dengan memilih aksi lihat detail soal. Pada halaman tersebut terdapat menu cetak yang digunakan untuk mengeksport halaman detail soal dan pertanyaan menjadi lembar soal. Data soal dan pertanyaan yang telah diexpor akan menjadi bentuk pdf dan dapat dicetak langsung dengan ukuran kertas A4. Data soal dan pertanyaan yang berbentuk pdf akan menampilkan pertanyaan lengkap beserta detail soalnya. Adapun tampilan cetak soal ujian dalam print window dapat dilihat pada Gambar 6.52.



7/13/2017 Dashboard - Sistem Ujian Online SMA Negeri 1 Gombong

Id Soal	28
Nama Ujian	Ujian Akhir Semester
Nama Soal	A
Nama Mapel	Biologi
Kelas	X MIPA 1
Waktu Mulai	2017-07-12 23:04:00
Waktu Akhir	2017-07-13 01:00:00

DATA PERTANYAAN

(1) Pengertian biologi yang paling tepat dijabarkan sebagai berikut...

A. **Biologi adalah ilmu yang menjelaskanarti keseimbangan alam.**

B. **Biologi adalah ilmu yang mempelajari hubungan antara produsen dan konsumen.**

C. **Biologi adalah ilmu yang mempelajari semua makhluk yang hidup sekarang ini.**

D. **Biologi adalah ilmu yang mempelajari semua makhluk yang hidup di masa kini dan masa silam.**

E. **Biologi adalah ilmu yang mempelajari adanya jaring-jaring kehidupan.**

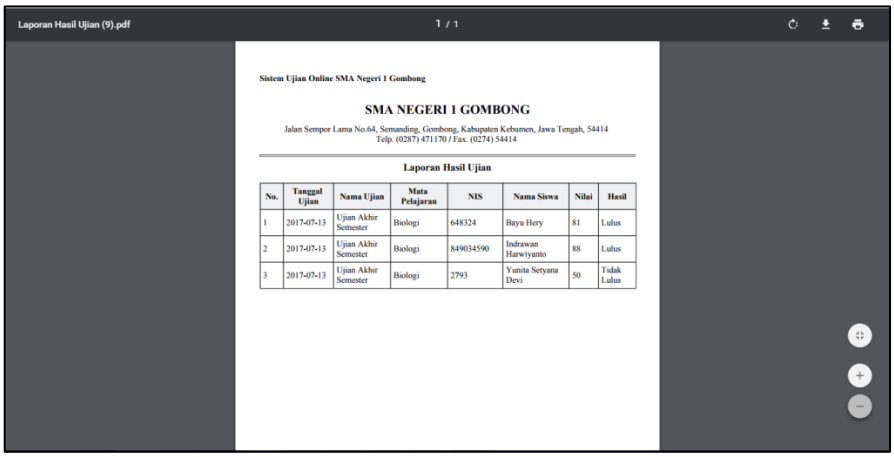
(2) Biologi sebagai ilmu memiliki objek kajian berupa berbagai permasalahan makhluk hidup dalam berbagai tingkat organisasi. Fenomena / kejadian berikut ini yang *bukan* merupakan objek kajian biologi adalah .....

A. **Mangga mentah menjadi manis dan berwarna merah ketika masak**

Gambar 6.52 Tampilan Cetak Soal Ujian Dalam Print Window

5.2.13 Pengelolaan Cetak Hasil Ujian

Cetak hasil ujian dapat dilakukan setelah admin maupun guru masuk ke halaman hasil ujian, kemudian cari data berdasarkan tanggal, nama ujian, dan mata pelajaran. Pada halaman tersebut terdapat menu cetak yang digunakan untuk mengeksport halaman nilai nilai hasil ujian siswa menjadi laporan. Hasil ujian yang telah dieksport akan menjadi bentuk pdf dan dapat dicetak langsung dengan ukuran kertas A4. Hasil ujian yang berbentuk pdf akan menampilkan nilai siswa dan hasil lulus atau tidaknya ujian. Gambar 6.53 menampilkan hasil ujian dalam bentuk pdf.



Laporan Hasil Ujian (9).pdf 1 / 1

Sistem Ujian Online SMA Negeri 1 Gombong

SMA NEGERI 1 GOMBONG

Jalan Sempor Lama No.64, Semanding, Gombong, Kabupaten Kebumen, Jawa Tengah, 54414
Telp. (0287) 471170 / Fax. (0274) 54414

Laporan Hasil Ujian

No.	Tanggal Ujian	Nama Ujian	Mata Pelajaran	NIS	Nama Siswa	Nilai	Hasil
1	2017-07-13	Ujian Akhir Semester	Biologi	648324	Bayu Hery	81	Lulus
2	2017-07-13	Ujian Akhir Semester	Biologi	849034590	Indrawan Harwiyanto	88	Lulus
3	2017-07-13	Ujian Akhir Semester	Biologi	2793	Yunita Setyana Devi	50	Tidak Lulus

Gambar 6.53 Tampilan Hasil Ujian Siswa Dalam Bentuk PDF

BAB VII
KESIMPULAN DAN SARAN

7.1 Kesimpulan

Berdasarkan tahap-tahap yang telah dilakukan pada penelitian dimulai dari penemuan masalah, pemahaman proses bisnis, hingga perancangan, implementasi dan pengujian sistem, maka diperoleh sebuah hasil dan kesimpulan terhadap sistem informasi Ujian Online berbasis web pada SMA Negeri 1 Gombong. Adapun pencapaian dari fokus permasalahan yang dapat diselesaikan dengan adanya sistem informasi ini adalah sebagai berikut :

1. Sistem informasi Ujian Online berbasis web pada SMA Negeri 1 Gombong telah berhasil dibangun dan dapat memenuhi kebutuhan dalam permasalahan utama sekolah yaitu pelaksanaan ujian secara komputerisasi, pengelolaan data pengajar, pengelolaan soal ujian, pengelolaan hasil ujian, cetak soal ujian dan cetak laporan hasil ujian.
2. Berdasarkan hasil pengujian, sistem informasi ujian online telah mampu dimplementasikan sesuai rancangan dalam membuat ujian online berdasarkan proses bisnis yang telah dilakukan.

7.2 Saran

Berdasarkan penelitian yang telah penulis lakukan, terdapat beberapa saran untuk pengembangan sistem selanjutnya. Adapun beberapa saran tersebut adalah sebagai berikut :

1. Pengembangan sistem dengan menambahkan tipe soal essay yang dapat digunakan untuk pelaksanaan ujian online.
2. Pengembangan sistem mengenai penilaian secara statistik setiap mata pelajaran yang dapat digunakan untuk pengambilan keputusan, seperti menentukan bagus tidaknya soal yang dibuat oleh guru.
3. Pengembangan sistem mengenai pengacakan soal yang digunakan untuk ujian pada setiap user siswa, sehingga meminimalisir kecurangan siswa.

DAFTAR PUSTAKA

Adha, M.L, 2011, Implementasi Aplikasi Ujian Online Pada Training Karyawan Menggunakan Metode Computerized Classification Test dengan Adaptive Feedback, *Skripsi*, Fakultas Teknik, Universitas Indonesia, Jakarta.

Cochran, D., 2012, *Twitter Bootstrap Web Development How-To*, PACKT Publisher.

Connolly, TM., dan Carolyn, EB., 2002, Database Systems : *A Practical Approach to Design, Implementation, and Management, Third Edition.* Addison-Wesley, Reading, Massachusetts.

Darmawan, D., 2000, Aplikasi PHP Pada Website Online Examination, *Penelitian,* Universitas Diponegoro, Semarang

Hambra A., 2017, Sistem Informasi E-Raport Berbasis Web, *Tugas Akhir*, Fakultas Sekolah Vokasi, Universitas Gadjah Mada, Yogyakarta.

Ihsan,R.R., 2012, Sistem Informasi Ujian Online dan Penilaian Siswa Berbasis Web Pada SMK Pasundan 4 Bandung, *Skripsi*, Universitas Komputer Indonesia, Bandung.

Jogiyanto, HM., 2005, *Analisis dan Desain Sistem Informasi*, Andi, Yogyakarta.

Kadir, A., 1999. *Konsep dan Tuntutan Praktis Basis Data*, Andi Offset, Yogyakarta.

Kusworo, A.P, 2010, Pembuatan Sistem Ujian Online, *Penelitian*, Program Studi Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sebelas Maret, Surakarta

Lyna, A., 2010, Sistem Ujian Online Pada SMA PGRI 109 Tangerang, *Penelitian*, Universitas Mercu Buana, Yogyakarta.

Nugroho, A., 2011, *Perancangan dan Implementasi Sistem Basis Data*, Andi Publisher, Yogyakarta.

Ramadhani, G. 2003, *Modul Pengenalan Internet*, Yogyakarta.

Sukoco, D., 2012, Aplikasi E-Learning Berbasis Web Pada SMA N 2 Klaten, *Penelitian*, STMIK AMIKOM, Yogyakarta.

Supriyanto, A., 2005.*Pengantar Teknologi Informasi*, Salemba Infotek, Jakarta.

Sutabri, Tata. 2012. *Konsep Dasar Informasi*. Yogyakarta: Andi.