BAB V

IMPLEMENTASI DAN PENGUJIAN SISTEM

5.1. Implementasi

Implementasi Sistem Informasi Akademik berbasis *web* pada Yayasan SMA Babussalam Bandung ini dilakukan dengan menggunakan bahasa dan pemograman *PHP*, dengan basis data yang digunakan ialah *Xampp*, dapat dijalankan pada berbagai *platform* sistem operasi dan perangkat keras, tetapi implementasi dan pengujian sepenuhnya hanya dilakukan di perangkat keras *PC* (*Personal Computer*) dengan *sistem* operasi *Microsoft Windows XP*.

5.1.1. Batasan Implementasi

Dalam mengimplementasikan perangkat lunak, pemantauan dan pengendalian aplikasi ini, ada beberapa hal yang menjadi batasan implementasi yaitu :

1. Dalam sistem informasi akademik berbasis *web* ini hanya sebatas pengelolaan pendaftaran siswa baru, pembagian kelas dan walikelas, pembuatan jadwal pelajaran, dan pengolahan nilai akhir.
2. Basis data yang digunakan dalam pengimplementasian ini adalah *Xampp* baik untuk basis data utama maupun ekstraksi basis data yang disebarkan ke *User*. Pengaksesan basis data di *server* dilakukan melalui *Localhost*.
3. Alamat IP dari komputer yang digunakan merupakan alamat IP statis.

4. Sistem ini hanya berupa perancangan dan belum bisa langsung diterapkan menggantikan sistem yang lama.

5.1.2. Implementasi Perangkat Lunak

Untuk pengembangan perangkat lunak ini digunakan bahasa pemrograman PHP. PHP dipilih sebagai perangkat lunak pengembang karena menyediakan fasilitas yang memadai untuk membuat perangkat lunak yang berbasis *web*. Sementara *Xampp* digunakan sebagai perangkat lunak pengembang dalam pembuatan basis data.

5.1.3. Implementasi Perangkat Keras

Perangkat keras yang dibutuhkan berdasarkan kebutuhan minimal yang harus terpenuhi antara lain :

1. *Server*
    a. Menggunakan minimal *Prossesor Intel Pentium* atau yang sekelasnya
    b. Menggunakan *RAM* minimal 512 MB
    c. Tersedia *Hard Drive,* untuk media penyimpanan, minimal 10MB untuk *server* diluar basis data
    d. *LAN Card*
    e. *Mouse, Keyboard* dan *Monitor* sebagai peralatan antar muka
2. *User*
    a. Menggunakan minimal *prossesor Intel Pentium* atau sekelasnya
    b. Menggunakan *RAM* minimal 64MB
    c. *LAN Card*
    d. *Mouse, Keyboard* dan *Monitor* sebagai peralatan antarmuka

5.1.4. Implementasi Basis Data

Pembuatan basis data dilakukan dengan menggunakan bahasa *SQL*, dimana *DBMS* yang digunakan adalah *Xampp*, Implementasi basis datanya dalam *SQL* adalah sebagai berikut :

```
-- Database: `yayababu`
-- Struktur dari tabel `calon_siswa`
CREATE TABLE IF NOT EXISTS `calon_siswa` (
  `no_pendaftaran` varchar(50) CHARACTER SET utf8 NOT NULL,
  `nama_siswa` varchar(50) CHARACTER SET utf8 NOT NULL,
  `tempat_lahir` varchar(50) CHARACTER SET utf8 NOT NULL,
  `tanggal_lahir` date NOT NULL,
  `jenis_kelamin` varchar(50) CHARACTER SET utf8 NOT NULL,
  `alamat` varchar(50) CHARACTER SET utf8 NOT NULL,
  `nama_ayah` varchar(50) CHARACTER SET utf8 NOT NULL,
  `pekerjaan_ayah` varchar(50) CHARACTER SET utf8 NOT NULL,
  `nama_ibu` varchar(50) CHARACTER SET utf8 NOT NULL,
  `pekerjaan_ibu` varchar(50) CHARACTER SET utf8 NOT NULL,
  `agama` varchar(50) CHARACTER SET utf8 NOT NULL,
  `asal_sekolah` varchar(50) CHARACTER SET utf8 NOT NULL,
  `golongan_darah` varchar(50) CHARACTER SET utf8 NOT NULL,
  `jumlah_saudara` varchar(50) CHARACTER SET utf8 NOT NULL,
  `anak_ke` int(20) NOT NULL,
  `no_sttb` int(20) NOT NULL,
  `tahun_ajaran` int(20) NOT NULL,
  `nilai_un` int(20) NOT NULL,
  `telp` int(20) NOT NULL,
  `tgl_daftar` timestamp NOT NULL
  `status_daftar` varchar(50) NOT NULL,
  PRIMARY KEY (`no_pendaftaran`)
-- Dumping data untuk tabel `calon_siswa`
INSERT INTO `calon_siswa` (`no_pendaftaran`, `nama_siswa`, `tempat_lahir`, `tanggal_lahir`, `jenis_kelamin`, `alamat`, `nama_ayah`, `pekerjaan_ayah`, `nama_ibu`, `pekerjaan_ibu`, `agama`, `asal_sekolah`, `golongan_darah`, `jumlah_saudara`, `anak_ke`, `no_sttb`, `tahun_ajaran`, `nilai_un`, `telp`, `tgl_daftar`, `status_daftar`) VALUES
('PSB201207092', 'Rino', 'Kediri', '1990-09-16', 'Laki-laki', 'jl.bogo', 'Ayah', 'Swasta', 'Ibu', 'Ibu Rumah Tangga', 'ISLAM', 'SMP 1 Kediri', 'A', '2', 2, 2147483647, 0, 90, 2147483647, '2012-07-09 21:17:08', 'Setuju'),
-- Struktur dari tabel `jadwal`
CREATE TABLE IF NOT EXISTS `jadwal` (
  `id_jadwal` int(20) NOT NULL,
  `nip` bigint(50) NOT NULL,
```

```
  `kode_pelajaran` varchar(20) NOT NULL,
  `hari` varchar(20) NOT NULL,
  `jam_mulai` time NOT NULL,
  `jam_selesai` time NOT NULL,
  `kode_kelas` varchar(50) NOT NULL,
  `tgl_input` timestamp NOT NULL
  PRIMARY KEY (`id_jadwal`)
-- Dumping data untuk tabel `jadwal`
INSERT INTO `jadwal` (`id_jadwal`, `nip`, `kode_pelajaran`, `hari`, `jam_mulai`,
`jam_selesai`, `kode_kelas`, `tgl_input`) VALUES
-- --------------------------------------------------------
-- Struktur dari tabel `kelas`
CREATE TABLE IF NOT EXISTS `kelas` (
  `kode_kelas` varchar(50) NOT NULL,
  `nama_kelas` varchar(20) NOT NULL,
  `kapasitas_kelas` int(11) NOT NULL,
  `jumlah_siswa` int(15) NOT NULL,
  `untuk_kelas` int(10) NOT NULL,
  `nip` varchar(20) NOT NULL,
  `tgl_input` timestamp NOT NULL
  PRIMARY KEY (`kode_kelas`)
-- Dumping data untuk tabel `kelas`
INSERT INTO `kelas` (`kode_kelas`, `nama_kelas`, `kapasitas_kelas`,
`jumlah_siswa`, `untuk_kelas`, `nip`, `tgl_input`) VALUES
- --------------------------------------------------------
-- Struktur dari tabel `nilai`
CREATE TABLE IF NOT EXISTS `nilai` (
  `id_nilai` int(20) NOT NULL,
  `nis` int(20) NOT NULL,
  `nip` bigint(20) NOT NULL,
  `kode_pelajaran` varchar(20) NOT NULL,
  `kode_kelas` varchar(50) NOT NULL,
  `nilai_tugas` int(11) NOT NULL,
  `nilai_ulangan` int(11) NOT NULL,
  `nilai_uts` int(11) NOT NULL,
  `nilai_uas` int(11) NOT NULL,
  `rata-rata` int(20) NOT NULL,
  `indeks` varchar(50) NOT NULL,
  `tgl_input` timestamp NOT NULL
  PRIMARY KEY (`id_nilai`)
-- Dumping data untuk tabel `nilai`
- --------------------------------------------------------
-- Struktur dari tabel `pegawai`
CREATE TABLE IF NOT EXISTS `pegawai` (
  `nip` bigint(50) NOT NULL,
  `nama_pegawai` varchar(50) NOT NULL,
```

```
  `tempat_lahir` varchar(50) NOT NULL,
  `tanggal_lahir` date NOT NULL,
  `alamat` varchar(50) NOT NULL,
  `agama` varchar(50) NOT NULL,
  `pendidikan` varchar(50) NOT NULL,
  `no_telp` int(20) NOT NULL,
  `jenis_kelamin` varchar(50) NOT NULL,
  `status` varchar(50) NOT NULL,
  `kode_pelajaran` varchar(50) NOT NULL,
  `password` varchar(50) NOT NULL,
  `foto` varchar(50) NOT NULL,
  `tgl_daftar` timestamp NOT NULL
  PRIMARY KEY (`nip`)
-- Dumping data untuk tabel `pegawai`
INSERT INTO `pegawai` (`nip`, `nama_pegawai`, `tempat_lahir`, `tanggal_lahir`, `alamat`, `agama`, `pendidikan`, `no_telp`, `jenis_kelamin`, `status`, `kode_pelajaran`, `password`, `foto`, `tgl_daftar`) VALUES
(2201207041, 'Lala Karamel', 'Banyumas', '1985-06-06', 'Jl.Haseum No.123', 'Islam', 'S1', 1236864989, 'Perempuan', 'Tata Usaha', '', 'a', 'Tulips.jpg', '2012-07-07 01:38:09'),
-- --------------------------------------------------------
-- Struktur dari tabel `pelajaran`
CREATE TABLE IF NOT EXISTS `pelajaran` (
  `kode_pelajaran` varchar(50) NOT NULL,
  `nama_pelajaran` varchar(50) NOT NULL,
  `tgl_input` timestamp NOT NULL
  PRIMARY KEY (`kode_pelajaran`)
-- Dumping data untuk tabel `pelajaran`
INSERT INTO `pelajaran` (`kode_pelajaran`, `nama_pelajaran`, `tgl_input`) VALUES
('BI2012', 'Bhs_Inggris', '2012-07-07 08:41:12'),
- --------------------------------------------------------
-- Struktur dari tabel `siswa`
CREATE TABLE IF NOT EXISTS `siswa` (
  `nis` int(20) NOT NULL,
  `nip` bigint(20) NOT NULL,
  `no_pendaftaran` varchar(20) NOT NULL,
  `kode_kelas` varchar(20) NOT NULL,
  `tgl_diterima` timestamp NOT NULL
  `foto` varchar(50) NOT NULL,
  `pass` varchar(20) NOT NULL,
  PRIMARY KEY (`nis`)
-- Dumping data untuk tabel `siswa`
INSERT INTO `siswa` (`nis`, `nip`, `no_pendaftaran`, `kode_kelas`, `tgl_diterima`, `foto`, `pass`) VALUES
(201207092, 0, 'PSB201207092', 'KL2012 A', '2012-07-09 21:17:08', '', 'b'),
```

5.1.5. Implementasi Antarmuka

Dalam *PHP* (5.2.3.), implementasi antarmuka dilakukan dengan sebuah *webpage* yang berekstensi *.PHP*. Setiap halaman dalam perangkat lunak dibuat *file* program yang ditulis dengan berekstensi.*php*. Berikut ini adalah implementasi dari setiap halaman yang dibuat dan dibedakan berdasarkan *User* dan admin.

1. Implementasi Halaman Utama *User*

Implementasi halaman yang terdapat pada halaman utama *user* dibuat dalam bentuk *file* program dengan berekstensi *PHP* dapat dilihat pada tabel 5.1 berikut :

Tabel 5.1. Implementasi Sub Menu *User*

Sub Menu	Deskripsi	Nama *File*
Pendaftaran	*File* program untuk melakukan pendaftaran calon siswa baru	pendaftaran.*php*
Login	*File* program untuk menangani *user login*	index.*php*
Halaman Utama	*File* program untuk melihat sejarah yayasan	index.*php*
Profil	*File* program untuk melihat visi dan misi	visidanmisi.*php*
Tanya Jawab	*File* program untuk memberikan pertanyaan mengenai Yayasan SMA Babussalam	tanyajawab.*php*
Hubungi Kami	*File* program untuk memberi informasi alamat dan nomor telepon yayasan	contactus.*php*
Form Input Data Guru	*File* program untuk menginputkan data guru	inputdataguru.*php*
Form Input Nilai Akhir	*File* program untuk menginputkan nilai akhir	inputnilai.*php*
Logout	*File* program untuk *user* keluar dari aplikasi	index.*php*

2. Implementasi Halaman menu Admin

Implementasi halaman yang terdapat pada Admin yang dibuat dalam bentuk *file* program dengan berekstensi *PHP* dapat dilihat pada Tabel 5.2 berikut :

Tabel 5.2. Implementasi Sub Menu Admin

Sub Menu	Deskripsi	Nama *File*
Halaman Utama	*File* program untuk melihat beranda	index.*php*
Daftar Siswa Baru	*File* program untuk melihat daftar siswa baru	daftarsiswa.*php*
Form Input Data Kelas	*File* program untuk input data kelas	inputdatakelas.*php*
Form Input Data Pelajaran	*File* program untuk input data pelajaran	inputdatapelajaran.*php*
Form Input Jadwal Pelajaran	*File* program untuk input jadwal pelajaran	inputjadwalpelajaran.*php*
Daftar Jadwal Pelajaran	*File* program untuk melihat daftar jadwal pelajaran	jadwalpelajaran.*php*
Logout	*File* program untuk admin keluar dari Aplikasi	logout.*php*

5.1.6. Implementasi Instalasi Program

Kegiatan pembuatan program terdiri dari pembentukan program-program yang akan saling mendukung dalam pembuatan aplikasi koperasi berbasis *web*. Untuk keperluan menampung data di *server*, kita memerlukan sebuah *database. Xampp* merupakan sebuah program aplikasi untuk membuat *database* yang umum digunakan pada *web server*. *Xampp* banyak digunakan karena kemampuannya dalam menampung data yang cukup besar dan proses *loading* yang cepat dalam pemanggilan data.

Implementasinya adalah sebagai berikut:

1. Instalkan program *web server* yang dimiliki.
2. Copy folder yayasan pada C:\Xampp\htdocs untuk programnya.
3. Copy folder Koperasi pada C:\Xampp\MySql\data untuk databasenya.
4. Ketikan pada URL *web browser* yang dimiliki yaitu http://localhost/yayasan.
5. Jika berhasil maka akan masuk pada tampilan awal *web* akademik.

5.1.7. Penggunaaan Program

Dalam penggunaan program ini langkah awalnya adalah memanggil tampilan awal sistem informasi akademik berbasis *web* dengan mengetikan alamat http://localhost/yayasan. Maka akan tampil halaman utama *User*.

1. Tampilan *User*

a. Tampilan Utama *user*

Merupakan tampilan awal pada saat *user* masuk dalam alamat *website*.



Gambar 5.1. Tampilan Utama *user*

b. Tampilan *Input* Pendaftaran Calon Siswa

Calon siswa baru daftar terlebih dahulu, apabila nilai UN kurang dari 70 maka sistem akan menolak apabila lebih dari 70 maka sistem akan menyimpan dalam database sebagai calon siswa.



Gambar 5.2. Tampilan *Input* Pendaftaran Siswa Baru

c. Tampilan Keterangan Sukses Pendaftaran Calon Siswa

Jika inputan data valid maka selanjutnya akan ditampilkan keterangan sukses pendaftaran calon siswa.



Gambar 5.3. Tampilan Keterangan Sukses Pendaftaran Calon Siswa

2. Tampilan Admin

a. Tampilan Login Tata Usaha

Pada keterangan sukses pendaftaran, calon siswa wajib menyimpan nomor pendaftar untuk daftar ulang. Tata usaha melakukan login untuk daftar ulang calon siswa.



Gambar 5.4. Tampilan Login Tata Usaha

Apabila lengkap maka Tata Usaha akan memvalidasi sebagai siswa baru, dengan mengklik tombol detail.

b. Tampilan Input Data Penerimaan Siswa



Gambar 5.5. Tampilan Input Data Penerimaan Siswa

c. Tampilan Detail Pendaftar

Apabila Tata Usaha sudah mengklik tombol detail maka akan muncul tampilan seperti di bawah ini, dan apabila persyaratan lengkap maka klik tombol setuju dan mengisi password untuk siswa baru sebagai login untuk *user*.



Gambar 5.6. Tampilan Detail Pendaftar

d. Tampilan Daftar Siswa

Setelah calon siswa disetujui sebagai siswa baru, maka siswa tersebut akan langsung mendapatkan kelas secara otomatis dan siswa dapat mengetahui dengan cara melakukan pencarian berdasarkan no pendaftar.



Gambar 5.7. Tampilan Daftar Siswa

e. Tampilan Login Kurikulum

Kurikulum melakukan login untuk membuat jadwal pelajaran.



Gambar 5.8. Tampilan Login Kurikulum

f. Tampilan Input Data Pelajaran

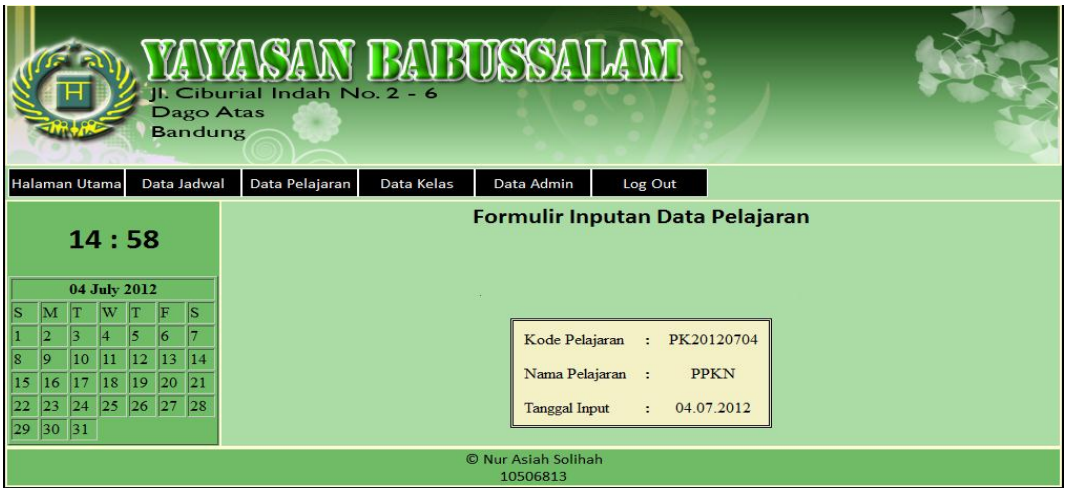
Setelah melakukan login, kemudian Kurikulum melakukan input data pelajaran dengan memilih mata pelajaran dan mengklik tombol proses untuk mendapatkan kode pelajaran.



Gambar 5.9. Tampilan Input Data Pelajaran

g. Tampilan Formulir Inputan Data Pelajaran

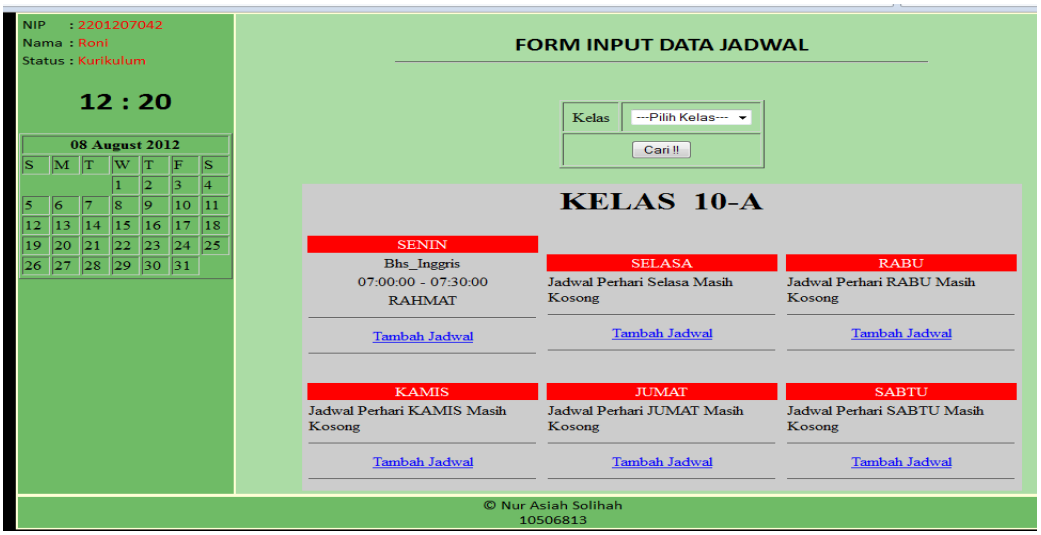
Setelah mengklik tombol proses maka akan keluar formulir inputan data pelajaran.



5.10. Tampilan Formulir Inputan Data Pelajaran

h. Tampilan Input Jadwal Pelajaran

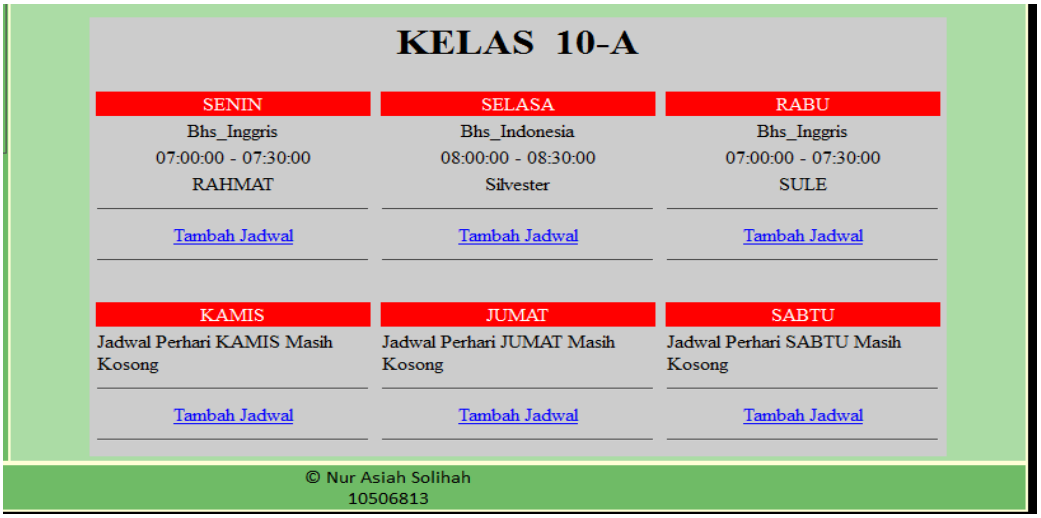
Kurikulum membuat jadwal pelajaran, dengan menginput jadwal terlebih dahulu.



Gambar 5.11 Tampilan Input Jadwal Pelajaran

i. Tampilan Jadwal Pelajaran

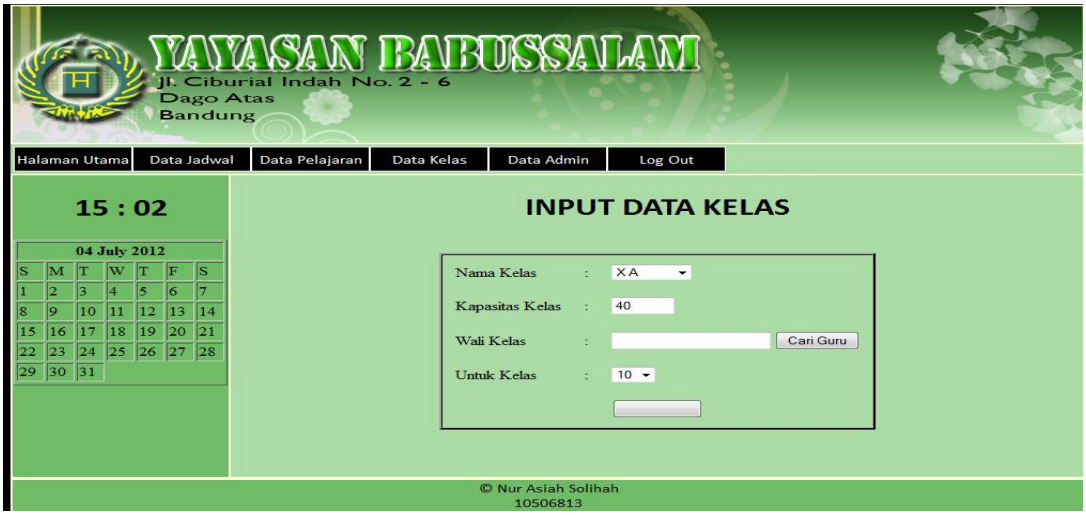
Setelah mengklik tombol simpan maka akan keluar output jadwal pelajaran.



Gambar 5.12 Tampilan Jadwal Pelajaran

j. Tampilan Input Data Kelas

Kurikulum melakukan input kelas, dapat dilihat pada gambar di bawah ini.



Gambar 5.13. Tampilan Input Data Kelas

k. Tampilan Daftar Kelas

Setelah input data kelas dan mengklik tombol simpan maka akan keluar daftar kelas.



Gambar 5.14. Tampilan Daftar Kelas

l. Tampilan Login untuk Siswa dan Guru

Siswa melakukan login agar dapat melihat jadwal pelajaran, nilai dan pembagian kelas.



Gambar 5.15 Tampilan Login Siswa dan Guru

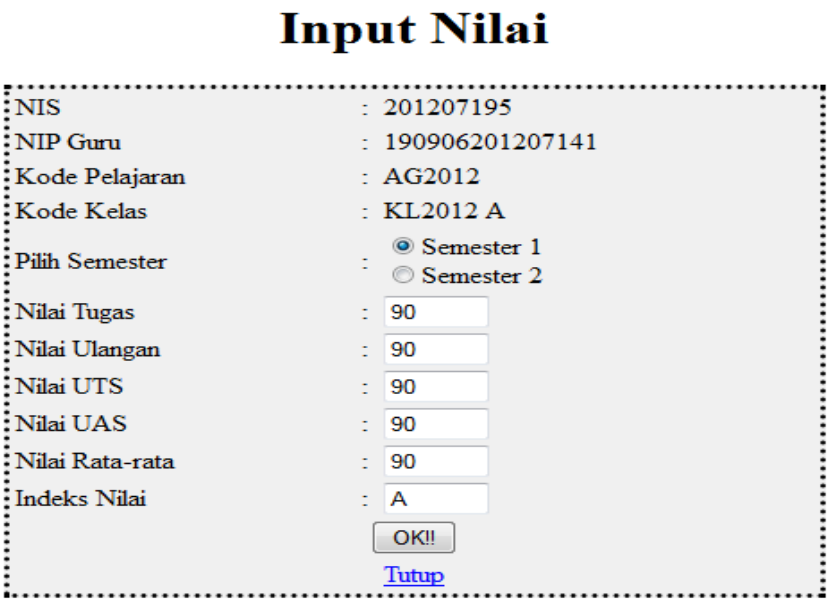
m. Tampilan Input Nilai

Guru melakukan login terlebih dahulu, kemudian memilih nama siswa untuk input nilai.:



Gambar 5.16 Tampilan Pilih Nama Siswa

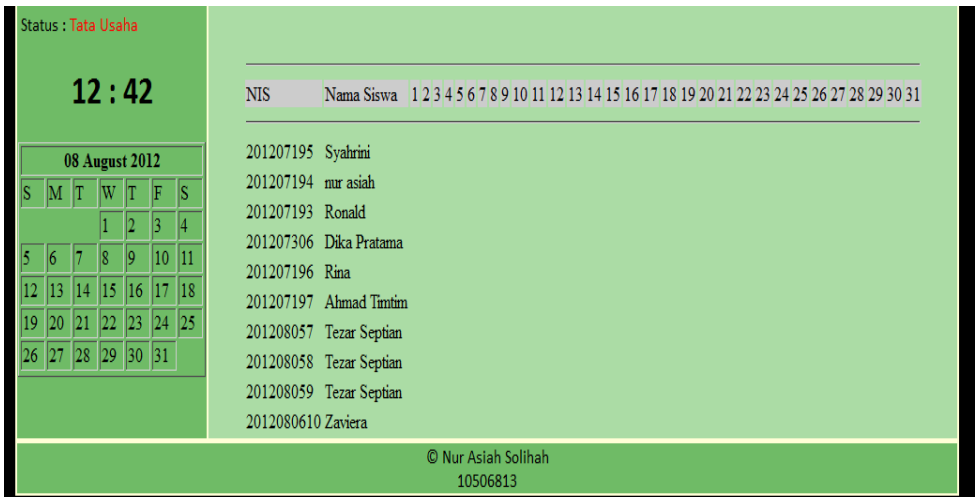
Setelah memilih nama siswa maka akan muncul tampilan seperti di bawah ini :



Gambar 5.17 Tampilan Input Nilai

n. Tampilan Daftar Nilai

Setelah nilai diinput, maka dapat dilihat daftar nilai untuk masing-masing kelas.



Gambar 5.18 Tampilan Daftar Nilai

5.2. Pengujian

Pengujian merupakan bagian yang penting dalam siklus pembangunan perangkat lunak. Pengujian dilakukan untuk menjamin kualitas dan juga mengetahui kelemahan dari perangkat lunak tersebut. Tujuan dari pengujian ini adalah untuk mengetahui kemampuan perangkat lunak dan menjamin bahwa perangkat lunak yang dibangun memiliki kualitas yang handal, yaitu merepresentasikan kajian pokok dari spesifikasi, analisis perancangan dan pengkodean dari perangkat lunak itu sendiri.

Dalam tahap ini penulis menggunakan pengujian *black box*, karena dengan pengujian *black box* kita dapat mengetahui apakah perangkat lunak yang dibuat dapat berfungsi dengan benar dan telah sesuai dengan yang diharapkan. Pengujian *black box* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika intern perangkat lunak. Pengujian *black box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dibangkitkan, dieksekusi pada perangkat lunak dan kemudian keluaran dari perangkat lunak dicek apakah telah sesuai dengan yang diharapkan dan berfokus pada persyaratan fungsional perangkat lunak yang dibuat.

5.2.1. Rencana Pengujian

Rencana pengujian *Web* Yayasan SMA Babussalam Bandung ini menggunakan data uji berupa sebuah data masukan dari *user* dan admin pada *website* yang telah dibuat sehingga akan menghasilkan keluaran yang diharapkan. Adapun pengujian *Web* ini penulis mempercayakan pada para tim penguji demi keakuratan hasil pengujian.

5.2.2. Kasus dan Hasil Pengujian

1. Pengujian *Login User*

Berikut ini tabel pengujian *login* untuk pengecekan *user* yang telah terdaftar

Tabel 5.3. Pengujian *login User* yang telah terdaftar

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data *login user* terdaftar. NIP : 05201201 *Password* : Zaviera	Tercantum pada textbox NIP & *password user*	Dapat mengisi *login user* sesuai yang diharapkan	[ ] diterima [ ] ditolak
Klik *Login*	Dapat masuk ke halaman selanjutnya untuk *user* yang telah terdaftar	Tombol *login* dapat berfungsi sesuai yang diharapkan	[ ] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data *login user* tidak terdaftar. NIP : XXXXXXXX *Password* : xxx	Tidak dapat *login* dan menampilkan pesan & Halaman *Login* kembali	*User* gagal *login* dan menampilkan pesan “Please check your admin name and your *password* again, *Login* Failed”	[ ] diterima [ ] ditolak

2. Pengujian *Login* Admin

Berikut ini adalah tabel pengujian *login* admin untuk pengecekan admin telah terdaftar .

Tabel 5.4. Pengujian *Login* Admin

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpilan
Admin : Meisya *Password* : 10506813	Tercantum pada *textbox username* & *password* admin	Dapat mengisi *login* admin sesuai yang diharapkan	[ ] diterima [ ] ditolak
Klik *Login*	Dapat masuk ke halaman selanjutnya untuk *user* admin yang telah terdaftar	Tombol *login* dapat berfungsi sesuai yang diharapkan	[ ] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpilan
Data *login* admin Admin : xxx *Password* : xxx	Tidak dapat *login* dan menampilkan pesan & kembali ke *form login*	Admin tidak dapat *login* dan menampilkan pesan “Please check your admin name and your *password* again, *Login* Failed”	[ ] diterima [ ] ditolak

3. Pengujian Pendaftaran *User*

Berikut ini adalah tabel pengujian pendaftaran *User* untuk pengecekan *user* yang belum terdaftar :

Tabel 5.5. Pengujian Pendaftaran *User*

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
User mengisi *Form* pendaftaran secara lengkap	Tercantum pada textbox *form* pendaftaran calon siswa	Dapat melakukan pengisian *form* pendaftaran sesuai dengan yang diharapkan	[ ] diterima [ ] ditolak
Klik proses	Menampilkan tabel inputan data *user* beserta pesan "masukan data sukses klik dibawah ini untuk melakukan *login*"	Data pengisian anggota tersimpan sesuai yang diharapkan	[ ] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Pengisian data anggota tidak lengkap	Tidak dapat melakukan pengisian *form* pendaftaran calon siswa	Mengeluarkan pesan "Maaf !! data yang anda masukan kurang lengkap. Mohon isi ulang kembali secara lengkap. Terima kasih"	[ ] diterima [ ] ditolak

5.2.3. Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian yang dilakukan dengan kasus uji diatas dapat ditarik kesimpulan bahwa perangkat lunak bebas dari kesalahan sintak dan secara fungsional dapat mengeluarkan hasil yang sesuai dengan yang diharapkan.