

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

A. Analisis Kebutuhan

1. Analisis Kebutuhan Fungsional

Peneliti melakukan pengumpulan data dengan observasi lapangan dan wawancara. Kemudian data dianalisis untuk mendapatkan daftar kebutuhan pengguna atau *user requirement list* yang dibutuhkan oleh SMK Negeri 2 Sewon. Observasi lapangan dilakukan saat melakukan praktik pengalaman lapangan (PPL) untuk mencari permasalahan yang ada di SMK N 2 Sewon. Wawancara dilakukan dengan Sunardi, S.Pd. selaku guru BK dan Arifah Suryaningsih, S.Pd. selaku ketua jurusan multimedia. Berikut adalah hasil analisis kebutuhan fungsional Sistem Informasi Presensi SMK Negeri 2 Sewon:

- a. Aplikasi mempunyai *server database online* untuk menyatukan data di Unit 1 dan Unit 2.
- b. Aplikasi berupa *website* dengan halaman utama yang digunakan oleh siswa sebagai antarmuka ketika melakukan presensi dan terdapat menu *Login* untuk akses masuk *user* ke dalam sistem informasi.
- c. Ada 4 jenis pengguna atau *user roles* yang mengakses aplikasi dan mempunyai hak akses tertentu sesuai jenisnya yaitu:

1) Admin

Hak akses yang dimiliki oleh admin adalah dapat menjalankan fungsi sebagai berikut:

- a) Melakukan *login*.

- b) Melakukan *logout*.
- c) Melakukan *reset password* petugas dan guru.
- d) Mengirim pesan kepada petugas dan guru.
- e) Melakukan pengaturan KOP untuk kepentingan cetak/*export* data rekap.
- f) Mengelola data siswa berupa lihat, cari, tambah, ubah, hapus, *upload*, dan cetak/*export*.
- g) Mengelola data guru berupa lihat, cari, tambah, ubah, hapus, *upload* dan cetak/*export*.
- h) Mengelola data petugas berupa lihat, cari, tambah, ubah, hapus, dan cetak/*export*.
- i) Mengelola data admin berupa lihat, cari, tambah, ubah, hapus, dan cetak/*export*.
- j) Mengelola data akun berupa lihat, cari, tambah, ubah, hapus, dan cetak/*export*.
- k) Mengelola data kelas berupa lihat, cari, tambah, ubah, hapus, dan cetak/*export*.
- l) Mengelola jadwal piket berupa lihat, cari, tambah, ubah, hapus, dan cetak/*export*.
- m) Mengelola jadwal sekolah berupa lihat, cari, tambah, ubah, hapus, *upload* dan cetak/*export*.
- n) Mengelola data rekap presensi berupa lihat, cari, dan cetak/*export*.
- o) Mengelola data rekap keterlambatan berupa lihat, cari, dan cetak/*export*.
- p) Mengelola data rekap perizinan berupa lihat, cari, dan cetak/*export*.
- q) Melihat profil.
- r) Mengubah profil.

s) Mengubah *password*.

2) Petugas

Hak akses yang dimiliki oleh petugas adalah dapat menjalankan fungsi sebagai berikut:

a) Melakukan *login*.

b) Melakukan *logout*.

c) Mengirim pesan kepada admin dan guru.

d) Mengelola data presensi per hari berupa lihat, cari, dan ubah.

e) Mengelola data perizinan per hari berupa lihat, cari, tambah, ubah, dan hapus.

f) Mengelola data keterlambatan per hari berupa lihat, cari, dan ubah.

g) Memverifikasi data presensi per hari.

h) Melihat data siswa

i) Melihat data guru

j) Melihat data jadwal piket

k) Melihat profil.

l) Mengubah profil.

m) Mengubah *password*.

3) Guru

Hak akses yang dimiliki oleh admin adalah dapat menjalankan fungsi sebagai berikut:

a) Melakukan *login*.

b) Melakukan *logout*.

c) Mengirim pesan kepada admin dan petugas.

- d) Melihat data presensi per hari.
 - e) Melihat data perizinan per hari.
 - f) Melihat data keterlambatan per hari.
 - g) Melihat data jadwal piket.
 - h) Melihat profil.
 - i) Mengubah profil.
 - j) Mengubah *password*.
- 4) Siswa

Siswa adalah pengguna yang melakukan presensi sebagai penginputan data ke dalam sistem informasi. Presensi dilakukan dengan sensor *fingerprint*.

2. Analisis Kebutuhan Software dan Hardware

- a. *Laptop/Komputer* dengan koneksi internet.
- b. *Arduino Uno Board, Ethernet Shield, Fingerprint Scanner*, dan Kabel LAN.
- c. *Web Server* Apache 2.4.3, PHP/5.4.7, dan MySQL 5.5.27.
- d. *Framework* CodeIgniter.
- e. Sublime Text 3.
- f. *Web Browser*.
- g. *Software Arduino*

B. Desain Sistem

1. Desain *Unified Modelling Language* (UML)

Keseluruhan rancangan kinerja program dideskripsikan menggunakan *Unified Modelling Language* (UML). Sebelum merancang UML, terlebih dahulu menentukan aktor untuk menentukan hak akses penggunaan sistem informasi

presensi berdasarkan fungsi yang ada pada program tersebut. Aktor dibagi menjadi 4 berdasarkan *user requirement list* yaitu: admin, petugas, guru, dan siswa. Diagram UML yang digunakan untuk mendeskripsikan alur kerja sistem adalah: *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram*.

a. Use Case Diagram

Use case diagram menggambarkan interaksi antara satu aktor atau lebih dengan sistem dan juga menjelaskan cara kerja sistem tersebut. *Use case diagram* masing-masing aktor digambarkan lebih jelas pada Gambar 24 sampai 26 (terlampir). Tabel 6 berikut adalah deskripsi aktor pada sistem informasi presensi:

Tabel 1. Deskripsi Aktor dalam Sistem Informasi Presensi

No.	Aktor	Deskripsi
1.	Admin	Admin adalah pengguna yang mempunyai <i>privilege</i> atau hak akses untuk mengelola data utama berupa data guru, data siswa, jadwal pelajaran, jadwal piket, data rekap presensi, data rekap keterlambatan, dan data rekap perizinan. Admin mempunyai <i>privilege</i> untuk melihat, menambah, menghapus, mengubah, <i>upload</i> dan mencetak/export data. Admin juga dapat mengirim pesan kepada petugas dan guru. Untuk dapat mengelola data tersebut diperlukan <i>login</i> ke dalam sistem terlebih dahulu.
2.	Petugas	Petugas adalah pengguna yang mempunyai <i>privilege</i> untuk mengelola data presensi per hari, data keterlambatan per hari, dan data perizinan perhari. Petugas dapat melihat, menambah, menghapus, mengubah, dan memverifikasi data yang masuk. Petugas dapat mengirim pesan kepada guru dan admin. Petugas juga diharuskan untuk <i>login</i> ke dalam sistem terlebih dahulu.
3.	Guru	Guru adalah pengguna yang tidak mendapat <i>privilege</i> . Guru hanya dapat melihat data presensi perhari, data keterlambatan per hari, dan data perizinan per hari. Guru dapat melakukan pengiriman pesan untuk konfirmasi kepada petugas dan admin. Guru perlu melakukan <i>login</i> terlebih dahulu ke dalam sistem.

No.	Aktor	Deskripsi
4.	Siswa	Siswa adalah pengguna yang menggunakan alat presensi untuk pengambilan data.

b. Class Diagram

Class diagram menggambarkan struktur sistem pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki atribut yaitu variabel-variabel yang dimiliki oleh kelas dan operasi yang merupakan fungsi-fungsi yang dimiliki oleh suatu kelas. Langkah pembuatan class diagram dimulai dari pembuatan nama kelas, kemudian pada masing-masing kelas dibuat pendefinisian atribut-atribut kelas. Langkah terakhir yaitu pembuatan operasi. Framework Codeigniter dengan *design pattern Model-View-Controller (MVC)*, sehingga setiap kelas *controller* yang dibuat harus turunan dari *CI_Controller* sedangkan setiap kelas *model* yang dibuat harus turunan dari *CI_Model*. Desain class diagram dari Sistem Informasi Presensi Siswa ditunjukkan pada gambar 27 (terlampir).

c. Sequence Diagram

1) Sequence Diagram Login

Diagram ini menjelaskan tentang proses *user* masuk ke dalam sistem menggunakan *username* dan *password* sebagai *authenticator*. Pertama-tama *user* memasukkan *username* dan *password* pada halaman *login (view)*, kemudian mengakses *controller* untuk mengaktifkan fungsi aksi_login(). Fungsi tersebut digunakan untuk mengakses fungsi cek_login() pada *model* untuk melakukan pengecekan data di *database*. Di *database username* dan *password* yang dimasukkan divalidasi. Jika data ditemukan maka *user* akan di *direct* ke halaman

dashboard. Jika tidak ditemukan maka akan muncul notifikasi dan *redirect* ke halaman *login*. *Sequence diagram Login* ditunjukkan dalam gambar 28 (terlampir).

2) *Sequence Diagram* Tambah Data

Diagram ini menjelaskan tentang proses memasukkan atau menambahkan data baru. Setelah semua data dimasukkan pada form yang tersedia, *view* akan mengakses *controller* untuk mengaktifkan fungsi *create()*. Kemudian *controller* akan melakukan fungsi *create()* untuk mendapatkan data dari *user*. Kemudian *controller* akan mengakses model untuk mengaktifkan fungsi *save()* untuk menyimpan data tersebut ke dalam tabel yang telah ditentukan pada *database*. Jika data berhasil disimpan ke *database*, halaman akan *redirect* ke halaman awal. Jika gagal akan muncul notifikasi. *Sequence diagram* tambah data ditunjukkan dalam gambar 29 (terlampir).

3) *Sequence Diagram* Ubah Data

Diagram ini menjelaskan proses mengubah data yang sudah ada. Setelah *user* mengubah data yang salah dengan data yang benar pada *form* yang ada pada *view*, kemudian *view* akan mengakses *controller* untuk mengaktifkan fungsi *ajax_update*. *Controller* melakukan fungsi *ajax_update()* untuk menerima data dari *view* kemudian mengakses fungsi *update()* pada *model*. Fungsi *update()* akan melakukan perintah *update* yaitu mengganti data yang sudah ada dengan data yang baru pada *record* yang dipilih pada tabel tertentu di *database*. Jika data berhasil diupdate, halaman akan *redirect* ke halaman penyimpanan data. Jika gagal maka akan muncul notifikasi untuk *user*. *Sequence diagram* ubah data ditunjukkan dalam gambar 30 (terlampir).

4) *Sequence Diagram* Hapus Data

Diagram ini menjelaskan proses menghapus data pada sistem informasi. Setelah *user* memilih data yang akan di hapus pada halaman tampil data (*view*) kemudian *view* akan mengakses *controller* untuk mengaktifkan fungsi *ajax_delete()*. *Controller* akan melakukan fungsi *ajax_delete()* untuk menerima data berupa 'id' data yang akan dihapus kemudian mengakses *Model* untuk mengaktifkan fungsi *delete_by_id()*. *Model* melakukan fungsi *delete_by_id()* untuk memberikan perintah pada database untuk menghapus data pada tabel yang ditentukan berdasarkan 'id' yang sama dengan 'id' yang dikirimkan oleh *Model*. Jika data berhasil dihapus, halaman akan *redirect* ke halaman tampil data. Jika gagal menghapus data maka akan muncul notifikasi untuk *user*. *Sequence diagram* hapus data ditunjukkan dalam gambar 31 (terlampir).

5) *Sequence Diagram* Impor Data

Diagram ini menjelaskan proses memasukkan data ke dalam sistem dari *file external* berupa *file* Microsoft Excel. *User* memilih menu 'Upload' pada halaman tampil data (*view*). Kemudian *View* akan mengakses *Model* untuk mengaktifkan fungsi *form()*. *Model* melakukan fungsi *form()* untuk kembali ke *View* untuk menampilkan halaman *form* impor data. Setelah itu *user* memilih *file* yang akan diimpor. Pada halaman *form* impor data terdapat tombol *preview*. Ketika *user* klik tombol tersebut maka data yang ada pada excel akan dimasukkan ke dalam sistem (bukan *database*) kemudian ditampilkan pada *user* untuk memastikan apakah data tersebut sudah benar. Jika sudah, *user* akan klik tombol Impor untuk mengakses *Controller* dan mengaktifkan fungsi *import()*. Fungsi ini dilakukan untuk

mengkonversi dan menampung data dari excel kemudian memanggil fungsi `insert_multiple()` pada *Model*. Fungsi `insert_multiple()` melakukan perintah ke *database* untuk melakukan fungsi menambahkan banyak data ke dalam tabel dan kolom yang telah ditentukan. Jika berhasil maka halaman akan *redirect* ke halaman tampil data, jika gagal maka akan muncul tampilan notifikasi *error* dan penyebab *error*-nya. *Sequence diagram* impor data ditunjukkan dalam gambar 32 (terlampir).

6) *Sequence Diagram* Lihat Data

Diagram ini menjelaskan proses *user* menampilkan data yang ingin dilihat. Langkah pertama adalah *user* memilih data yang akan ditampilkan pada menu (*View*). Kemudian *view* akan memanggil fungsi `index()` pada *Controller* untuk mengakses fungsi `view()` pada *Model*. Fungsi `view()` berfungsi untuk memberikan perintah pada *database* untuk menampilkan data pada tabel yang ditentukan. Kemudian menampilkan kembali hasilnya ke *View*. *Sequence diagram* lihat data ditunjukkan dalam gambar 33 (terlampir).

7) *Sequence Diagram* Export Data

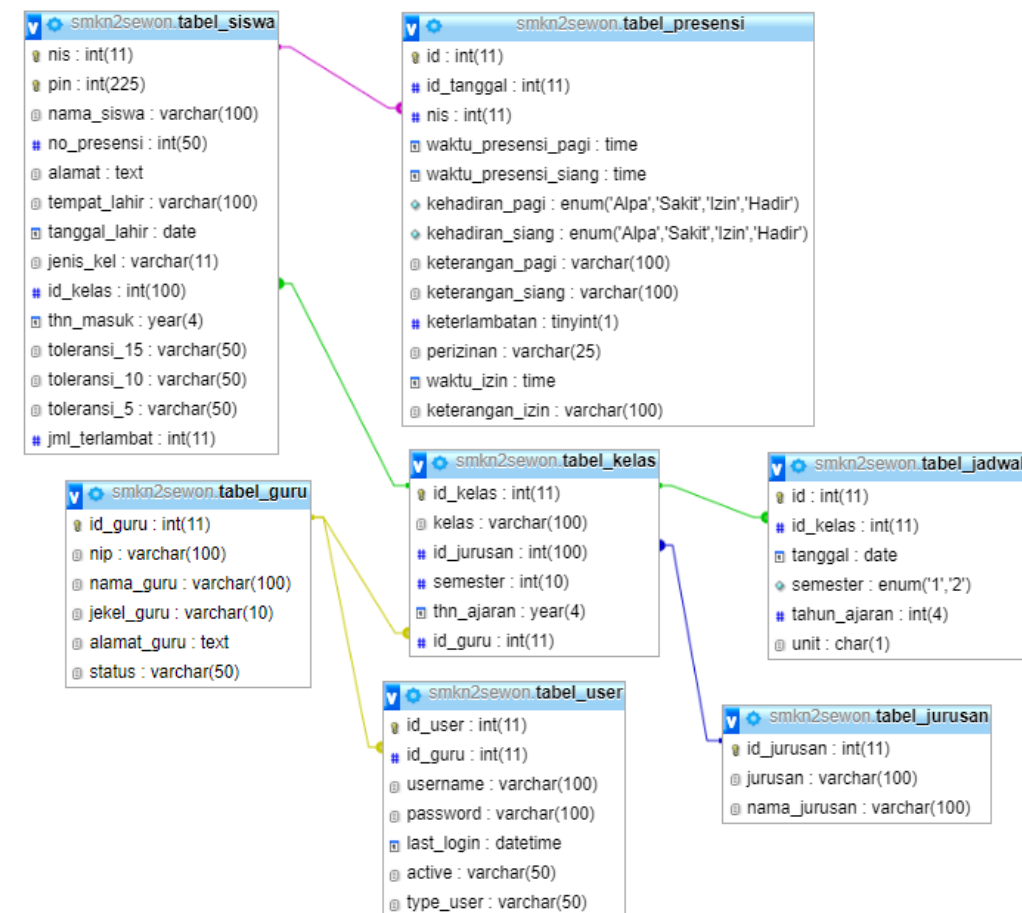
Diagram ini menjelaskan proses *user* mengunduh data yang ada di dalam sistem menjadi *file* excel. Langkah pertama adalah *user* memilih menu *export* pada halaman tampil data (*View*). Kemudian dari *View* akan memanggil fungsi `excel_data()` pada *Controller* untuk menampung data yang akan di-*export* dari *View*. Fungsi `excel_data()` memanggil fungsi `get_data()` pada *Model*. Fungsi ini dilakukan untuk memberikan perintah kepada *database* untuk mencocokkan data dari *View* dengan data yang ada dalam tabel yang ditentukan kemudian

mengambilnya untuk kemudian di-export ke dalam *file* excel. *Sequence diagram* ekspor data ditunjukkan dalam gambar 34 (terlampir).

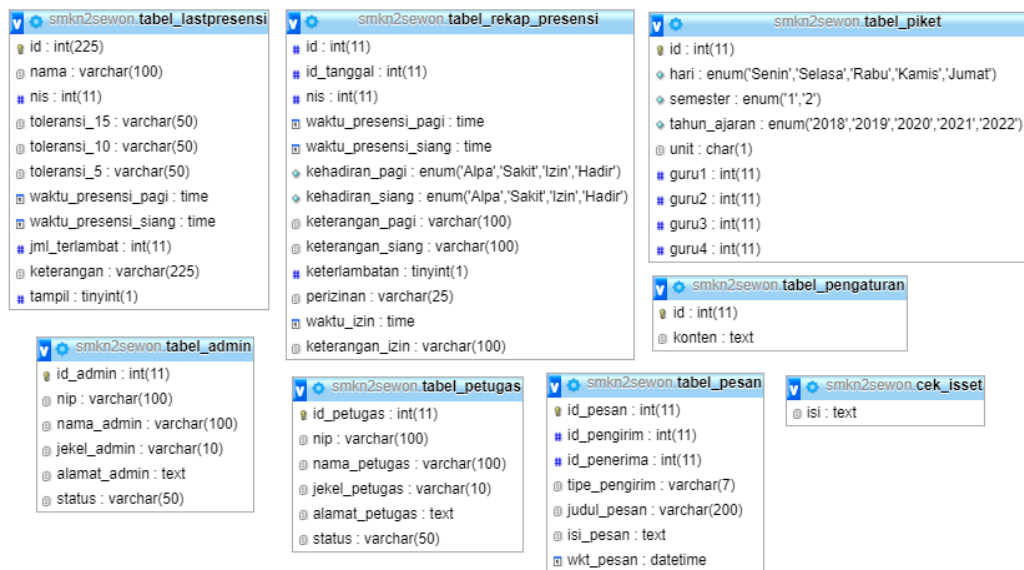
2. Desain Database

Database yang digunakan dalam sistem informasi presensi ini adalah MySQL.

Gambar 5 dan 6 berikut adalah rancangan hubungan antar tabel pada MySQL yang menggambarkan alur penyimpanan data pada sistem informasi presensi:



Gambar 1. Desain Basis Data



Gambar 2. Desain Basis Data

3. Desain Interface

Desain *interface* atau *user interface* merupakan rancangan awal tampilan aplikasi *website* ketika diakses oleh *user* atau pengguna. Rancangan dibuat untuk memudahkan ketika *website* mulai diimplementasikan dalam bentuk *code* html dan css. Desain *interface* berupa *prototype* sederhana yang mendeskripsikan *layout* halaman *website* beserta kontennya dan juga navigasi menu dan *button* yang efektif dan efisien ketika digunakan oleh *user*. Hasil desain *user interface* terlampir pada gambar 36 sampai dengan gambar 66.

C. Implementasi

Setelah menentukan *user requirements list* dan merancang tampilan antarmuka pengguna (*user interface*) berdasarkan *user requirements list*, tahap berikutnya adalah implementasi. Implementasi adalah penerjemahan desain *user interface* ke dalam bahasa pemrograman. Tahap implementasi terbagi menjadi 3 bagian yaitu

implementasi *user interface*, implementasi *database* (basis data), dan implementasi program.

1. Hasil Implementasi *User Interface*

Aplikasi *website* sistem informasi presensi dibangun menggunakan *framework* CodeIgniter dengan desain *user interface*-nya diimplementasikan ke dalam html dan css menggunakan *framework* css Bootstrap. Berikut adalah hasil implementasi *user interface* sistem informasi presensi:

a. Implementasi *Landing Page* untuk siswa

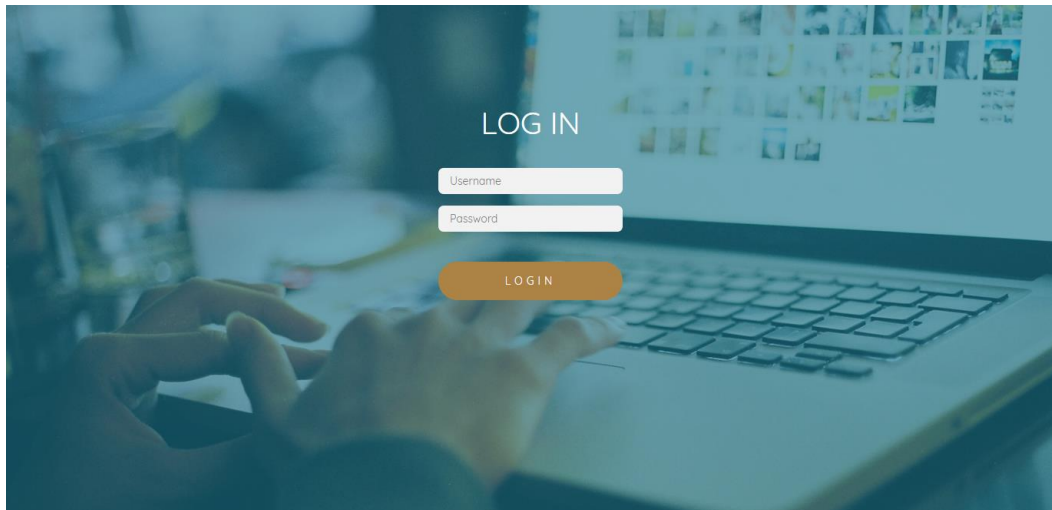
Gambar 7 berikut adalah halaman *Landing Page* untuk sistem informasi presensi siswa. Halaman tersebut merupakan halaman utama ketika siswa akan melakukan presensi. Setelah siswa berhasil presensi dengan sidik jari, data siswa tersebut akan muncul pada halaman landing page beserta keterangan presensi dan terlambat.



Gambar 3. Implementasi *Landing Page* untuk Siswa

b. Implementasi *Login Page*

Gambar 8 berikut adalah halaman implementasi *Login*. Halaman tersebut digunakan oleh *user* (admin, petugas, dan guru) untuk masuk ke dalam sistem dan mengakses sistem tersebut.



Gambar 4 Implementasi Halaman *Login*

c. Implementasi Halaman Profil

Gambar 67 (terlampir) adalah implementasi halaman profil. Fungsi dari halaman profil adalah ketika user ingin melihat profil ataupun mengubah profil dan *password*, *user* dapat melakukannya pada halaman tersebut.

d. Implementasi Halaman Pengaturan

Gambar 68 (terlampir) adalah implementasi halaman pengaturan. Halaman ini digunakan untuk mengubah KOP sekolah yang dipakai untuk pencetakan data rekap atau data lain yang ingin dicetak atau *export*.

e. Implementasi Halaman *Dashboard*

Gambar 69 (terlampir) adalah implementasi halaman *dashboard*. Halaman ini adalah halaman yang pertama kali dituju ketika *user* berhasil *login* ke dalam sistem. Halaman ini berisi informasi tentang data terbaru yang masuk pada hari tersebut.

f. Implementasi Halaman Pesan

Gambar 70 sampai 72 (terlampir) adalah implementasi halaman Pesan Baru, Pesan Masuk, dan Pesan Keluar. Berfungsi untuk mengirim pesan baru kepada user lain, membalas pesan, memeriksa pesan masuk, dan pesan keluar

g. Implementasi Halaman Data Presensi per Hari

Gambar 73 (terlampir) adalah implementasi halaman data presensi per hari. Halaman ini hanya dapat diakses oleh petugas piket. Halaman ini digunakan untuk menampilkan data siswa, waktu presensi, keterangan presensi, keterangan toleransi presensi, dan keterangan terlambat. Petugas piket dapat mengubah keterangan presensi dengan memilih tombol-tombol yang tersedia.

h. Implementasi Halaman Data Perizinan per Hari

Gambar 74 (terlampir) adalah implementasi halaman data perizinan per hari. Halaman ini hanya dapat diakses oleh petugas piket. Halaman ini digunakan untuk menampilkan data siswa yang tidak hadir pada hari tersebut. Petugas juga dapat menambahkan data izin ketika ada siswa yang ingin izin keluar sekolah atau ada keperluan lain.

i. Implementasi Halaman Data Keterlambatan per Hari

Gambar 75 (terlampir) adalah implementasi halaman keterlambatan per hari. Halaman ini hanya dapat diakses oleh petugas piket. Halaman ini berfungsi untuk

menampilkan data siswa yang terlambat pada hari tersebut. Petugas dapat menambahkan dan mengubah keterangan alasan siswa tersebut terlambat.

j. Implementasi Halaman Verifikasi Data Presensi

Gambar 76 (terlampir) adalah implementasi halaman verifikasi data presensi per hari. Halaman ini hanya dapat diakses oleh petugas piket untuk memverifikasi data presensi pada hari tersebut. Verifikasi dilakukan ketika kegiatan belajar mengajar pada hari tersebut selesai. Sebelumnya petugas harus memeriksa data presensi terlebih dahulu untuk memastikan tidak ada kesalahan data. Setelah dipastikan semua data benar, baru verifikasi dapat dilakukan. Setelah diverifikasi nantinya data presensi pada hari tersebut akan masuk ke data rekapitulasi.

k. Implementasi Halaman Rekap Data

Gambar 77 dan 78 (terlampir) merupakan implementasi halaman rekap untuk data presensi, perizinan, dan keterlambatan. Yang dapat mengakses halaman ini hanya admin. Halaman ini digunakan untuk keperluan pembuatan laporan harian, bulanan, atau semester. Terdapat dua macam halaman rekap yaitu halaman rekap per kelas dan halaman rekap per siswa.

l. Implementasi Halaman Master Data Siswa

Gambar 79 (terlampir) adalah implementasi halaman data siswa. Halaman ini digunakan untuk mengelola data siswa meliputi tambah data, ubah, hapus, unggah *external file*, dan unduh atau cetak. Semua *user* dapat mengakses halaman

tersebut tetapi hanya admin yang dapat mengelola data. Selain admin hanya dapat melihat.

m. Implementasi Halaman Master Data Guru

Gambar 80 (terlampir) adalah implementasi halaman data guru. Halaman ini digunakan untuk mengelola data siswa meliputi tambah data, ubah, hapus, unggah *external file*, dan unduh atau cetak. Semua user dapat mengakses halaman tersebut tetapi hanya admin yang dapat mengelola data. Selain admin hanya dapat melihat.

n. Implementasi Halaman Master Data Petugas

Gambar 81 (terlampir) adalah implementasi halaman data petugas. Halaman ini digunakan untuk mengelola data siswa meliputi tambah data, ubah, hapus, unggah *external file*, dan unduh atau cetak. Semua user dapat mengakses halaman tersebut tetapi hanya admin yang dapat mengelola data. Selain admin hanya dapat melihat.

o. Implementasi Halaman Master Data Admin

Gambar 82 (terlampir) adalah implementasi halaman data admin. Halaman ini digunakan untuk mengelola data siswa meliputi tambah data, ubah, hapus, unggah

external file, dan unduh atau cetak. Semua user dapat mengakses halaman tersebut tetapi hanya admin yang dapat mengelola data. Selain admin hanya dapat melihat.

p. Implementasi Halaman Master Data Akun

Gambar 83 (terlampir) adalah implementasi halaman data akun admin, guru, dan petugas. Halaman ini digunakan untuk mengelola data siswa meliputi tambah data, ubah, hapus, unggah *external file*, dan unduh halaman data presensi per hari.

q. Implementasi Halaman Master Data Jadwal Kelas

Gambar 84 (terlampir) adalah implementasi halaman data kelas. Halaman ini digunakan untuk mengelola data kelas meliputi tambah data, ubah, hapus, unggah *external file*, dan unduh.

r. Implementasi Halaman Master Data Jadwal Sekolah

Gambar 85 di samping adalah implementasi halaman data jadwal sekolah. Halaman ini digunakan untuk mengelola data siswa meliputi tambah data, ubah, hapus, unggah *external file*, dan unduh atau cetak. Semua user dapat mengakses halaman tersebut tetapi hanya admin yang dapat mengelola data. Selain admin hanya dapat melihat.

s. Implementasi Halaman Master Jadwal Piket

Gambar 86 di samping adalah implementasi halaman data jadwal piket. Halaman ini digunakan untuk mengelola data siswa meliputi tambah data, ubah, hapus, unggah *external file*, dan unduh atau cetak. Semua user dapat mengakses

halaman tersebut tetapi hanya admin yang dapat mengelola data. Selain admin hanya dapat melihat.

2. Hasil Implementasi *Database*

Gambar 9 sampai dengan 31 berikut merupakan hasil implementasi perancangan basis data sistem informasi presensi dalam tabel MySQL:

a. Implementasi Database Tabel Admin

Gambar 9 berikut merupakan tampilan tabel pada MySQL yang digunakan untuk menyimpan data admin.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id_admin	int(11)			No	None	AUTO_INCREMENT	Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
2	nip	varchar(100)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
3	nama_admin	varchar(100)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
4	jekel_admin	varchar(10)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
5	alamat_admin	text	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
6	status	varchar(50)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext

Gambar 5. Implementasi *Database* Tabel Admin

b. Implementasi Database Tabel Guru

Gambar 10 berikut merupakan tampilan tabel pada MySQL yang digunakan untuk menyimpan data guru.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id_guru	int(11)			No	None	AUTO_INCREMENT	Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
2	nip	varchar(100)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
3	nama_guru	varchar(100)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
4	jekel_guru	varchar(10)	latin1_swedish_ci	Yes	NULL			Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
5	alamat_guru	text	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
6	status	varchar(50)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext

Gambar 6. Implementasi *Database* Tabel Guru

c. Implementasi Database Tabel Jurusan

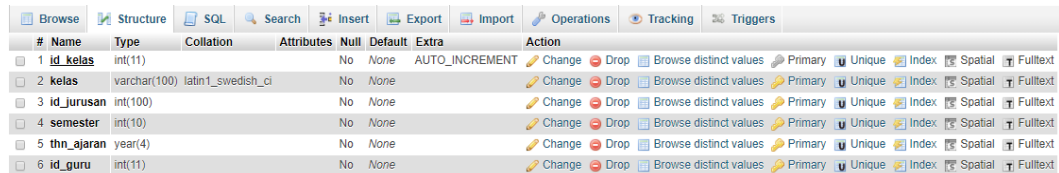
Gambar 11 berikut merupakan tampilan tabel pada MySQL yang digunakan untuk menyimpan data jurusan.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id_jurusan	int(11)			No	None	AUTO_INCREMENT	Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
2	jurusan	varchar(100)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
3	nama_jurusan	varchar(100)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext

Gambar 7. Implementasi *Database* Tabel Jurusan

d. Implementasi Database Tabel Kelas

Gambar 12 berikut merupakan tampilan tabel pada MySQL yang digunakan untuk menyimpan data kelas.

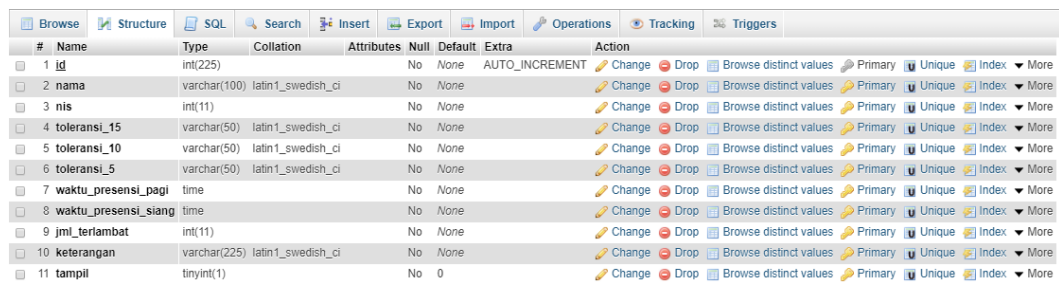


#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id_kelas	int(11)			No	None	AUTO_INCREMENT	Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
2	kelas	varchar(100)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
3	id_jurusan	int(100)			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
4	semester	int(10)			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
5	thn_ajaran	year(4)			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
6	id_guru	int(11)			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext

Gambar 8. Implementasi *Database* Tabel Kelas

e. Implementasi Database Tabel Lastpresensi

Gambar 13 berikut merupakan tampilan tabel pada MySQL yang digunakan untuk menyimpan data presensi terbaru.

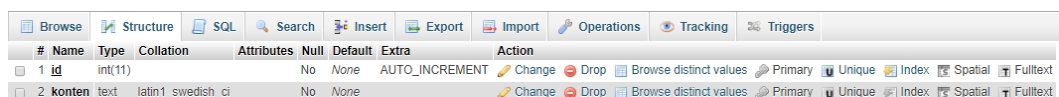


#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(225)			No	None	AUTO_INCREMENT	Change Drop Browse distinct values Primary Unique Index More
2	nama	varchar(100)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index More
3	nis	int(11)			No	None		Change Drop Browse distinct values Primary Unique Index More
4	toleransi_15	varchar(50)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index More
5	toleransi_10	varchar(50)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index More
6	toleransi_5	varchar(50)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index More
7	waktu_presensi_pagi	time			No	None		Change Drop Browse distinct values Primary Unique Index More
8	waktu_presensi_siang	time			No	None		Change Drop Browse distinct values Primary Unique Index More
9	jml_terlambat	int(11)			No	None		Change Drop Browse distinct values Primary Unique Index More
10	keterangan	varchar(225)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index More
11	tampil	tinyint(1)			No	0		Change Drop Browse distinct values Primary Unique Index More

Gambar 9. Implementasi *Database* Tabel Lastpresensi

f. Implementasi Database Tabel Pengaturan

Gambar 14 berikut merupakan tampilan tabel pada MySQL yang digunakan untuk menyimpan data pengaturan.



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(11)			No	None	AUTO_INCREMENT	Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
2	konten	text	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext

Gambar 10. Implementasi *Database* Tabel Pengaturan

g. Implementasi Database Tabel Pesan

Gambar 15 berikut merupakan tampilan tabel pada MySQL yang digunakan untuk menyimpan data pesan.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id_pesanan	int(11)			No	None	AUTO_INCREMENT	Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
2	id_pengirim	int(11)			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
3	id_penerima	int(11)			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
4	tipe_pengirim	varchar(7)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
5	judul_pesanan	varchar(200)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
6	isi_pesanan	text	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
7	wkt_pesanan	datetime			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext

Gambar 11. Implementasi *Database* Tabel Pesan

h. Implementasi Database Tabel Petugas

Gambar 16 berikut merupakan tampilan tabel pada MySQL yang digunakan untuk menyimpan data petugas.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id_petugas	int(11)			No	None	AUTO_INCREMENT	Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
2	nip	varchar(100)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
3	nama_petugas	varchar(100)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
4	jkel_petugas	varchar(10)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
5	alamat_petugas	text	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
6	status	varchar(50)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext

Gambar 12. Implementasi *Database* Tabel Petugas

i. Implementasi Database Tabel Piket

Gambar 17 berikut merupakan tampilan tabel pada MySQL yang digunakan untuk menyimpan data jadwal piket.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(11)			No	None	AUTO_INCREMENT	Change Drop Browse distinct values Primary Unique Index Spatial More
2	hari	enum('Senin', 'Selasa', 'Rabu', 'Kamis', 'Jumat')	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial More
3	semester	enum('1', '2')	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial More
4	tahun_ajaran	enum('2018', '2019', '2020', '2021', '2022')	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial More
5	unit	char(1)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial More
6	guru1	int(11)			No	None		Change Drop Browse distinct values Primary Unique Index Spatial More
7	guru2	int(11)			No	None		Change Drop Browse distinct values Primary Unique Index Spatial More
8	guru3	int(11)			No	None		Change Drop Browse distinct values Primary Unique Index Spatial More
9	guru4	int(11)			No	None		Change Drop Browse distinct values Primary Unique Index Spatial More

Gambar 13. Implementasi *Database* Tabel Piket

j. Implementasi Database Tabel Presensi

Gambar 18 berikut merupakan tampilan tabel pada MySQL yang digunakan untuk menyimpan data presensi yang sudah diubah dan sudah siap diverifikasi.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(11)			No	None	AUTO_INCREMENT	Change Drop Browse distinct values Primary Unique Index ▼ More
2	id_tanggal	int(11)			No	None		Change Drop Browse distinct values Primary Unique Index ▼ More
3	nis	int(11)			No	None		Change Drop Browse distinct values Primary Unique Index ▼ More
4	waktu_presensi_pagi	time			No	None		Change Drop Browse distinct values Primary Unique Index ▼ More
5	waktu_presensi_siang	time			No	None		Change Drop Browse distinct values Primary Unique Index ▼ More
6	kehadiran_pagi	enum('Alpa', 'Sakit', 'Izin', 'Hadir')	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index ▼ More
7	kehadiran_siang	enum('Alpa', 'Sakit', 'Izin', 'Hadir')	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index ▼ More
8	keterangan_pagi	varchar(100)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index ▼ More
9	keterangan_siang	varchar(100)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index ▼ More
10	keterlambatan	tinyint(1)			No	0		Change Drop Browse distinct values Primary Unique Index ▼ More
11	perizinan	varchar(25)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index ▼ More
12	waktu_izin	time			No	None		Change Drop Browse distinct values Primary Unique Index ▼ More
13	keterangan_izin	varchar(100)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index ▼ More

Gambar 14. Implementasi *Database* Tabel Presensi

k. Implementasi Database Tabel Rekap Presensi

Gambar 19 berikut merupakan tampilan tabel pada MySQL yang digunakan untuk menyimpan data rekap presensi. Data ini dipindah dari tabel presensi ketika user sudah memverifikasi data presensi pada sistem informasi presensi.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(11)			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
2	id_tanggal	int(11)			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
3	nis	int(11)			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
4	waktu_presensi_pagi	time			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
5	waktu_presensi_siang	time			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
6	kehadiran_pagi	enum('Alpa', 'Sakit', 'Izin', 'Hadir')	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
7	kehadiran_siang	enum('Alpa', 'Sakit', 'Izin', 'Hadir')	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
8	keterangan_pagi	varchar(100)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
9	keterangan_siang	varchar(100)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
10	keterlambatan	tinyint(1)			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
11	perizinan	varchar(25)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
12	waktu_izin	time			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
13	keterangan_izin	varchar(100)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext

Gambar 15. Implementasi *Database* Tabel Rekap Presensi

l. Implementasi Database Tabel Siswa

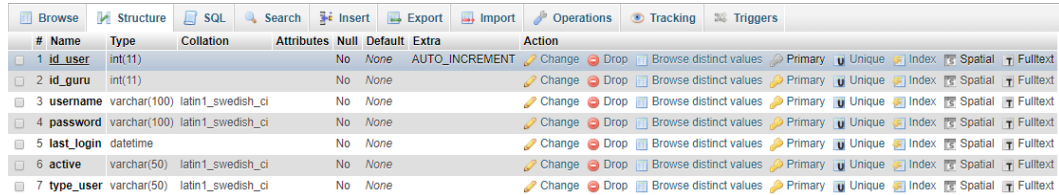
Gambar 20 berikut merupakan tampilan tabel pada MySQL yang digunakan untuk menyimpan data siswa.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	nis	int(11)			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
2	pin	int(225)		Yes	NULL			Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
3	nama_siswa	varchar(100)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
4	no_presensi	int(50)			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
5	alamat	text	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
6	tempat_lahir	varchar(100)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
7	tanggal_lahir	date			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
8	jenis_kel	varchar(11)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
9	id_kelas	int(100)			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
10	thn_masuk	year(4)			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
11	toleransi_15	varchar(50)	latin1_swedish_ci		No	Akuf		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
12	toleransi_10	varchar(50)	latin1_swedish_ci		No	Akuf		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
13	toleransi_5	varchar(50)	latin1_swedish_ci		No	Akuf		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
14	jml_terlambat	int(11)			No	0		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext

Gambar 16. Implementasi *Database* Tabel Siswa

m. Implementasi Database Tabel *User*

Gambar 21 berikut merupakan tampilan tabel pada MySQL yang digunakan untuk menyimpan data akun *user* yang menggunakan sistem informasi presensi.



The screenshot shows the MySQL database structure for a table named 'User'. The table has 7 columns: id_user, id_guru, username, password, last_login, active, and type_user. The columns are defined with their respective data types, collations, and attributes. The 'id_user' column is the primary key and is auto-incrementing. The 'username' and 'password' columns are indexed and have unique constraints. The 'last_login' column is indexed. The 'active' column is indexed and has a unique constraint. The 'type_user' column is indexed.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id_user	int(11)			No	None	AUTO_INCREMENT	Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
2	id_guru	int(11)			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
3	username	varchar(100)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
4	password	varchar(100)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
5	last_login	datetime			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
6	active	varchar(50)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
7	type_user	varchar(50)	latin1_swedish_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext

Gambar 17. Implementasi *Database* Tabel *User*

3. Hasil Implementasi Program

Proses pengembangan *website* sistem informasi presensi siswa menggunakan *text editor* Sublime Text 3 dan menggunakan *framework* PHP yaitu CodeIgniter. CodeIgniter adalah PHP *framework* yang mempunyai pola desain *Model-View-Controller*. *Model* berfungsi untuk mengelola data dari aplikasi ke *database* atau sebaliknya. *View* berfungsi untuk menampilkan data. Dan *Controller* berfungsi untuk menampilkan data dari *view* untuk diolah *Model* atau mengolah data dari *Model* untuk ditampilkan ke *View*. Gambar 113 sampai 115 (terlampir) adalah salah satu implementasi pola MVC untuk halaman data presensi per hari.

D. Pengujian

Tahap pengujian sistem dilakukan dengan menilai beberapa aspek dalam ISO/IEC 9126. Aspek yang dipilih untuk pengujian karakteristik *website* sistem informasi presensi didasarkan pada model WebQEM (*Web Quality Evaluation Method*) yang dikembangkan oleh Olsina yaitu aspek *functionality*, *reliability*, *efficiency*, dan *usability*. Berikut adalah hasil pengujian aspek-aspek tersebut:

1. Aspek *Functionality*

Pengujian aspek *functionality* dilakukan oleh ahli yang bekerja dalam bidang *IT* atau yang memahami tentang pengembangan *website* berjumlah 3 orang menggunakan angket yang berisi *test case* berupa daftar fungsi yang terdapat dalam *website* sistem informasi yang disusun berdasarkan kebutuhan *user*. Tabel 7 berikut adalah hasil pengujian aspek *functionality*:

Tabel 2. Hasil Pengujian *Functionality*

No.	Fungsi	Hasil	
		Sukses	Gagal
1.	Melakukan <i>log in</i> ke dalam sistem	3	0
2.	Melakukan <i>log out</i> dari sistem	3	0
3.	Melihat profil	3	0
4.	Mengubah profil	3	0
5.	Mengubah <i>password</i>	3	0
6.	Mengubah <i>username</i>	3	0
A	Administrator		
7.	Melihat data siswa	3	0
8.	Mencari data siswa	3	0
9.	Menambah data siswa	3	0
10.	Mengubah data siswa	3	0
11.	Menghapus data siswa	3	0
12.	<i>Upload</i> data siswa	3	0
13.	Mencetak/ <i>export</i> data siswa	3	0
14.	Melihat data guru	3	0
15.	Mencari data guru	3	0
16.	Menambah data guru	3	0
17.	Mengubah data guru	3	0
18.	Menghapus data guru	3	0
19.	<i>Upload</i> data guru	3	0
20.	Mencetak/ <i>export</i> data guru	3	0
21.	Menambah data petugas	3	0
22.	Mengubah data petugas	3	0
23.	Menghapus data petugas	3	0
24.	Menambah data admin	3	0
25.	Mengubah data admin	3	0
26.	Menghapus data admin	3	0
27.	Menambah data akun	3	0
28.	Mengubah data akun	3	0

No.	Fungsi	Hasil	
		Sukses	Gagal
29.	Menghapus data akun	3	0
30.	Melihat data kelas	3	0
31.	Mencari data kelas	3	0
32.	Menambah data kelas	3	0
33.	Mengubah data kelas	3	0
34.	Menghapus data kelas	3	0
35.	<i>Upload</i> data kelas	3	0
36.	Mencetak/ <i>export</i> data kelas	3	0
37.	Melihat data jadwal piket	3	0
38.	Mencari data jadwal piket	3	0
39.	Menambah data jadwal piket	3	0
40.	Mengubah data jadwal piket	3	0
41.	Menghapus data jadwal piket	3	0
42.	Mencetak/ <i>export</i> data jadwal piket	3	0
43.	Melihat data jadwal sekolah	3	0
44.	Mencari data jadwal sekolah	3	0
45.	Menambah data jadwal sekolah	3	0
46.	Mengubah data jadwal sekolah	3	0
47.	Menghapus data jadwal sekolah	3	0
48.	Upload data jadwal sekolah	3	0
49.	Melihat data rekap presensi per kelas	3	0
50.	Mencari data rekap presensi per kelas	3	0
51.	Mencetak data rekap presensi per kelas	3	0
52.	Melihat data rekap presensi per siswa	3	0
53.	Mencari data rekap presensi per siswa	3	0
54.	Mencetak/ <i>export</i> data rekap presensi per siswa	3	0
55.	Melihat data rekap perizinan per kelas	3	0
56.	Mencari data rekap perizinan per kelas	3	0
57.	Mencetak data rekap perizinan per kelas	3	0
58.	Melihat data rekap perizinan per siswa	3	0
59.	Mencari data rekap perizinan per siswa	3	0
60.	Mencetak/ <i>export</i> data rekap perizinan per siswa	3	0
61.	Melihat data rekap keterlambatan per kelas	3	0
62.	Mencari data rekap keterlambatan per kelas	3	0
63.	Mencetak data rekap keterlambatan per kelas	3	0
64.	Melihat data rekap keterlambatan per siswa	3	0
65.	Mencari data rekap keterlambatan per siswa	3	0
66.	Mencetak/ <i>export</i> data rekap keterlambatan per siswa	3	0

No.	Fungsi	Hasil	
		Sukses	Gagal
B	Petugas		
67.	Melihat data presensi per hari	3	0
68.	Mencari data presensi per hari	3	0
69.	Mengubah data presensi per hari	3	0
70.	Melihat data perizinan per hari	3	0
71.	Mencari data perizinan per hari	3	0
72.	Menambah data perizinan per hari	3	0
73.	Mengubah data perizinan per hari	3	0
74.	Menghapus data perizinan per hari	3	0
75.	Melihat data keterlambatan per hari	3	0
76.	Mencari data keterlambatan per hari	3	0
77.	Mengubah data keterlambatan per hari	3	0
78.	Memverifikasi data pesensi per hari	3	0
79.	Mengirim pesan	3	0
80.	Melihat data pesan masuk	3	0
81.	Menghapus data pesan masuk	3	0
82.	Melihat data pesan keluar	3	0
83.	Menghapus data pesan keluar	3	0
C	Guru		
84.	Melihat data presensi per hari	3	0
85.	Mencari data presensi per hari	3	0
86.	Melihat data perizinan per hari	3	0
87.	Mencari data perizinan per hari	3	0
88.	Melihat data keterlambatan per hari	3	0
89.	Mencari data keterlambatan per hari	3	0
90.	Mengirim pesan	3	0
91.	Melihat data pesan masuk	3	0
92.	Menghapus data pesan masuk	3	0
93.	Melihat data pesan keluar	3	0
94.	Menghapus data pesan keluar	3	0
95.	Melihat data siswa	3	0
96.	Melihat data guru	3	0
97.	Melihat data jadwal piket	3	0
D	Siswa		
98.	Melakukan <i>enroll</i> sidik jari ke dalam sistem	3	0
99.	Melakukan presensi menggunakan <i>fingerprint</i>	3	0
Jumlah		297	0

Untuk mendapatkan hasil akhir pengujian instrumen, kemudian dilakukan penghitungan menggunakan rumus (1) yang telah dijelaskan pada teknik analisis data. Berikut hasil penghitungannya:

$$\begin{aligned} X &= 1 - \frac{A}{B} \\ X &= 1 - \left(\frac{0}{297} \right) \\ &= 1 \end{aligned}$$

Menurut aturan yang telah dibuat ISO/IEC TR 9126-2: 2002, hasil penghitungan pengujian aspek *functionality* jika nilainya (X) mendekati **1**, maka dapat dikatakan bahwa sistem tersebut **Baik** dalam aspek *functionality*.

2. Aspek *Reliability*

Pengujian aspek *reliability* dilakukan menggunakan *software* WAPT. Dalam *software* tersebut dilakukan *stress testing* terhadap *website* sistem informasi yang akan diuji aspek reliabilitasnya. *Stress testing* dilakukan dengan memberikan sejumlah *virtual user* yang mengakses *website* tersebut dalam suatu waktu tertentu. Kemudian akan dihasilkan jumlah *successful sessions*, *failed sessions*, *successful pages*, *failed pages*, *successful hits*, dan *failed hits*. Berikut adalah rincian hasil dari pengujian terhadap *website* sistem informasi presensi menggunakan WAPT.

Pass/Fail Criteria

Name	Result	Comment
Session error rate for each profile	SUCCESS	

Summary

Profile	Successful sessions	Failed sessions	Successful pages	Failed pages	Successful hits	Failed hits	Other errors	Total KBytes sent	Total KBytes received	Avg response time, sec (with page resources)
Profile1	76	0	688	0	2268	0	0	805	132804	1.09(1.10)

Number of active users

Profile	0:00:00 - 0:01:00	0:01:00 - 0:02:00	0:02:00 - 0:03:00	0:03:00 - 0:04:00	0:04:00 - 0:05:00	0:05:00 - 0:06:00	0:06:00 - 0:07:00	0:07:00 - 0:08:00	0:08:00 - 0:09:00	0:09:00 - 0:10:00
Profile1	5	5	5	5	5	5	5	5	5	5
Total	5	5	5	5	5	5	5	5	5	5

Successful sessions (Failed sessions)

Profile	0:00:00 - 0:01:00	0:01:00 - 0:02:00	0:02:00 - 0:03:00	0:03:00 - 0:04:00	0:04:00 - 0:05:00	0:05:00 - 0:06:00	0:06:00 - 0:07:00	0:07:00 - 0:08:00	0:08:00 - 0:09:00	0:09:00 - 0:10:00	Total
Profile1	2(0)	8(0)	8(0)	7(0)	8(0)	9(0)	8(0)	9(0)	7(0)	10(0)	76(0)
Total	2(0)	8(0)	8(0)	7(0)	8(0)	9(0)	8(0)	9(0)	7(0)	10(0)	76(0)

Successful pages (Failed pages)

Profile	0:00:00 - 0:01:00	0:01:00 - 0:02:00	0:02:00 - 0:03:00	0:03:00 - 0:04:00	0:04:00 - 0:05:00	0:05:00 - 0:06:00	0:06:00 - 0:07:00	0:07:00 - 0:08:00	0:08:00 - 0:09:00	0:09:00 - 0:10:00	Total
Profile1	34(0)	74(0)	67(0)	73(0)	69(0)	73(0)	79(0)	72(0)	73(0)	74(0)	688(0)
Total	34(0)	74(0)	67(0)	73(0)	69(0)	73(0)	79(0)	72(0)	73(0)	74(0)	688(0)

Successful hits (Failed hits)

Profile	0:00:00 - 0:01:00	0:01:00 - 0:02:00	0:02:00 - 0:03:00	0:03:00 - 0:04:00	0:04:00 - 0:05:00	0:05:00 - 0:06:00	0:06:00 - 0:07:00	0:07:00 - 0:08:00	0:08:00 - 0:09:00	0:09:00 - 0:10:00	Total
Profile1	174(0)	234(0)	227(0)	213(0)	229(0)	233(0)	239(0)	272(0)	213(0)	234(0)	2268(0)
Total	174(0)	234(0)	227(0)	213(0)	229(0)	233(0)	239(0)	272(0)	213(0)	234(0)	2268(0)

Gambar 18 Rincian Hasil Pengujian *Stress Testing* menggunakan WAPT Pro 5.0

Berdasarkan rincian tersebut, hasil dari *stress testing* menggunakan WAPT dirangkum dalam tabel 8 sebagai berikut:

Tabel 3. Rangkuman Hasil Stress Testing WAPT

Aspek yang dinilai	<i>Successfull</i>	<i>Failed</i>
<i>Session</i>	76	0
<i>Pages</i>	688	0
<i>Hits</i>	2268	0
Total	3032	0

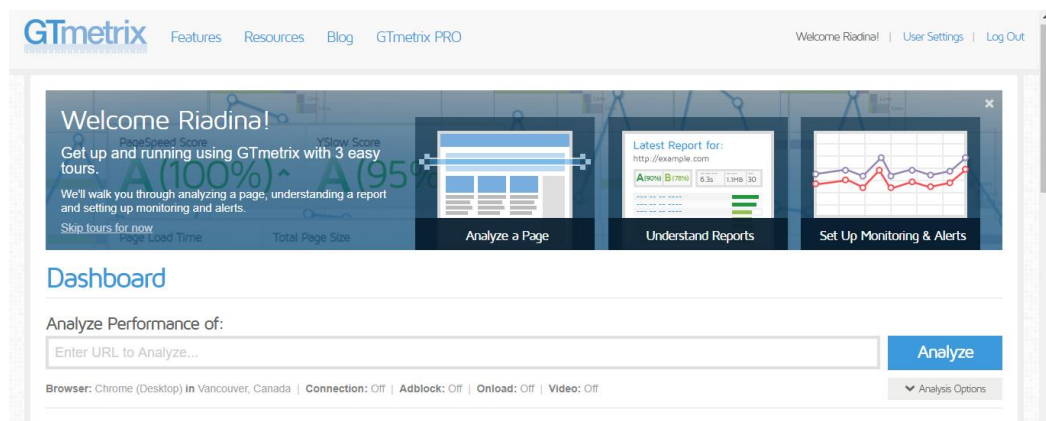
Setelah didapatkan hasil seperti pada tabel di atas, kemudian dihitung menggunakan rumus nomor (2) seperti yang telah dijelaskan pada teknik analisis data. Rumus perhitungan model Nelson menurut Maurya dan Malviya (2014: 6) sebagai berikut:

$$R = 1 - \frac{f}{n} = 1 - \frac{0}{3032} = 1$$

Berdasarkan hasil perhitungan yang didapatkan menggunakan rumus nelson, jika disesuaikan dengan Telecordia Standar R3-34 dalam GR 282 “*Software Reliability and Quality Acceptance Criteria*” (Asthana & Olivieri, 2009:2), yang menetapkan bahwa hasil pengujian dapat dikatakan *reliable* jika mempunyai presentase minimal 95% atau 0.95, maka *website* sistem informasi presensi SMK Negeri 2 Sewon telah memenuhi aspek *reliability* karena hasil perhitungannya menunjukkan nilai 1 atau 100%.

3. Aspek *Efficiency*

Pengujian aspek *efficiency* dilakukan untuk menguji kinerja *website* ketika sedang diakses oleh *user*. Penilaian kinerja *website* didasarkan pada kriteria-kriteria yang dikembangkan oleh YSlow. Pengujian ini dilakukan dengan menggunakan *website performance analytics* GTmetrix.com. *Website* ini menggunakan dua dasar kriteria penilaian yang dikembangkan oleh YSlow dan PageSpeed. *Website* ini akan menilai aspek atau komponen apa saja yang dapat menyebabkan halaman *website* berjalan lamban kemudian memberikan rekomendasi dan saran bagaimana mengoptimasi halaman *website* tersebut.



Gambar 19 Website GTMetrix

Gambar 87 sampai 112 (terlampir) merupakan hasil pengujian menggunakan website GTmetrix. Tabel 9 berikut adalah rangkuman hasil penilaian YSlow menggunakan GTmetrix di setiap halaman *website* sistem informasi presensi SMK Negeri 2 Sewon.

Tabel 4. Rangkuman Hasil Pengujian *Efficiency*

No.	Halaman	PageSpeed Score	YSlow Score	Skor yang diharapkan
1.	<i>Landing page</i>	A (94%)	A (92%)	100
2.	<i>Login</i>	A (98%)	A (94%)	100
3.	<i>Dashboard</i>	A (97%)	A (91%)	100
4.	Profil	A (99%)	A (94%)	100
5.	Pengaturan	A (96%)	A (92%)	100
6.	Pesan Baru	A (98%)	A (92%)	100
7.	Pesan Masuk	A (97%)	A (92%)	100
8.	Pesan Keluar	A (97%)	A (92%)	100
9.	Data Presensi Per Hari	A (99%)	A (92%)	100
10.	Data Perizinan Per Hari	A (99%)	A (92%)	100
11.	Data Keterlambatan Per Hari	A (99%)	A (93%)	100
12.	Verifikasi Data Presensi	A (99%)	A (93%)	100
13.	Rekap Presensi per Kelas	A (96%)	A (90%)	100
14.	Rekap Presensi per Siswa	A (97%)	A (92%)	100
15.	Rekap Perizinan per Kelas	A (97%)	A (92%)	100
16.	Rekap Perizinan per Siswa	A (97%)	A (92%)	100
17.	Rekap Keterlambatan per Hari	A (97%)	A (93%)	100
18.	Rekap Keterlambatan per Siswa	A (97%)	A (93%)	100
19.	Master Data Siswa	A (99%)	A (91%)	100
20.	Master Data Guru	A (97%)	A (90%)	100
21.	Master Data Kelas	A (97%)	A (90%)	100
22.	Master Data Petugas	A (97%)	A (90%)	100
23.	Master Data Admin	A (97%)	A (90%)	100
24.	Master Data Akun	A (97%)	A (90%)	100
25.	Master Data Jadwal Sekolah	A (99%)	A (92%)	100
26.	Master Data Jadwal Piket	A (97%)	A (90%)	100
Total		2533	2384	2600

Berdasarkan hasil rekapitulasi di atas selanjutnya dilakukan perhitungan untuk mendapatkan persentase skor pada aspek *efficiency* sebagai berikut menggunakan rumus (3).

$$\text{Presentase Kelayakan (\%)} = \frac{\text{Skor yang diobservasi}}{\text{Skor yang diharapkan}} \times 100\%$$

$$\text{Presentase Kelayakan (\%)} = \frac{2384}{2600} \times 100\% = \mathbf{91,69\%}$$

Setelah dilakukan perhitungan skor pengujian *efficiency*, dihasilkan presentase kelayakan sebesar **91,69%**. Presentase tersebut dibandingkan dengan tabel interpretasi skor YSlow pada tabel 4 mendapatkan nilai A atau dinyatakan memenuhi aspek *efficiency*.

4. Aspek Usability

Pengujian aspek *Usability* dilakukan di SMK Negeri 2 Sewon melibatkan 20 guru dan pegawai serta 32 siswa sebagai sampel. Pengujian dilakukan dengan memberikan kuisisioner kepada 25 responden (guru, pegawai, dan siswa) setelah mereka melakukan percobaan menggunakan *website* sistem informasi presensi. Kuisisioner yang digunakan adalah model USE *Questionnaire* yang dikembangkan oleh A.M. Lund. Tabel 10 berikut adalah hasil pengisian kuisisioner beserta analisis data aspek *usability*.

Tabel 5. Hasil Jawaban Pengisian Kuisisioner Aspek *Usability*

Pernyataan	SS	S	RR	KS	TS
1	5	11	2	2	0
2	4	11	3	2	0
3	6	14	0	0	0
4	4	12	4	0	0
5	4	9	7	0	0

Pernyataan	SS	S	RR	KS	TS
6	4	13	3	0	0
7	4	13	3	0	0
8	2	13	4	1	0
9	4	15	0	1	0
10	3	17	0	0	0
11	2	15	2	1	0
12	3	14	2	1	0
13	3	15	1	1	0
14	1	16	2	1	0
15	0	11	6	3	0
16	0	20	0	0	0
17	0	18	2	0	0
18	0	18	2	0	0
19	0	18	2	0	0
20	1	14	4	1	0
21	1	11	7	1	0
22	2	14	4	0	0
23	0	15	5	0	0
24	2	18	0	0	0
25	2	18	0	0	0
26	1	19	0	0	0
27	1	17	2	0	0
28	2	17	1	0	0
29	1	18	1	0	0
30	1	18	1	0	0
Total	63	452	70	15	0

Berdasarkan tabel di atas, dapat disimpulkan jawaban dari pengisian angket sebagai berikut:

Kriteria	Jumlah Jawaban
SS	63
S	452
RR	70
KS	15
TS	0
Total	600

Jumlah jawaban dari pengisian angket kemudian dihitung untuk mendapatkan presentase pengujian *usability*. Berikut adalah perhitungan presentasi aspek *usability* menggunakan rumus (4):

$$\begin{aligned} Skor_{total} &= (63 \times 5) + (452 \times 4) + (70 \times 3) + (15 \times 2) + (0 \times 1) \\ &= 315 + 1808 + 210 + 30 + 0 = \mathbf{2363} \end{aligned}$$

$$P_{skor} = \frac{2363}{30 \times 20 \times 5} \times 100\% = \mathbf{78,76\%}$$

Dari hasil perhitungan, didapatkan presentase **78,76%**. Presentase tersebut kemudian disesuaikan dengan tabel interpretasi skor pada Tabel 5. Dari tabel tersebut, hasil interpretasi skor dari presentase tersebut adalah **Layak** dan memenuhi aspek *usability*.