

LAPORAN TUGAS BESAR

Sistem Informasi Sekolah

Diajukan untuk memenuhi tugas Mata Kuliah Pemrograman Berorientasi Objek



Disusun oleh : Kelompok 2

Arpriansah Yonathan	(NIM 1301194112)
Igga Febrian Virgiani	(NIM 1301194283)
Manuel Benedict	(NIM 1301194182)

**PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM
2021**

DAFTAR ISI

DAFTAR GAMBAR	3
DAFTAR TABEL	3
BAB I	4
PENDAHULUAN	4
Latar Belakang	4
Rumusan Masalah	5
Batasan Masalah	5
Tujuan	5
BAB II	6
PEMBAHASAN	6
Desain Arsitektur	6
Class Diagram	7
Detail pada Kelas-kelas yang Dibangun	8
<i>Graphical User Interface (GUI)</i>	9
BAB III	13
PENUTUP	13
Simpulan	13
Saran	13

DAFTAR GAMBAR

Gambar 1 Model MVC	6
Gambar 2 <i>Class Diagram</i> pada sistem informasi sekolah	7
Gambar 3 Tampilan halaman <i>log in</i>	9
Gambar 4 Tampilan halaman guru	10
Gambar 5 Tampilan halaman siswa	11
Gambar 6 Tampilan halaman administrator	12

DAFTAR TABEL

Tabel 1 Kelas-kelas yang dibangun pada sistem informasi sekolah	8
---	---

BAB I

PENDAHULUAN

A. Latar Belakang

Pendidikan merupakan salah satu cara manusia untuk mendapatkan ilmu, adapun langkah untuk menempuh pendidikan tersebut dengan cara bersekolah. Sekolah merupakan tempat belajar mengajar dimana guru dan siswa saling berinteraksi dalam kegiatan pembelajaran. Siswa memiliki kewajiban untuk mengerjakan tugas dan melaksanakan ujian untuk setiap mata pelajaran yang ada, sedangkan guru memiliki tanggung jawab untuk memberikan tugas dan ujian, memeriksa tugas yang ada, dan juga melakukan rekapitulasi nilai untuk setiap siswa yang nantinya akan dilaporkan kepada sekolah.

Banyak sekolah di Indonesia yang masih memakai sistem manual dalam manajemennya. Ada banyak keterbatasan manajemen pada sistem manual, salah satunya mengenai penilaian siswa. Dalam proses penilaian tersebut rekapitulasi nilai menggunakan sistem manual dirasa belum efisien, serta dengan adanya keterbatasan seperti halnya membawa berkas fisik dengan jumlah banyak, serta adanya kemungkinan terjadi peristiwa tidak diinginkan seperti berkas yang tertinggal, ataupun rusak dapat menjadi hambatan bagi guru dalam mengolah nilai siswa, dan seperti yang kita tahu bahwa seorang guru biasanya dapat mengajar lebih dari satu kelas dan di setiap kelas sendiri terdiri dari banyak siswa yang kurang lebih berjumlah 40 orang. Hal ini membuat guru sering merasa kewalahan dengan hal tersebut ditambah tuntutan dari sekolah untuk segera memberikan laporan dalam rentang waktu yang ditentukan.

Dengan demikian diperlukannya inovasi untuk membantu dalam penyelesaian masalah tersebut, salah satunya dengan cara membuat sebuah sistem informasi. Sistem informasi sekolah merupakan sistem manajemen terpadu yang dibuat untuk mempermudah kegiatan operasional sekolah, seperti manajemen siswa dan guru, kelas, dsb. Adapun proses yang dilakukan pada sistem informasi sekolah antara lain, *input* nilai, menambahkan atau menghapus kelas, melihat informasi nilai milik siswa dengan cara melakukan pencarian, dsb.

B. Rumusan Masalah

Adapun masalah-masalah yang kami identifikasikan berkaitan dengan sistem informasi di sekolah sebagai berikut.

1. Data siswa dan guru di sekolah masih disimpan dalam dokumen fisik sehingga memerlukan waktu lama apabila data-data tersebut ingin diakses.
2. Transparansi nilai dinilai masih kurang transparan karena pada nilai rapor hanya tertera hasil akhir tanpa mengetahui detail-detail rekapitulasi nilai siswa.

C. Batasan Masalah

Sistem informasi yang dibangun pada tugas ini masih sederhana. Oleh karena ini, sistem informasi ini hanya terbatas untuk beberapa hal sebagai berikut.

1. Guru dapat melihat nama dan NIS siswa dari semua kelas dan melakukan *input* nilai untuk kegiatan dan mata pelajaran tertentu.
2. Siswa dapat melihat rekapitulasi nilai dengan cara memasukkan NIS serta melihat nama seluruh siswa berdasarkan kelas.
3. Administrator dapat mendaftarkan siswa ke satu kelas, mengedit nama kelas, serta menghapus kelas.

D. Tujuan

Dengan mengimplementasikan Pemrograman Berorientasi Objek, kami membuat sistem manajemen sekolah yang dapat menyelesaikan permasalahan yang dapat membantu kegiatan di sekolah seperti permasalahan yang telah dijabarkan pada poin sebelumnya.

Setelah diterapkannya Pemrograman Berorientasi Objek pada sistem informasi sekolah, tentunya ini dapat memudahkan sistem manajemen di sekolah. Berdasarkan yang telah disusun dari rumusan masalah di atas diharapkan bahwa:

1. Guru dapat lebih mudah merekap nilai siswa.
2. Siswa dapat melihat rekapitulasi nilai secara terperinci.
3. Administrator dapat lebih mudah melakukan pendaftaran siswa ke kelas serta mengatur kelas yang ada.

BAB II

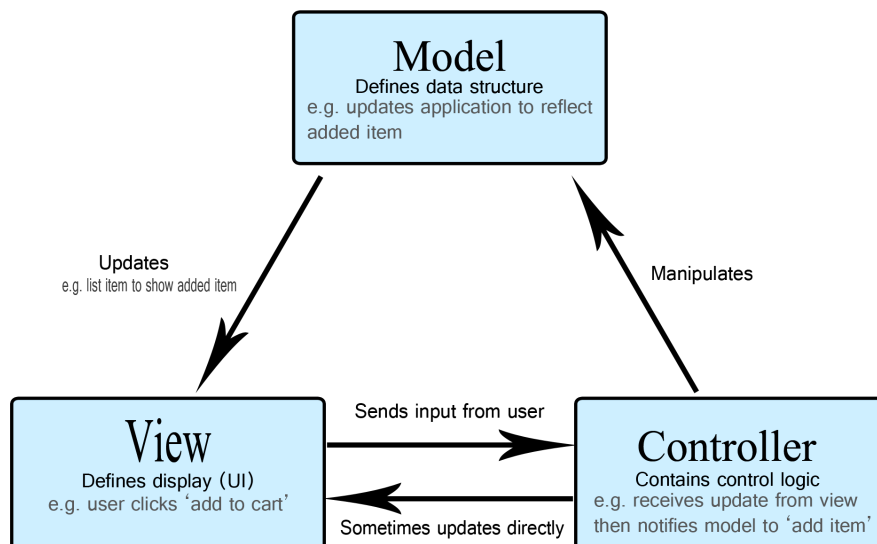
PEMBAHASAN

A. Desain Arsitektur

Tugas ini menerapkan konsep MVC. MVC (*Model, View, Controller*) adalah sebuah pola arsitektur dalam membuat sebuah aplikasi dengan cara memisahkan *code* menjadi tiga *package*, yaitu:

- **Model**: menyiapkan, mengatur, memanipulasi, dan mengorganisasikan data yang ada di *database*,
- **View**: menampilkan informasi dalam bentuk *Graphical User Interface* (GUI), dan
- **Controller**: menghubungkan serta mengatur *model* dan *view* agar dapat saling terhubung.

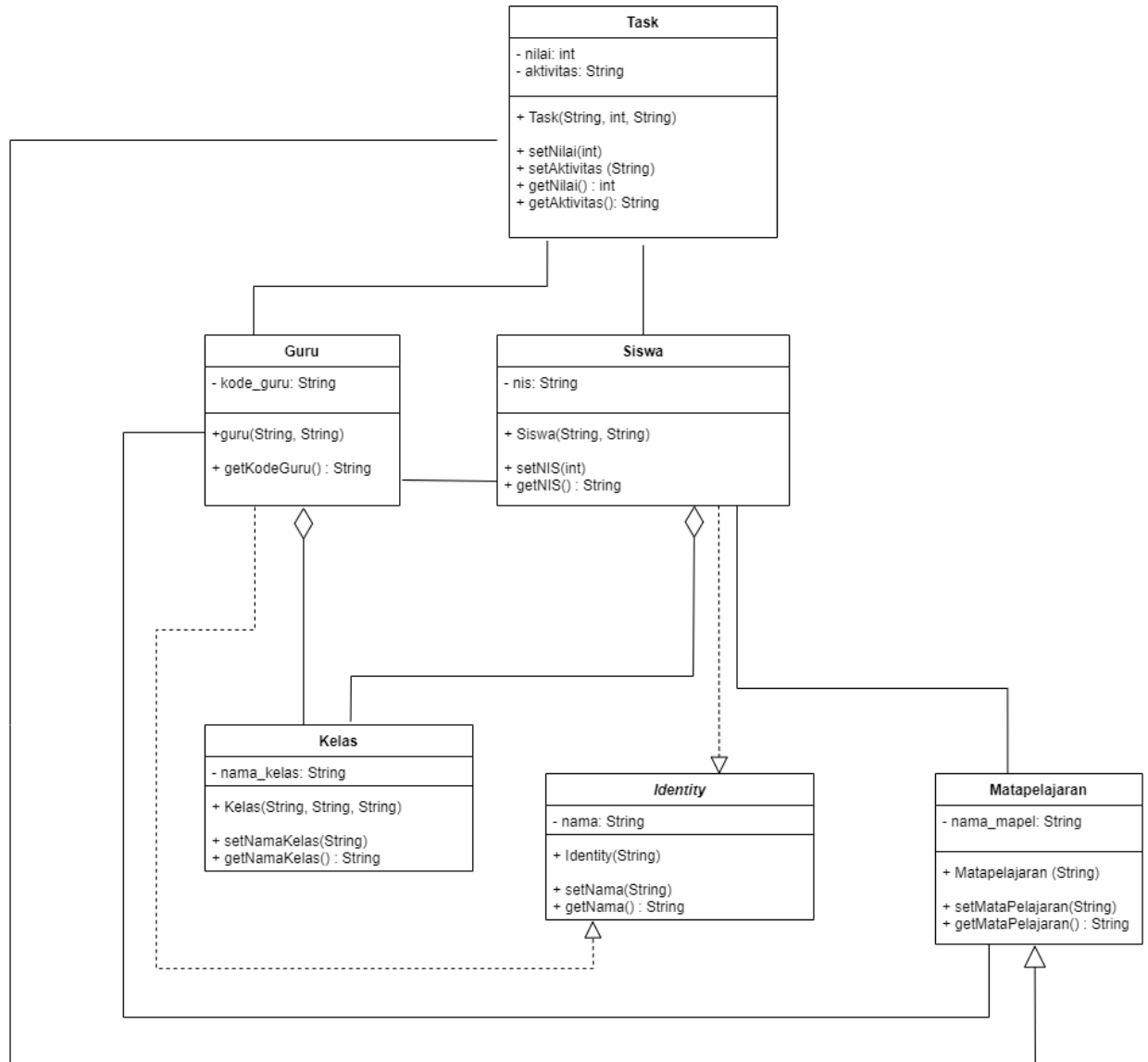
Konsep MVC sendiri direpresentasikan dalam diagram berikut.



Gambar 1 Model MVC

B. Class Diagram

Pada sebuah studi kasus mengenai sistem informasi sekolah dibuat sebuah *class diagram* yang berfungsi untuk menggambarkan sebuah sistem yang akan dibuat. Pada *class diagram* dibuat beberapa *class* dan *method* di dalamnya. *Class* yang ada berdasarkan *class diagram* yang telah dibuat yaitu *class* Task, Guru, Siswa, Kelas, Mata Pelajaran, dan Identity.



Gambar 2 Class diagram sistem informasi sekolah

C. Detail pada Kelas-kelas yang Dibangun

Pada sistem informasi, kita mengenal beberapa kelas. Kelas merupakan rancangan dari suatu objek. Setiap kelas dapat menghasilkan banyak objek dan terdiri dari dua anggota, yakni atribut dan *method*. Atribut dapat berupa tipe data dan *methods* dapat diartikan semua fungsi, yang melakukan suatu proses dan mengembalikan suatu nilai, maupun prosedur, yang hanya melakukan suatu proses, merupakan *behavior* dari suatu kelas.

Pada sistem informasi sekolah kita dapat menjadikan elemen-elemen di sekolah sebagai kelas, atribut, maupun *methods*. Berikut ini merupakan rincian dari setiap kelas yang dibangun, terdapat penjelasan mengenai variabel dan *method* yang ada.

Kelas	Variabel	Methods
Siswa	(-) nis : String	(+) Siswa(String, String) <i>Constructor</i> (+) setNIS(int) (+) getNIS() : int
Guru	(-) kode_guru : String	(+) Guru (String, String) <i>Constructor</i> (+) getKodeGuru() : String
Identity < <i>Abstract class</i> >	(-) nama: String	(+) Identity (String) <i>Constructor</i> (+) getNama() : String (+) setNama(String)
Task	(-) nilai: int (-) aktivitas: String	(+) Task(String, int, String) <i>Constructor</i> (+) setNilai(int) (+) setAktivitas (String) (+) getNilai() : float (+) getAktivitas(): String
Kelas	(-) nama_kelas: String	(+) kelas(String, String, String) <i>Constructor</i>

		(+) setNamaKelas(String) (+) getNamaKelas() : String
Matapelajaran	(-) nama_mapel: String	(+) matapelajaran (String) <i>Constructor</i> (+) setMataPelajaran(String) (+) getMataPelajaran() : String

Tabel 1 Kelas-kelas yang dibangun pada sistem informasi sekolah

D. *Graphical User Interface (GUI)*

Adapun tampilan dari aplikasi sederhana yang telah dibangun terdiri dari 4 halaman, yakni sebagai berikut.

- **Tampilan halaman *login***

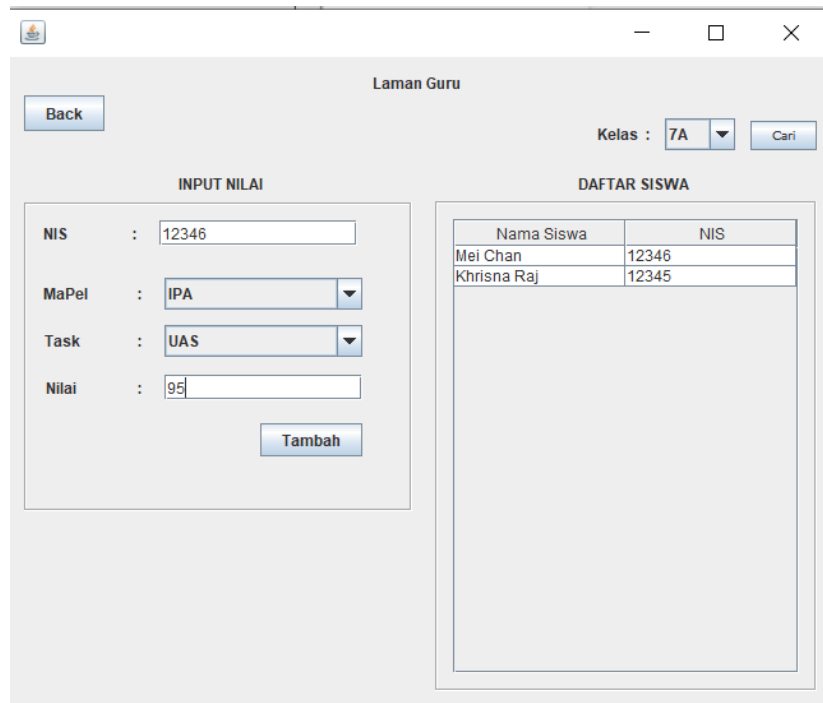
Pada halaman ini, pengguna akan masuk sesuai dengan *role*-nya di sekolah, entah sebagai guru, siswa, maupun administrator sistem informasi ini.

Gambar 3 Tampilan halaman *log in*

Pada halaman ini, pengguna akan masuk dengan memilih jenis *role* pengguna di sekolah tersebut, apakah sebagai administrator, guru, atau siswa. Kemudian, pengguna akan memasukkan *username* dan *password* nya. Pengguna dapat masuk ke salah satu laman (administrator, guru, atau siswa) apabila *role* pengguna, *username*, dan *password* sudah sesuai dengan yang ada di *database*.

- **Tampilan halaman Guru**

Pada halaman ini, guru dapat memasukkan nilai untuk setiap komponen dan setiap siswa. Guru juga dapat melihat *list* nama siswa beserta NIS di setiap kelas. Ketika guru memilih kelas dari *combo box*, dan klik tombol “Cari”, tabel akan menampilkan daftar siswa beserta NIS. *List* kelas pada *combo box* akan *ter-update* sesuai dengan kelas yang ada di *database*. Setelah guru mengetahui NIS dari siswa yang nilainya akan dimasukkan, guru akan mengisi NIS dan nilai pada *text field*, serta memilih mata pelajaran dan aktivitas pada *combo box*. Ketika guru mengklik tombol “Tambah”, semua komponen akan tersimpan di *database*. Komponen akan tersimpan apabila *text fields* terisi dan *combo box* pada “MaPel” dan “Task” sudah dipilih.

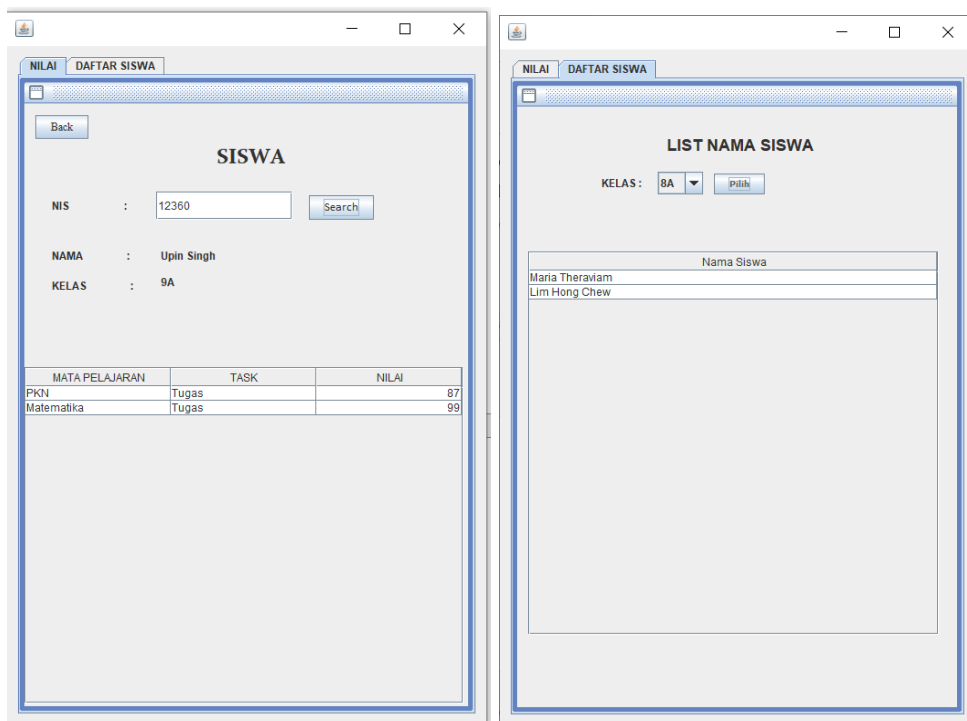


Nama Siswa	NIS
Mei Chan	12346
Khrisna Raj	12345

Gambar 4 Tampilan halaman guru

- **Tampilan halaman Siswa**

Pada halaman siswa, terdapat dua *tab*. *Tab* yang pertama, *tab* Nilai, bertujuan untuk melihat seluruh hasil rekapitulasi nilai dari berbagai komponen dan mata pelajaran. Siswa akan memasukkan NIS dan klik “Cari”. Setelah itu, akan ada informasi nama dan kelas siswa dari NIS tersebut beserta hasil rekapitulasi nilai pada tabel. Hal tersebut akan berhasil apabila NIS yang dicari terdaftar pada *database*.

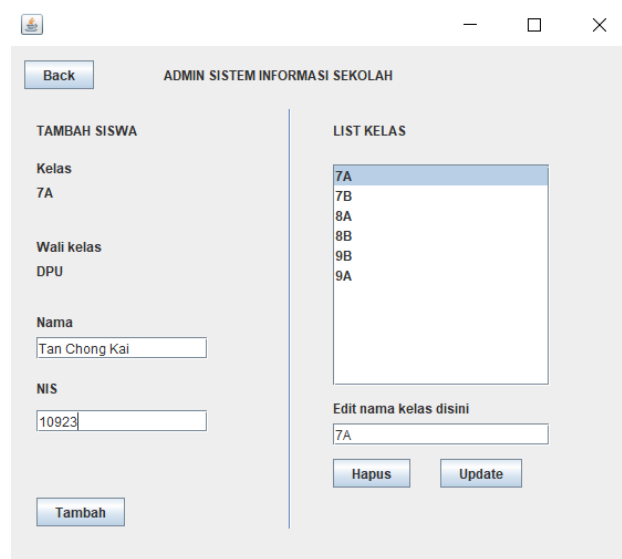


Gambar 5 Tampilan halaman siswa

Tab kedua, *tab* Daftar Siswa, bertujuan untuk menampilkan daftar nama siswa di semua kelas. Pertama, pengguna akan memilih kelas dari *combo box*, lalu klik “Pilih”. Setelah itu, akan muncul daftar siswa dari kelas yang dipilih. Kelas yang ada di *combo box* hanya dari kelas yang ada di *database*.

- **Tampilan halaman Administrator**

Administrator pada kasus ini bertugas untuk mendaftarkan siswa ke satu kelas, menghapus kelas, serta mengedit nama kelas.



Gambar 6 Tampilan halaman administrator

Saat admin mengakses aplikasi ini, akan ada *list* dari kelas yang ada di sekolah. Pertama kali, admin akan memilih kelas untuk diedit, dihapus, atau mendaftarkan siswa ke kelas tersebut. Setelah memilih kelas, akan ada informasi nama kelas dan wali kelas berupa kode dari kelas yang dipilih. Untuk menambahkan siswa, admin akan memasukkan nama dan NIS siswa pada *text field* yang ada, lalu klik “Tambah”. Penambahan akan berhasil apabila admin sudah memilih kelas serta mengisi *text field* nama dan NIS. Untuk menghapus kelas, admin hanya mengklik tombol “Hapus”. Penghapusan berhasil apabila admin sudah memilih kelas dari *list*. Untuk *update* juga ketika kita memilih kelas yang ingin di-*update*, akan muncul secara otomatis kelas pada *text field*. Untuk mengedit, admin mengubah nama kelas yang telah muncul di *text field* sebelumnya. Setelah edit, lalu klik “Update”. *Update* akan berhasil apabila admin memilih kelas dan mengedit namanya di *text field*.

BAB III PENUTUP

A. Simpulan

Adapun yang dapat diaplikasikan dari program yang telah dibangun adalah sebagai berikut.

- Administrator mendaftarkan siswa ke kelas yang ada.
- Guru dapat dengan mudah merekapitulasi nilai-nilai peserta didik.
- Siswa dapat mengakses semua detil dari hasil rekapitulasi untuk semua komponen dan mata pelajaran.
- Siswa dapat mengenali teman-teman sekelasnya.

B. Saran

Program yang telah dibangun dapat dikembangkan untuk hasil yang lebih baik dan kebutuhan yang lebih kompleks. Adapun saran untuk pengembangan kedepannya adalah sebagai berikut.

- Siswa dapat masuk ke halaman siswa dimana ia hanya dapat mengakses hasil rekapitulasi nilai sendiri sehingga privasi nilai orang lain dapat terjaga.
- Guru dapat masuk ke halaman guru dimana ia hanya dapat menambahkan nilai pada mata pelajaran yang diajar.
- Pada halaman login, fitur sembunyikan sandi juga dapat diterapkan agar data warga sekolah aman.