

## **BAB 5**

### **IMPLEMENTASI SISTEM**

Bab ini membahas tentang implementasi sistem dari rancangan solusi yang sudah direncanakan dari bab-bab sebelumnya. Implementasi solusi ini dibangun meliputi database dan bagaimana program ditulis.

#### **5.1 IMPLEMENTASI BASIS DATA**

Rancangan *Entity Relationship Diagram* (ERD) yang telah dibuat diimplementasikan dengan database manager MySQL. Pada bagian ini akan dijelaskan setiap bagian dari tabel yang terbentuk dalam database. Selain itu, pada bagian ini juga akan dijelaskan kegunaan dari setiap tabel yang telah dibentuk.

##### **1. Tabel Absents**

Tabel absents berguna untuk menyimpan catatan absen siswa. Data yang disimpan yaitu tanggal awal absen, tanggal berakhir absen, tipe absen, deskripsi/keterangan absen, bukti keterangan tidak hadir, id siswa, id staff/karyawan yang mencatat, dan akan tersimpan secara otomatis oleh bawaan sistem yaitu *created\_at* dan *updated\_at*. Struktur tabel absents bisa dilihat seperti pada gambar 5.1.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	<b>id</b> 🔑	int(10)		UNSIGNED	No	None	
2	<b>START_DATE</b>	date			No	None	
3	<b>END_DATE</b>	date			No	None	
4	<b>TYPE</b>	enum('ABSENT', 'LEAVE', 'SICK')	utf8mb4_unicode_ci		No	None	
5	<b>DESCRIPTION</b>	text	utf8mb4_unicode_ci		No	None	
6	<b>RECEIPT_IMG</b>	text	utf8mb4_unicode_ci		No	None	
7	<b>STUDENTS_ID</b> 🔑	int(10)		UNSIGNED	No	None	
8	<b>ACADEMIC_YEAR_ID</b> 🔑	int(10)		UNSIGNED	No	None	
9	<b>STAFFS_ID</b> 🔑	int(10)		UNSIGNED	No	None	
10	<b>created_at</b>	timestamp			Yes	NULL	
11	<b>updated_at</b>	timestamp			Yes	NULL	

Gambar 5.1. Struktur tabel absents

## 2. Tabel Academic\_Years

Tabel academic\_years adalah tabel master yang digunakan untuk menyimpan daftar tahun ajaran. Data yang akan disimpan dalam tabel ini adalah nama tahun ajaran, tipe tahun ajaran yaitu ganjil atau genap, tanggal awal dimulai tahun ajaran, tanggal berakhirnya tahun ajaran. Struktur tabel academic\_years dapat dilihat pada tabel 5.2.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	<b>id</b> 🔑	int(10)		UNSIGNED	No	None	
2	<b>NAME</b>	varchar(45)	utf8mb4_unicode_ci		No	None	
3	<b>TYPE</b>	enum('ODD', 'EVEN')	utf8mb4_unicode_ci		No	None	
4	<b>START_DATE</b>	date			No	None	
5	<b>END_DATE</b>	date			No	None	

Gambar 5.2. Struktur tabel academic years

### 3. Table Achievements

Tabel achievements adalah tabel master yang digunakan untuk menyimpan daftar penghargaan yang berlaku di sekolah. Data yang akan disimpan dalam tabel ini adalah deskripsi, poin, tingkat penghargaan, dan tipe penghargaan. Struktur tabel academic\_years dapat dilihat pada tabel 5.3.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	<b>id</b> 🔑	int(10)		UNSIGNED	No	None	
2	<b>DESCRIPTION</b>	text	utf8mb4_unicode_ci		No	None	
3	<b>POINT</b>	double			No	None	
4	<b>GRADE</b>	text	utf8mb4_unicode_ci		No	None	

**Gambar 5.3 Struktur tabel achievements**

### 4. Table Achievement\_Records


Tabel achievement\_records berguna untuk menyimpan catatan penghargaan yang diperoleh siswa dan detail catatannya. Data yang disimpan pada tabel ini adalah id siswa, id penghargaan, tanggal, deskripsi, id tahun ajaran, id staff/karyawan, *created\_at* dan *updated\_at*. Struktur tabel achievement\_records bisa dilihat pada gambar 5.4.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	<b>id</b> 🔑	int(10)		UNSIGNED	No	None	
2	<b>STUDENTS_ID</b> ☁	int(10)		UNSIGNED	No	None	
3	<b>ACHIEVEMENTS_ID</b> ☁	int(10)		UNSIGNED	No	None	
4	<b>DATE</b>	date			No	None	
5	<b>DESCRIPTION</b>	text	utf8mb4_unicode_ci		No	None	
6	<b>ACADEMIC_YEAR_ID</b> ☁	int(10)		UNSIGNED	No	None	
7	<b>STAFFS_ID</b> ☁	int(10)		UNSIGNED	No	None	
8	<b>created_at</b>	timestamp			Yes	NULL	
9	<b>updated_at</b>	timestamp			Yes	NULL	

**Gambar 5.4 Struktur tabel achievement records**

### 5. Tabel Activities




Tabel activities adalah tabel master yang digunakan untuk menyimpan daftar kegiatan/aktivitas pembelajaran yang berlaku di sekolah. Data yang akan disimpan dalam tabel ini adalah module dan deskripsi. Struktur tabel academic\_years dapat dilihat pada tabel 5.5.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	<b>id</b> 	int(10)		UNSIGNED	No	None	
2	<b>MODULE</b>	varchar(45)	utf8mb4_unicode_ci		No	None	
3	<b>DESCRIPTION</b>	text	utf8mb4_unicode_ci		No	None	

**Gambar 5.5 Struktur tabel activities**

### 6. Tabel Activities\_KD






Tabel activities\_kd adalah hubungan antara tabel activities dan tabel kd. Tabel ini berguna untuk menyimpan aktivitas yang dilakukan yang berkaitan dengan satu kompetensi dasar (kd) tertentu. Di dalam tabel ini akan menyimpan data yaitu, id kd, id aktivitas, *created\_at* dan *updated\_at*. Struktur tabel activities kd digambarkan pada gambar 5.6.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	<b>id</b> 	int(10)		UNSIGNED	No	None	
2	<b>KD_ID</b> 	int(10)		UNSIGNED	No	None	
3	<b>ACTIVITIES_ID</b> 	int(10)		UNSIGNED	No	None	
4	<b>created_at</b>	timestamp			Yes	NULL	
5	<b>updated_at</b>	timestamp			Yes	NULL	

**Gambar 5.6 Struktur tabel activities kd**

### 7. Tabel Activities\_Students


Tabel activities\_students digunakan untuk menyimpan detail transaksi data pada saat melakukan peng-inputan nilai suatu mata pelajaran oleh *user*. Di dalam tabel ini akan tersimpan data yaitu id siswa sebagai *foreign key*, id aktivitas sebagai *foreign key*, id mata pelajaran sebagai *foreign key*, id tahun ajaran sebagai *foreign key*, *created\_at* dan *updated\_at*. Struktur tabel activities\_students ini dapat digambarkan seperti pada gambar 5.7.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	id 	int(10)		UNSIGNED	No	None	
2	STUDENTS_ID 	int(10)		UNSIGNED	No	None	
3	ACTIVITIES_ID 	int(10)		UNSIGNED	No	None	
4	SUBJECTS_ID 	int(10)		UNSIGNED	No	None	
5	ACADEMIC_YEAR_ID 	int(10)		UNSIGNED	No	None	
6	SCORE	double			No	None	
7	created_at	timestamp			Yes	NULL	
8	updated_at	timestamp			Yes	NULL	

**Gambar 5.7 Struktur tabel activities\_students**

### 8. Tabel Banks


Tabel banks digunakan untuk menampung data master bank. Data yang disimpan dalam tabel ini yaitu nama bank, keterangan bank. Untuk membedakan setiap bank digunakan primary key. Struktur tabel bank dapat dilihat pada tabel 5.8.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	id 	int(10)		UNSIGNED	No	None	
2	NAME	varchar(45)	utf8mb4_unicode_ci		No	None	
3	DESCRIPTION	text	utf8mb4_unicode_ci		No	None	

**Gambar 5.8 Struktur tabel banks**

### 9. Tabel Departments




Tabel departments adalah tabel master yang digunakan untuk menyimpan daftar departemen yang berada di sekolah. Tabel ini akan menyimpan data yaitu nama departemen, dan deskripsi dari departemen. Struktur tabel ini bisa dilihat pada gambar 5.9.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	<b>id</b> 	int(10)		UNSIGNED	No	None	
2	<b>NAME</b>	varchar(45)	utf8mb4_unicode_ci		No	None	
3	<b>DESCRIPTION</b>	text	utf8mb4_unicode_ci		No	None	

**Gambar 5.9 Struktur tabel departments**

### 10. Tabel Departments\_Staffs



Tabel departments\_staffs adalah tabel hasil relasi *many to many* antara tabel departments dengan tabel staffs. Tabel ini digunakan sebagai penyimpan data id staff dan id departemen. Selain itu juga disimpan tanggal transaksi kapan hubungan antara satu id staff dan satu id departemen terbentuk yaitu dengan menyimpan *created\_at* dan *updated\_at*. Struktur tabel departments\_staffs digambarkan seperti pada gambar 5.10.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	<b>id</b> 	int(10)		UNSIGNED	No	None	
2	<b>STAFFS_ID</b> 	int(10)		UNSIGNED	No	None	
3	<b>DEPARTMENTS_ID</b> 	int(10)		UNSIGNED	No	None	
4	<b>created_at</b>	timestamp			Yes	NULL	
5	<b>updated_at</b>	timestamp			Yes	NULL	

**Gambar 5.10 Struktur tabel departments staffs**

### 11. Tabel Extracurriculars




Tabel extracurriculars adalah tabel master yang menyimpan daftar ekstrakurikuler yang berada di sekolah. Tabel ini menyimpan data seperti nama ekskul, deskripsi ekskul, dan id staff sebagai *foreign key* yang menandakan guru pengajar ekskul tersebut. Struktur tabel extracurriculars digambarkan seperti pada gambar 5.11.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	id 	int(10)		UNSIGNED	No	None	
2	NAME	varchar(45)	utf8mb4_unicode_ci		No	None	
3	DESCRIPTION	text	utf8mb4_unicode_ci		No	None	
4	STAFFS_ID 	int(10)		UNSIGNED	Yes	NULL	

**Gambar 5.11 Struktur tabel extracurriculars**

### 12. Tabel Extracurricular\_Records

Tabel extracurricular\_records adalah tabel yang digunakan untuk menyimpan catatan ekstrakurikuler setiap siswa pada tahun ajaran tertentu. Tabel ini akan menyimpan data yakni id tahun ajaran sebagai *foreign key*, dan juga id siswa sebagai *foreign key*, *created\_at* dan *updated\_at*. Struktur tabel extracurricular\_records digambarkan seperti pada gambar 5.12.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	id 	int(10)		UNSIGNED	No	None	
2	ACADEMIC_YEAR_ID 	int(10)		UNSIGNED	No	None	
3	STUDENTS_ID 	int(10)		UNSIGNED	No	None	
4	created_at	timestamp			Yes	NULL	
5	updated_at	timestamp			Yes	NULL	

**Gambar 5.12 Struktur tabel extracurricular records**

### 13. Tabel Extracurricular\_Reports

Tabel `extracurricular_reports` merupakan tabel hasil relasi *many to many* antara tabel `extracurricular_records` dan tabel `extracurriculars`. Tabel ini digunakan untuk menyimpan transaksi detail setiap mata ekstrakurikuler yang terdapat dalam satu id catatan ekstrakurikuler pada tabel `extracurricular_records`. Data yang akan disimpan pada tabel ini adalah id `extracurricular` sebagai *foreign key*, id dari tabel `extracurricular_records` sebagai *foreign key*, nilai, deskripsi, *created\_at* dan *updated\_at*. Tabel ini digambarkan seperti pada gambar 5.13.




#	Name	Type	Collation	Attributes	Null	Default
1	<b>id</b> 🔑	int(10)		UNSIGNED	No	None
2	<b>EXTRACURRICULARS_ID</b> 🔑	int(10)		UNSIGNED	No	None
3	<b>EXTRACURRICULAR_RECORD_ID</b> 🔑	int(10)		UNSIGNED	No	None
4	<b>SCORE</b>	double			No	None
5	<b>DESCRIPTION</b>	text	utf8mb4_unicode_ci		No	None
6	<b>created_at</b>	timestamp			Yes	NULL
7	<b>updated_at</b>	timestamp			Yes	NULL

**Gambar 5.13 Struktur tabel extracurricular reports**

### 14. Tabel Grades

Tabel `grades` digunakan untuk menyimpan daftar nama kelas. Data yang dicatat yaitu nama kelas, keterangan kelas, tingkat kelas, id unik dari tabel `program` sebagai *foreign key*, id unik dari tabel `staff` sebagai *foreign key*. Struktur tabel `grades` digambarkan seperti pada gambar 5.14.







#	Name	Type	Collation	Attributes	Null	Default	Comments
1	<b>id</b> 	int(10)		UNSIGNED	No	None	
2	<b>NAME</b>	varchar(45)	utf8mb4_unicode_ci		No	None	
3	<b>DESCRIPTION</b>	text	utf8mb4_unicode_ci		No	None	
4	<b>GRADE</b>	enum('10', '11', '12')	utf8mb4_unicode_ci		No	None	
5	<b>PROGRAMS_ID</b> 	int(10)		UNSIGNED	No	None	
6	<b>STAFFS_ID</b> 	int(10)		UNSIGNED	No	None	

Gambar 5.14 Struktur tabel grades

## 15. Tabel Grades\_Students



Tabel grades\_students adalah tabel hasil relasi antara tabel grades dan tabel students. Tabel ini untuk mencatat transaksi detail antara siswa dan kelas. Hal ini digunakan untuk memetakan bahwa satu siswa memiliki satu kelas disetiap tahun ajarannya. Artinya siswa tidak terus menerus menetap pada kelas itu, dalam kasus nyata setiap siswa pasti ada peningkatan kelas. Oleh itu data yang dicatat dalam tabel ini adalah id unik siswa yang menjadi *foreign key*, id unik kelas yang menjadi *foreign key*, dan id unik tahun ajaran sebagai *foreign key*, *created\_at* dan *updated\_at*. Struktur data tabel grades\_students dapat dilihat pada tabel 5.15.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	<b>id</b> 	int(10)		UNSIGNED	No	None	
2	<b>STUDENTS_ID</b> 	int(10)		UNSIGNED	Yes	NULL	
3	<b>GRADES_ID</b> 	int(10)		UNSIGNED	Yes	NULL	
4	<b>ACADEMIC_YEAR_ID</b> 	int(10)		UNSIGNED	Yes	NULL	
5	<b>created_at</b>	timestamp			Yes	NULL	
6	<b>updated_at</b>	timestamp			Yes	NULL	

Gambar 5.15 Struktur tabel grades students

## 16. Tabel Guardians

Tabel guardians adalah tabel yang digunakan untuk menyimpan data wali siswa. Setiap siswa memiliki perwakilan yang datanya harus disimpan. Adapun data yang disimpan pada tabel ini adalah nama depan wali siswa, nama akhir wali siswa, tanggal lahir wali siswa, status wali siswa apakah ayah/ibu/lainnya, pekerjaan wali siswa, pendidikan terakhir, nomor telepon, alamat email, alamat tempat tinggal, id siswa sebagai *foreign key*, *created\_at* dan *updated\_at*. Struktur tabel ini dapat dilihat seperti pada gambar 5.16.



#	Name	Type	Collation	Attributes	Null	Default	Comments
1	<b>id</b> 	int(10)		UNSIGNED	No	None	
2	<b>FIRST_NAME</b>	varchar(191)	utf8mb4_unicode_ci		No	None	
3	<b>LAST_NAME</b>	varchar(191)	utf8mb4_unicode_ci		No	None	
4	<b>B_YEAR</b>	varchar(191)	utf8mb4_unicode_ci		No	None	
5	<b>STATUS</b>	enum('FATHER', 'MOTHER', 'OTHER')	utf8mb4_unicode_ci		No	None	
6	<b>JOB</b>	varchar(191)	utf8mb4_unicode_ci		No	None	
7	<b>EDUCATION</b>	varchar(191)	utf8mb4_unicode_ci		No	None	
8	<b>PHONE</b>	varchar(191)	utf8mb4_unicode_ci		No	None	
9	<b>EMAIL</b>	varchar(191)	utf8mb4_unicode_ci		No	None	
10	<b>ADDRESS</b>	text	utf8mb4_unicode_ci		No	None	
11	<b>STUDENTS_ID</b> 	int(10)		UNSIGNED	Yes	NULL	
12	<b>created_at</b>	timestamp			Yes	NULL	
13	<b>updated_at</b>	timestamp			Yes	NULL	

**Gambar 5.16 Struktur tabel guardians**

## 17. Tabel KD

Tabel kd adalah tabel master yang berguna untuk menyimpan daftar kompetensi dasar. Tabel ini akan menyimpan nomor kompetensi dasar, deskripsi


kd, dan juga id mata pelajaran sebagai *foreign key*. Struktur tabel kd digambarkan seperti pada gambar 5.17.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	<b>id</b> 	int(10)		UNSIGNED	No	None	
2	<b>NUMBER</b>	varchar(45)	utf8mb4_unicode_ci		No	None	
3	<b>DESCRIPTION</b>	varchar(45)	utf8mb4_unicode_ci		No	None	
4	<b>SUBJECTS_ID</b> 	int(10)		UNSIGNED	No	None	

**Gambar 5.17 Struktur tabel kd**

#### 18. Tabel Programs


Tabel programs digunakan untuk menampung data mengenai program studi disekolah. Data yang disimpan yaitu nama program studi, keterangan program studi. Struktur tabel programs dapat dilihat pada tabel 5.18.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	<b>id</b> 	int(10)		UNSIGNED	No	None	
2	<b>NAME</b>	varchar(45)	utf8mb4_unicode_ci		No	None	
3	<b>DESCRIPTION</b>	text	utf8mb4_unicode_ci		No	None	

**Gambar 5.18 Struktur tabel programs**

#### 19. Tabel Religions


Tabel religion merupakan tabel master untuk menampung data mengenai agama. Data yang akan dicatat dalam tabel ini adalah nama agama. Struktur tabel religions dapat dilihat pada tabel 5.19.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	<b>id</b> 	int(10)		UNSIGNED	No	None	
2	<b>NAME</b>	varchar(45)	utf8mb4_unicode_ci		No	None	

**Gambar 5.19 Struktur tabel religions**

## 20. Tabel Staffs

Tabel staffs adalah tabel yang digunakan untuk menyimpan data karyawan/staff yang bekerja di sekolah. Tabel ini akan menyimpan data seperti nomor induk kepegawaian (NIK), nama lengkap staff, hak akses/*role* staff yaitu apakah terdaftar sebagai guru atau admin atau petugas uks atau guru BK atau kepala sekolah. Struktur tabel staffs digambarkan seperti pada gambar 5.20.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	<b>id</b> 	int(10)		UNSIGNED	No	None	
2	<b>NIK</b>	varchar(45)	utf8mb4_unicode_ci		No	None	
3	<b>NAME</b>	varchar(45)	utf8mb4_unicode_ci		No	None	
4	<b>ROLE</b>	enum('TEACHER', 'ADMIN', 'MEDICAL', 'ADVISOR', 'HEAD') utf8mb4_unicode_ci	utf8mb4_unicode_ci		No	None	

**Gambar 5.20 Struktur tabel staffs**

## 21. Tabel Students

Tabel students adalah tabel yang digunakan untuk menyimpan data siswa. Data siswa yang akan disimpan yaitu atribut id siswa, nomor kartu pelajar, nomor induk sekolah (NIS), nomor induk siswa nasional (NISN), nomor induk kependudukan jika ada (NIK), nama depan siswa, nama belakang siswa, jenis kelamin, tempat dan tanggal lahir, email, nomor telepon, alamat tempat tinggal, rt,



rw, kelurahan, kecamatan, kota tempat tinggal, provinsi, data lulusan dari sekolah menengah pertama mana sebelumnya, nomor rek bank, status siswa, catatan biodata, foto siswa, id bank sebagai *foreign key*, id agama sebagai *foreign key*, id tahun ajaran sebagai *foreign key* yang merupakan atribut penjelas pada tahun ajaran beberapa siswa tersebut terdaftar, id token sebagai *foreign key*, *created\_at* dan *updated\_at*. Struktur tabel siswa digambarkan seperti pada gambar 5.21.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	<b>id</b>	int(10)		UNSIGNED	No	None	
2	<b>ATT_ID</b>	varchar(45)	utf8mb4_unicode_ci		Yes	NULL	
3	<b>CARD_NUM</b>	varchar(45)	utf8mb4_unicode_ci		Yes	NULL	
4	<b>NIS</b>	varchar(45)	utf8mb4_unicode_ci		Yes	NULL	
5	<b>NISN</b>	varchar(45)	utf8mb4_unicode_ci		No	None	
6	<b>NIK</b>	varchar(45)	utf8mb4_unicode_ci		Yes	NULL	
7	<b>FNAME</b>	varchar(45)	utf8mb4_unicode_ci		No	None	
8	<b>LNAME</b>	varchar(45)	utf8mb4_unicode_ci		No	None	
9	<b>GENDER</b>	enum('MALE', 'FEMALE')	utf8mb4_unicode_ci		Yes	NULL	
10	<b>BPLACE</b>	varchar(45)	utf8mb4_unicode_ci		Yes	NULL	
11	<b>BDATE</b>	date			Yes	NULL	
12	<b>MAIL</b>	varchar(45)	utf8mb4_unicode_ci		Yes	NULL	
13	<b>PHONE</b>	varchar(20)	utf8mb4_unicode_ci		Yes	NULL	
14	<b>ADDRESS</b>	varchar(45)	utf8mb4_unicode_ci		Yes	NULL	
15	<b>RT</b>	varchar(45)	utf8mb4_unicode_ci		Yes	NULL	
16	<b>RW</b>	varchar(45)	utf8mb4_unicode_ci		Yes	NULL	
17	<b>DISTRICT</b>	varchar(45)	utf8mb4_unicode_ci		Yes	NULL	
18	<b>SUBDISTRICT</b>	varchar(45)	utf8mb4_unicode_ci		Yes	NULL	
19	<b>CITY</b>	varchar(45)	utf8mb4_unicode_ci		Yes	NULL	
20	<b>PROVINCE</b>	varchar(45)	utf8mb4_unicode_ci		Yes	NULL	
21	<b>GR_FROM</b>	varchar(45)	utf8mb4_unicode_ci		Yes	NULL	
22	<b>BANK_ACC</b>	varchar(45)	utf8mb4_unicode_ci		Yes	NULL	
23	<b>STATUS</b>	enum('STUDENT', 'ALUMNI', 'SUSPEND', 'INACTIVE')	utf8mb4_unicode_ci		Yes	NULL	
24	<b>NOTES</b>	text	utf8mb4_unicode_ci		Yes	NULL	
25	<b>IMG_PATH</b>	text	utf8mb4_unicode_ci		Yes	NULL	
26	<b>BANKS_ID</b>	int(10)		UNSIGNED	Yes	NULL	
27	<b>RELIGIONS_ID</b>	int(10)		UNSIGNED	Yes	NULL	
28	<b>ACADEMIC_YEAR_ID</b>	int(10)		UNSIGNED	Yes	NULL	
29	<b>TOKENS_ID</b>	int(10)		UNSIGNED	Yes	NULL	
30	<b>created_at</b>	timestamp			Yes	NULL	
31	<b>updated_at</b>	timestamp			Yes	NULL	

**Gambar 5.21** Struktur tabel students

## 22. Tabel Subjects




Tabel subjects adalah tabel yang digunakan untuk menyimpan daftar mata pelajaran sesuai departemen yang ada di sekolah. Data yang akan dicatat dalam tabel ini yaitu kode mata pelajaran, deskripsi mata pelajaran, nilai kkm, tipe mata pelajaran, id departemen sebagai *foreign key*. Struktur data tabel subjects dapat dilihat pada tabel 5.22.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	<b>id</b> 	int(10)		UNSIGNED	No	None	
2	<b>CODE</b>	varchar(45)	utf8mb4_unicode_ci		No	None	
3	<b>DESCRIPTION</b>	varchar(255)	utf8mb4_unicode_ci		No	None	
4	<b>MINIMALPOIN</b>	double			No	None	
5	<b>TYPE</b>	enum('MN', 'MK', 'MPK', 'DBK', 'DPK', 'KK')	utf8mb4_unicode_ci		No	None	
6	<b>DEPARTMENTS_ID</b> 	int(10)		UNSIGNED	No	None	

**Gambar 5.22 Struktur tabel subjects**

## 23. Tabel Subject\_Records

Tabel subject\_records adalah tabel yang menyimpan catatan mata pelajaran setiap satu id siswa dengan id tahun ajaran tertentu. Data yang tersimpan yaitu id tahun ajaran sebagai *foreign key*, id siswa sebagai *foreign key*, *created at* dan *updated at*. Struktur tabel ini digambarkan seperti pada gambar 5.23.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	id 	int(10)		UNSIGNED	No	None	
2	ACADEMIC_YEAR_ID 	int(10)		UNSIGNED	No	None	
3	STUDENTS_ID 	int(10)		UNSIGNED	No	None	
4	created_at	timestamp			Yes	NULL	
5	updated_at	timestamp			Yes	NULL	

**Gambar 5.23** Struktur tabel subject records

#### 24. Tabel Subject\_Reports

Tabel subject\_reports merupakan tabel hasil relasi *many to many* antara tabel subject\_records dan tabel subjects. Tabel ini digunakan untuk menyimpan transaksi detail setiap mata pelajaran yang terdapat dalam satu id catatan mata pelajaran. Data yang akan dicatat dalam tabel ini adalah id unik dari tabel subject\_record sebagai *foreign key*, id unik subjects sebagai *foreign key*, nilai akhir setiap mata pelajaran atau akumulasi nilai tiap-tiap mata pelajaran, verifikasi terhadap nilai tersebut apakah sudah diverifikasi atau belum. Selain itu tabel ini juga menyimpan data dalam format *json string* yang berguna untuk menyimpan rincian nilai dalam satu kolom berbentuk string yaitu tugas, ph, pts, pas, un, us, dan tentunya tanggal transaksi dicatat, dan tanggal transaksi diubah. Struktur tabel subjects\_reports dapat dilihat pada tabel 5.24.

#	Name	Type	Collation	Attributes	Null	Default
1	<b>id</b> 🔑	int(10)		UNSIGNED	No	None
2	<b>SUBJECTS_ID</b> 🔑	int(10)		UNSIGNED	No	None
3	<b>SUBJECT_RECORD_ID</b> 🔑	int(10)		UNSIGNED	No	None
4	<b>FINAL_SCORE</b>	double			No	None
5	<b>IS_VERIFIED</b>	tinyint(4)			No	None
6	<b>TUGAS</b>	text	utf8mb4_unicode_ci		No	None
7	<b>PH</b>	text	utf8mb4_unicode_ci		No	None
8	<b>PTS</b>	text	utf8mb4_unicode_ci		No	None
9	<b>PAS</b>	text	utf8mb4_unicode_ci		No	None
10	<b>UN</b>	text	utf8mb4_unicode_ci		No	None
11	<b>US</b>	text	utf8mb4_unicode_ci		No	None
12	<b>created_at</b>	timestamp			Yes	NULL
13	<b>updated_at</b>	timestamp			Yes	NULL

Gambar 5.24 Struktur tabel subject reports

## 25. Tabel Tokens

Tabel tokens digunakan untuk menyimpan nomor token setiap kali siswa terdaftar. Data yang disimpan adalah kode token, status token apakah aktif atau tidak, *created\_at* dan *updated\_at*. Struktur tabel tokens digambarkan seperti pada gambar 5.25.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	<b>id</b> 🔑	int(10)		UNSIGNED	No	None	
2	<b>CODE</b>	varchar(10)	utf8mb4_unicode_ci		No	None	
3	<b>STATUS</b>	enum('ACTIVE', 'INACTIVE')	utf8mb4_unicode_ci		No	None	
4	<b>created_at</b>	timestamp			Yes	NULL	
5	<b>updated_at</b>	timestamp			Yes	NULL	

Gambar 5.25 Struktur tabel tokens



## 26. Tabel Users

Tabel users berguna untuk menyimpan data-data user yang terdaftar pada sistem. Tabel users menyimpan data nama lengkap user, email user sebagai *key* yang unik agar user satu dengan yang lain tidak bisa memiliki email yang sama untuk login. Kemudian juga menyimpan tanggal email terverifikasi, *password* user, remember token, *role* atau hak akses user, id staff sebagai *foreign key*, id siswa sebagai *foreign key*, id wali siswa sebagai *foreign key*. Struktur tabel users digambarkan seperti pada gambar 5.26.


#	Name	Type	Collation	Attributes	Null	Default	Comments
1	<b>id</b> 🔑	int(10)		UNSIGNED	No	None	
2	<b>name</b>	varchar(191)	utf8mb4_unicode_ci		No	None	
3	<b>email</b> 🔑	varchar(191)	utf8mb4_unicode_ci		No	None	
4	<b>email_verified_at</b>	timestamp			Yes	NULL	
5	<b>password</b>	varchar(191)	utf8mb4_unicode_ci		No	None	
6	<b>remember_token</b>	varchar(100)	utf8mb4_unicode_ci		Yes	NULL	
7	<b>ROLE</b>	enum("STAFF", 'STUDENT', 'PARENT')	utf8mb4_unicode_ci		No	None	
8	<b>STAFFS_ID</b> 🔑	int(10)		UNSIGNED	Yes	NULL	
9	<b>STUDENTS_ID</b> 🔑	int(10)		UNSIGNED	Yes	NULL	
10	<b>GUARDIANS_ID</b> 🔑	int(10)		UNSIGNED	Yes	NULL	
11	<b>created_at</b>	timestamp			Yes	NULL	
12	<b>updated_at</b>	timestamp			Yes	NULL	

**Gambar 5.26 Struktur tabel users**

## 27. Tabel Violations

Tabel violations merupakan tabel master untuk menyimpan daftar pelanggaran yang berlaku disekolah. Data yang dicatat dalam tabel ini adalah nama pelanggaran, deskripsi pelanggaran, poin pelanggaran. Tabel ini berelasi *many to*

many dengan tabel students. Struktur data tabel violations dapat dilihat pada tabel 5.27.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	<b>id</b> 	int(10)		UNSIGNED	No	None	
2	<b>NAME</b>	varchar(45)	utf8mb4_unicode_ci		No	None	
3	<b>DESCRIPTION</b>	varchar(255)	utf8mb4_unicode_ci		No	None	
4	<b>POINT</b>	varchar(45)	utf8mb4_unicode_ci		No	None	

**Gambar 5.27 Struktur tabel violations**

## 28. Tabel Violation\_Records

Tabel violation records merupakan tabel hasil relasi antara tabel violation dengan tabel students. Tabel ini digunakan untuk mencatat transaksi detail pelanggaran yang dibuat siswa.

Data yang dicatat dan akan disimpan dalam tabel ini yaitu tanggal terjadi pelanggaran, total poin pelanggaran, deskripsi pelanggaran, bentuk sanksi, id unik siswa sebagai *foreign key*, id unik pelanggaran sebagai *foreign key*, id unik tahun ajaran sebagai *foreign key*, id unik staff sebagai *foreign key*, tanggal transaksi dicatat, dan tanggal transaksi diubah. Struktur data tabel violationrecords dapat dilihat pada tabel 5.28.

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	<b>id</b> 🔑	int(10)		UNSIGNED	No	None	
2	<b>DATE</b>	date			No	None	
3	<b>TOTAL</b>	double			No	None	
4	<b>DESCRIPTION</b>	text	utf8mb4_unicode_ci		No	None	
5	<b>PUNISHMENT</b>	text	utf8mb4_unicode_ci		No	None	
6	<b>STUDENTS_ID</b> 🔑	int(10)		UNSIGNED	No	None	
7	<b>VIOLATIONS_ID</b> 🔑	int(10)		UNSIGNED	No	None	
8	<b>ACADEMIC_YEAR_ID</b> 🔑	int(10)		UNSIGNED	No	None	
9	<b>STAFFS_ID</b> 🔑	int(10)		UNSIGNED	No	None	
10	<b>created_at</b>	timestamp			Yes	NULL	
11	<b>updated_at</b>	timestamp			Yes	NULL	

Gambar 5.28 Struktur tabel violation records

## 5.2 IMPLEMENTASI TAMPILAN

Implementasi tampilan akan menjelaskan setiap desain *user interface* yang telah dirancang untuk diimplementasikan. Pada subbab ini juga akan menjelaskan kegunaan dari setiap halaman yang ada, bagaimana proses tampilan berjalan.

Implementasi tampilan dibagi menjadi 3 bagian. Hal ini dilakukan untuk mendefinisikan setiap halaman yang dimiliki *user* berdasarkan *role* yang dimiliki. Penjelasan implementasi tampilan akan dijelaskan untuk *role* kepala sekolah, guru/wali kelas dan siswa.

### 5.2.1 Tampilan User Kepala Sekolah

Kepala sekolah memiliki akses penuh untuk melihat seluruh data sekolah. Namun semua halaman yang dimiliki menunjukkan bahwa kepala sekolah hanya bisa melihat informasi tetapi tidak melakukan proses input atau ubah atau hapus data. Rincian tampilan untuk kepala sekolah seperti berikut.

- **Halaman Login**

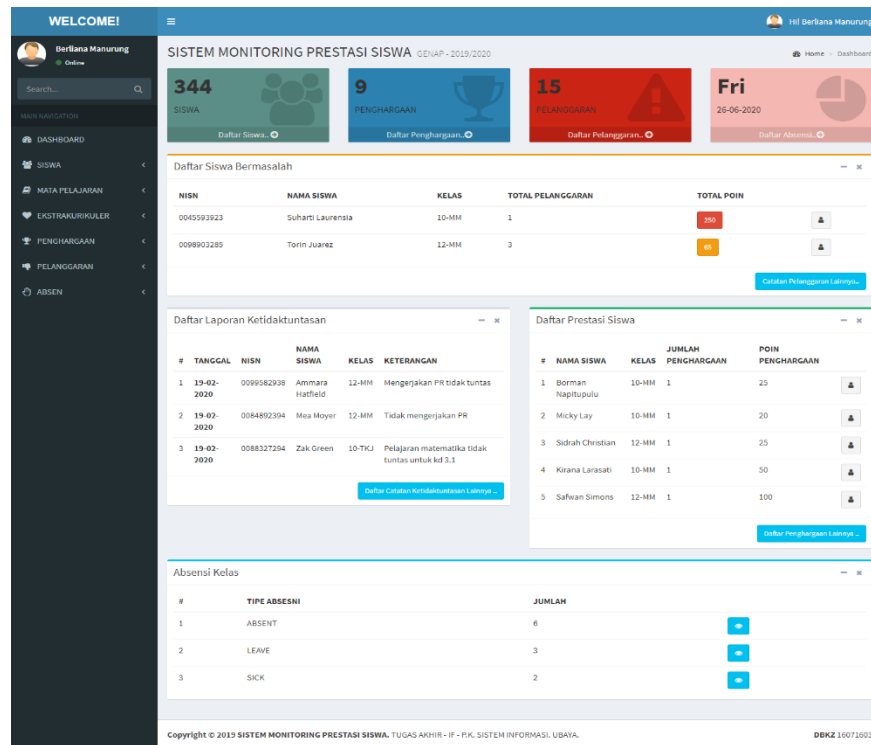
Halaman login adalah halaman pertama yang akan dijumpai *user* ketika menggunakan sistem. Terdapat dua *field* yang harus diisi oleh *user* yaitu email dan password. Tampilan halaman *dashboard* dapat dilihat pada gambar 5.29.



**Gambar 5.29 Halaman login**

- **Halaman Dashboard**

Halaman *dashboard* yang dimiliki kepala sekolah adalah *highlights*/cuplikan setiap data yang tercatat dalam sistem. Pada halaman ini diberikan informasi jumlah keseluruhan siswa, jumlah total penghargaan keseluruhan, jumlah total pelanggaran keseluruhan. Kemudian terdapat 4 kolom lagi yaitu informasi daftar siswa bermasalah, informasi daftar laporan ketidaktuntasan, informasi daftar prestasi siswa, dan informasi daftar absen siswa. Gambar 5.30 menunjukkan tampilan halaman *dashboard* bagi kepala sekolah.



**Gambar 5.30 Tampilan halaman dashboard kepala sekolah**

- Halaman Daftar Siswa**

Halaman daftar siswa akan menampilkan daftar nama-nama siswa seluruh siswa yang terdaftar di sekolah. Kepala sekolah bisa melakukan filter daftar nama siswa berdasarkan kelas yang tersedia pada *combobox*. Daftar nama-nama siswa ditampilkan pada tabel. Tabel yang tersedia menyediakan 2 tombol aksi yang bisa digunakan oleh kepala sekolah untuk melihat halaman profil siswa atau halaman mata pelajaran yang dimiliki siswa. Gambar 5.31 menunjukkan tampilan halaman daftar siswa untuk kepala sekolah.

WELCOME! | Hi Berikana Manurung

Berikana Manurung | Online

DAFTAR NAMA SISWA

DAFTAR KELAS: 10-MM

Show 10 entries

#	NISN	NAMA SISWA	KELAS	ANGKATAN
1	0084930204	Tri Magdalena	10-MM	2019/2020
2	0027383342	Purwanti Asdik	10-MM	2019/2020
3	0082738949	Kirana Larasati	10-MM	2019/2020
4	0017377288	Wilson Alfonso	10-MM	2019/2020
5	0093720083	Micky Lay	10-MM	2019/2020
6	0011777288	Borman Napilupulu	10-MM	2019/2020
7	0030019938	Robert Kristanto	10-MM	2019/2020
8	0045553923	Suharti Laurensia	10-MM	2019/2020
9	0088590020	Kornelius Saputra	10-MM	2019/2020
10	0069949274	Monica Clay	10-MM	2019/2020

Showing 1 to 10 of 21 entries

Previous 1 2 3 Next

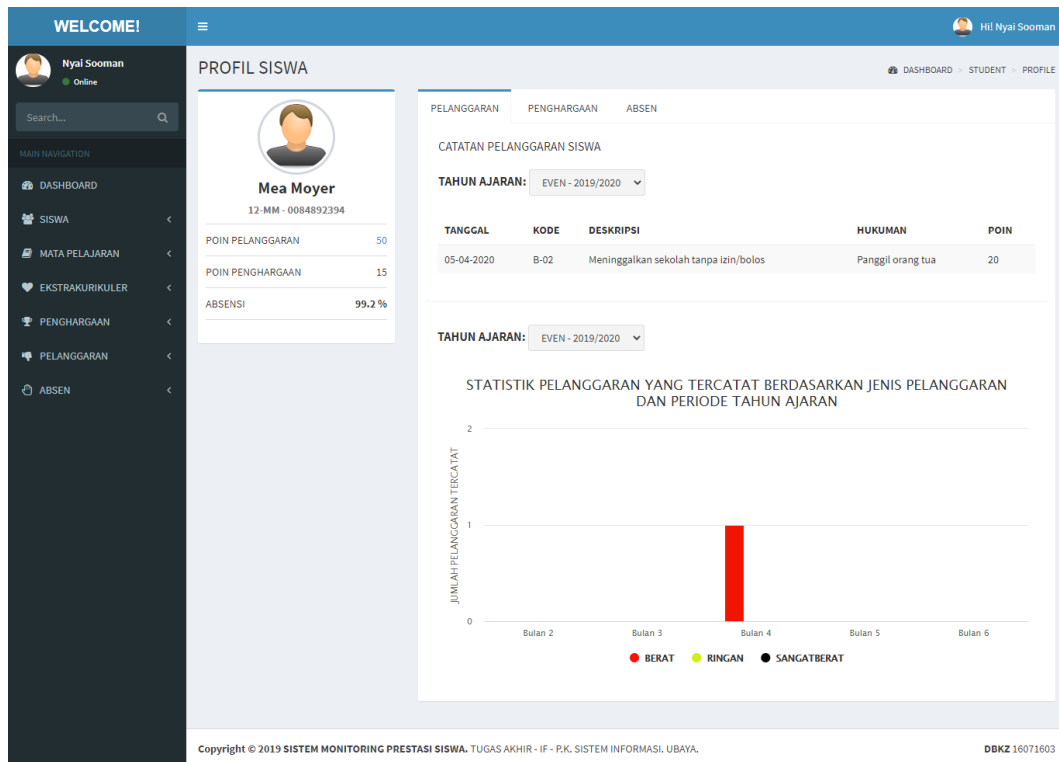
Copyright © 2019 SISTEM MONITORING PRESTASI SISWA, TUGAS AKHIR - IF - P.K. SISTEM INFORMASI UBAYA. DBKZ 16071603

**Gambar 5.31 Tampilan halaman daftar siswa kepala sekolah**

- **Halaman Profil Siswa**

Halaman profil siswa yang adalah halaman yang menampilkan informasi mengenai catatan penghargaan, catatan pelanggaran, dan catatan absen siswa yang merupakan halaman untuk memantau nilai karakter siswa. Halaman ini terbagi atas 2 kolom kiri dan kanan.

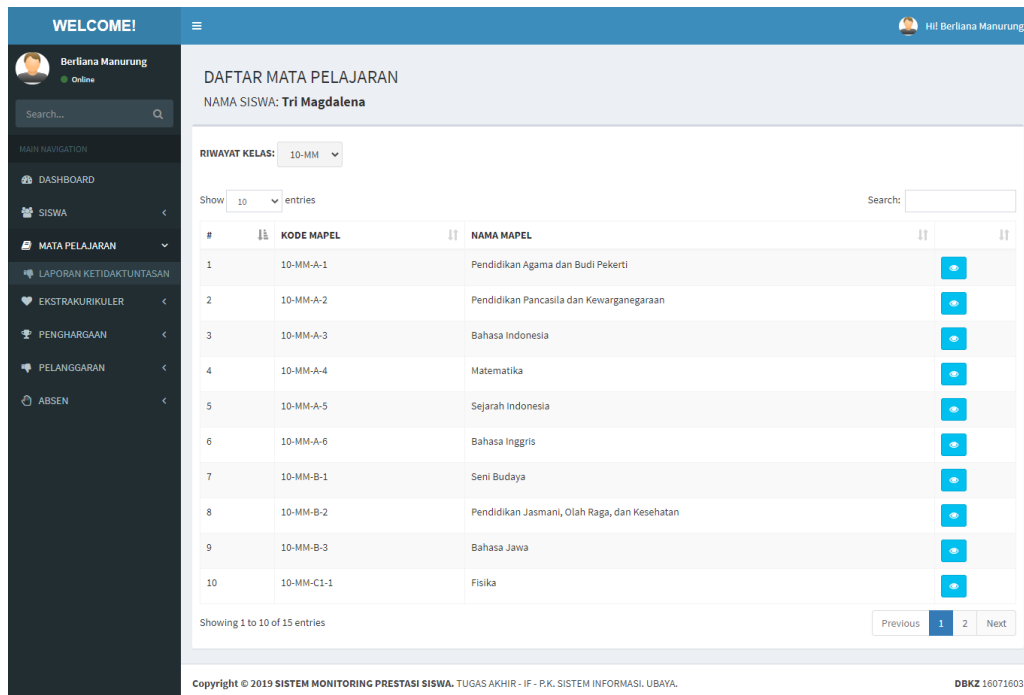
Kolom kiri memberikan informasi poin pelanggaran siswa selama aktif di sekolah, informasi poin penghargaan siswa selama aktif di sekolah dan informasi persentase absen untuk setiap tahun ajarannya. Sedangkan kolom kanan adalah rincian data-data atau detail informasi pelanggaran, penghargaan, dan daftar absensi setiap siswa. Halaman profil siswa dapat dilihat pada gambar 5.32.



**Gambar 5.32 Tampilan halaman profil siswa kepala sekolah**

- **Halaman Daftar Mata Pelajaran Siswa**

Halaman daftar mata pelajaran siswa adalah halaman rincian mata pelajaran yang dimiliki setiap siswa. Kepala sekolah memiliki akses untuk melihat daftar mata pelajaran siswa. Selain itu kepala sekolah dapat melihat bahkan *record* mata pelajaran siswa tidak hanya pada satu kelas, namun riwayat kelas yang sudah dimiliki siswa. Tampilan halaman daftar mata pelajaran siswa dapat dilihat pada gambar 5.33.

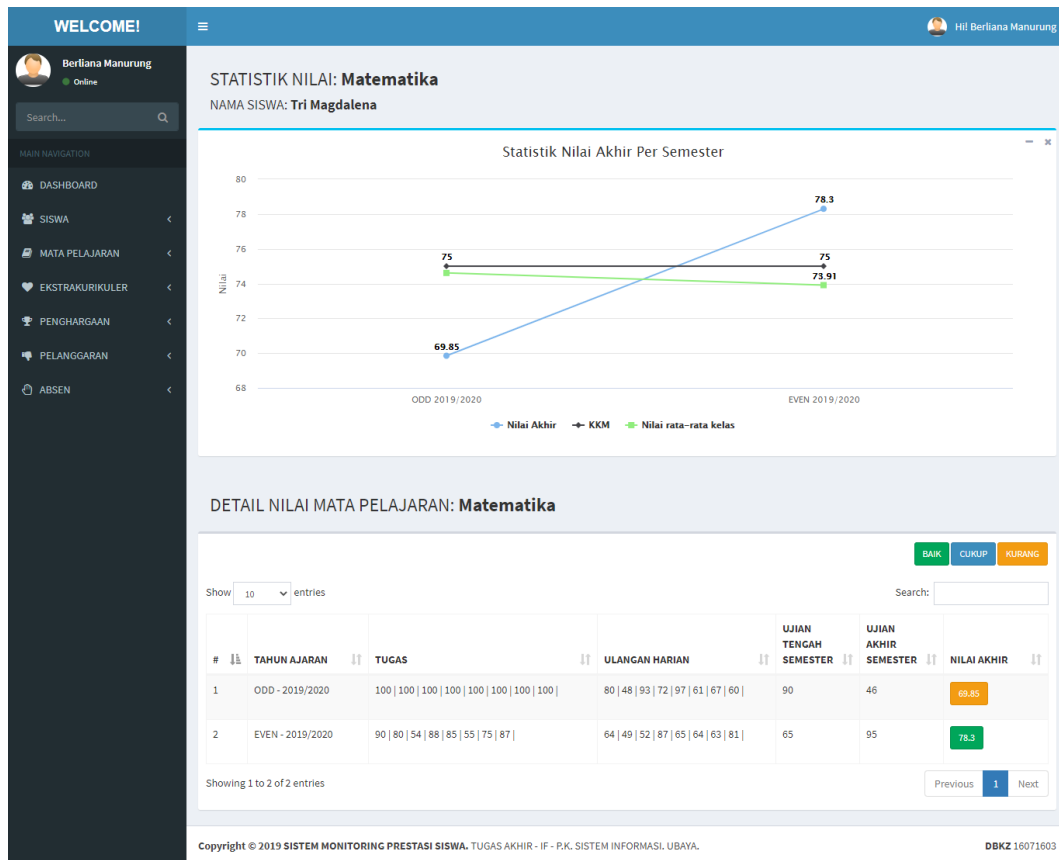


**Gambar 5.33 Tampilan halaman daftar mata pelajaran siswa kepala sekolah**

- **Halaman Detail Mata Pelajaran**

Halaman detail mata pelajaran bisa di akses dari halaman daftar mata pelajaran dengan menekan tombol dengan *icon* mata. Halaman ini berguna menampilkan detail setiap mata pelajaran yang dimiliki siswa. Detail yang ditampilkan seperti grafik nilai dan juga tabel rincian nilai. Halaman detail mata pelajaran dapat dilihat pada gambar 5.34.





**Gambar 5.34 Tampilan halaman detail mata pelajaran siswa kepala sekolah**

- **Halaman Daftar Laporan Ketidaktuntasan Siswa**

Halaman daftar laporan ketidaktuntasan memberikan informasi daftar-daftar ketidaktuntasan. Rincian data yang ditampilkan adalah tanggal tercatat, nama siswa, kelas, kode, dan deksripsi. Halaman ini dapat dilihat seperti pada gambar 5.35.

#	TANGGAL	NAMA SISWA	KODE	DEKSRIPSI
1	19-02-2020	Zak Green	TTS	Pelajaran matematika tidak tuntas untuk kd 3.1
2	19-02-2020	Mea Moyer	TTS	Tidak mengerjakan PR
3	19-02-2020	Ammara Hatfield	TTS	Mengerjakan PR tidak tuntas

**Gambar 5.35** Tampilan halaman daftar laporan ketidaktuntasan siswa kepala sekolah

- **Halaman Daftar Master Ekstrakurikuler**

Halaman daftar master ekstrakurikuler adalah daftar ekstrakurikuler yang terdapat di sekolah. Halaman ini memberikan detail nama ekstrakurikuler dan deskripsi ekstrakurikuler. Halaman ini juga bisa halaman yang bisa dimiliki oleh *role* guru/wali kelas. Gambar 5.36 menunjukkan daftar master ekstrakurikuler.

#	NAMA EKSUL	DESKRIPSI
1	Pramuka	Ekskul pramuka
2	Basket	Ekskul basket putra dan putri
3	Futsal	Ekskul futsal putra
4	Karate	Ekskul karate

**Gambar 5.36** Tampilan halaman daftar master ekstrakurikuler kepala sekolah

- **Halaman Daftar Catatan Ekskul**

Halaman daftar catatan ekskul adalah *report* nilai dan kegiatan ekskul yang diambil siswa. Halaman ini memberikan informasi nama kelas, nama siswa, nama ekskul, tahun ajaran dan nilainya. Kepala sekolah juga bisa melakukan filter data berdasarkan tahun ajaran dengan *combobox* filter tahun ajaran yang telah disediakan. Halaman daftar catatan ekskul dapat dilihat pada gambar 5.37.

#	KELAS	NAMA SISWA	EKSUL	TAHUN AJARAN	NILAI
1	10-MM	Purwanti Asidik	Pramuka	EVEN - 2019/2020	80
2	12-MM	Ammara Hatfield	Pramuka	EVEN - 2019/2020	77
3	12-MM	Mea Moyer	Pramuka	EVEN - 2019/2020	90
4	12-MM	Colleen Berg	Pramuka	EVEN - 2019/2020	75

**Gambar 5.37 Tampilan halaman daftar catatan ekskul kepala sekolah**

- **Halaman Daftar Penghargaan**

Halaman daftar penghargaan adalah halaman yang menampilkan data master penghargaan yang terdapat di sekolah. Halaman ini juga bisa halaman yang bisa dimiliki oleh *role* guru/wali kelas. Halaman daftar penghargaan dapat dilihat pada gambar 5.38 yang memberikan informasi tentang tipe penghargaan, deskripsi, tingkat, dan poin.

**WELCOME!** Hi! Berliana Manurung

**DAFTAR PENGHARGAAN**

Show 10 entries Search:

#	TIPE	DEKSRIPSI	TINGKAT	POIN
1	PR	Kejuaraan, kompetisi atau konser tingkat nasional	Nasional	100
2	PR	Kejuaraan, kompetisi atau konser tingkat propinsi	Propinsi	75
3	PR	Kejuaraan, kompetisi atau konser tingkat kota/kabupaten	Kota	50
4	PR	Kejuaraan, kompetisi atau konser tingkat kecamatan	Kota	25
5	PR	Kejuaraan, kompetisi atau konser sebagai peserta (tidak juara)	Kota	10
6	PR	Mengikuti LDKS	Sekolah	15
7	PR	Diangkat menjadi pengurus OSIS	Sekolah	20
8	PR	Diangkat menjadi ketua OSIS	Sekolah	25
9	NPR	Tidak pernah alpha (HANYA bagi siswa yang catatan pelanggaran bersih)	Sekolah	25
10	NPR	Tidak pernah terlambat 1 bulan berturut-turut (HANYA bagi siswa yang catatan pelanggaran bersih)	Sekolah	15

Showing 1 to 10 of 10 entries Previous 1 Next

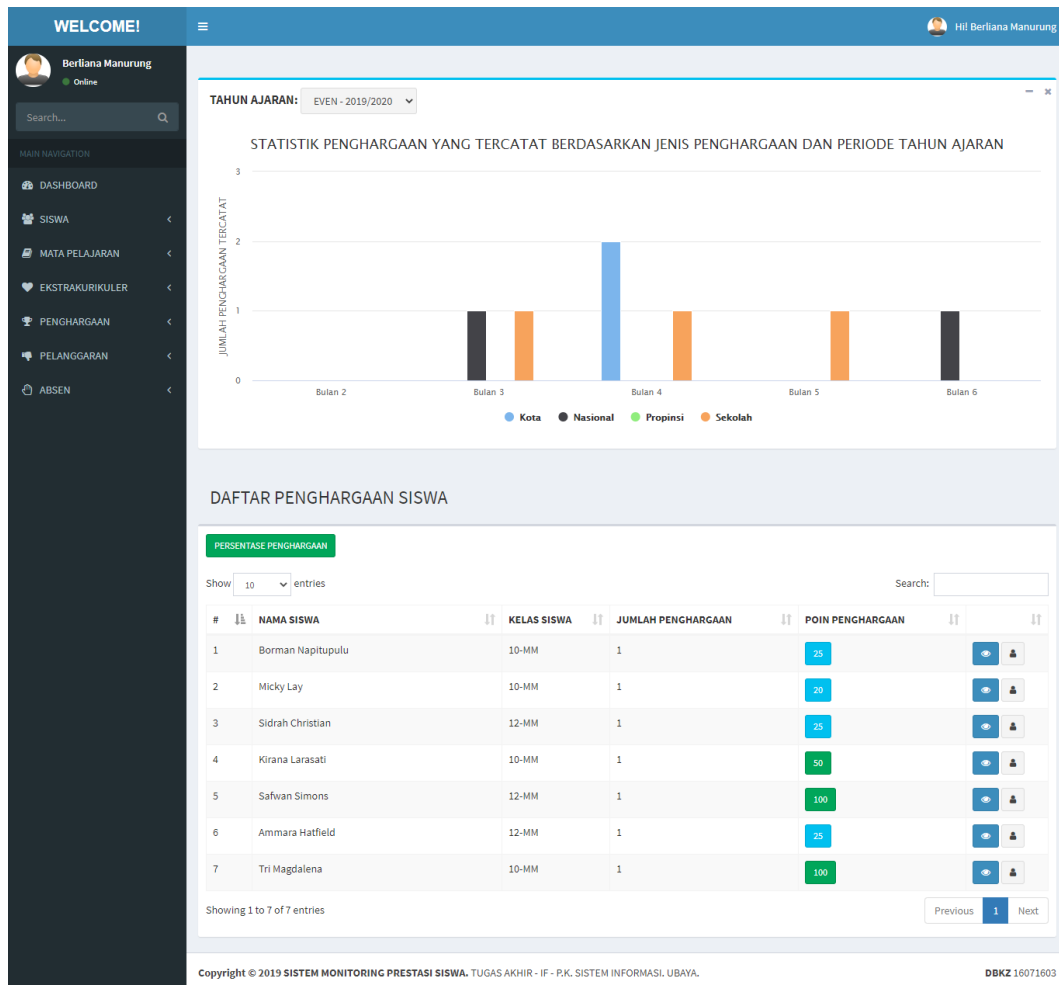
Copyright © 2019 SISTEM MONITORING PRESTASI SISWA, TUGAS AKHIR - IF - P.K. SISTEM INFORMASI, UBAYA. DBKZ 16071603

**Gambar 5.38 Tampilan halaman daftar penghargaan kepala sekolah**

- **Halaman Daftar Catatan Penghargaan**

Halaman daftar catatan penghargaan adalah halaman yang berisi *record* penghargaan yang diperoleh siswa. Kepala sekolah memiliki akses penuh untuk menerima informasi daftar catatan penghargaan yang seluruhnya tercatat dalam *database* berdasarkan tahun ajaran.

Detail informasi yang ditampilkan adalah kolom grafik yang merupakan informasi statistik penghargaan yang tercatat berdasarkan jenis penghargaan dan periode tahun ajaran. Sedangkan tabel daftar penghargaan siswa adalah rincian berupa nama siswa, kelas, jumlah penghargaaa dan poin penghargaan masing-masing siswa. Kepala sekolah juga dapat melihat detail catatan penghargaan yang dimiliki siswa dengan menekan tombol *icon* mata yang akan ditampilkan dalam bentuk modal. Halaman catatan penghargaan dapat dilihat seperti pada gambar 5.39.



**Gambar 5.39 Tampilan halaman daftar catatan penghargaan kepala sekolah**

Terdapat detail informasi lain yang berada pada halaman ini. Jika kepala sekolah menekan tombol hijau persentase penghargaan akan muncul sebuah tabel informasi persentase penghargaan. Tabel ini memberikan data persentase penghargaan setiap tingkatnya hasil akumulasi jumlah catatan penghargaan untuk tiap-tiap tingkat yang dibagi dengan jumlah master data penghargaan.

Kemudian dalam tabel persentase penghargaan tersebut terdapat satu tombol aksi lagi yang berguna untuk melihat detail penghargaan apa sajakah yang

tercatat untuk setiap tingkat. Tabel ini memberikan informasi untuk detail tiap-tiap tingkat. Tampilan tabel persentase penghargaan dan detail penghargaan bersarkan tingkat dapat dilihat seperti pada gambar 5.40.

PERSENTASE PENGHARGAAN			
#	KATEGORI	JUMLAH TERCATAT	PERSENTASE
1	Sekolah	3	30 %
2	Nasional	2	20 %
3	Kota	2	20 %
*Keterangan: Persentase penghagraan adalah berdasarkan jumlah penghargaan tercatat untuk setiap tingkatnya yang dibagi dengan jumlah daftar penghargaan yang berlaku			

PENGHARGAAN BERDASARKAN TINGKAT			
TINGKAT	NAMA PENGHARGAAN	JUMLAH TERJADI	
Sekolah	Diangkat menjadi pengurus OSIS	1	
Sekolah	Diangkat menjadi ketua OSIS	2	

**Gambar 5.40 Tabel persentase penghargaan dan detail tiap tingkat**

- **Halaman Daftar Pelanggaran**

Halaman daftar pelanggaran adalah halaman yang menampilkan data master pelanggaran yang berlaku di sekolah. Halaman ini juga bisa halaman yang bisa dimiliki oleh *role* guru/wali kelas. Halaman daftar pelanggaran dapat dilihat pada gambar 5.41 yang memberikan informasi tentang kode pelanggaran, deskripsi dan poin.

Kolom pelanggaran juga memberikan *insight* yang memiliki arti. Jika kode pelanggaran ‘R’ maka pelanggaran tersebut masuk ke dalam kategori ringan. Jika kode pelanggaran ‘B’ maka pelanggaran tersebut masuk ke dalam kategori berat. Jika kode pelanggaran ‘SB’ maka pelanggaran tersebut masuk ke dalam kategori sangat berat.

**WELCOME!** Hti Berliana Manurung

**DAFTAR PELANGGARAN**

Show 10 entries Search:

#	KODE PELANGGARAN	DESKRIPSI	POIN
1	R-01	Tidak memakai ikat pinggan	10
2	R-02	Ikat pinggan tidak berwarna sesuai ketentuan	10
3	R-03	Tidak memakai topi/atribut sesuai ketentuan saat upacara	10
4	R-04	Tidak memakai sepatu hitam sesuai ketentuan	10
5	R-05	Memakai kaos kaki dengan warna bukan putih/tidak sesuai ketentuan	10
6	B-01	Datang terlambat	20
7	B-02	Meninggalkan sekolah tanpa izin/bolos	20
8	B-03	Menghina/melecehkan teman, guru, kepala sekolah dan atau karyawan	25
9	B-04	Membawa/mengonsumsi rokok di lingkungan sekolah	100
10	B-05	Merokok di luar sekolah dengan memakai baju/atribut SMK St. Louis	100

Showing 1 to 10 of 15 entries Previous 1 2 Next

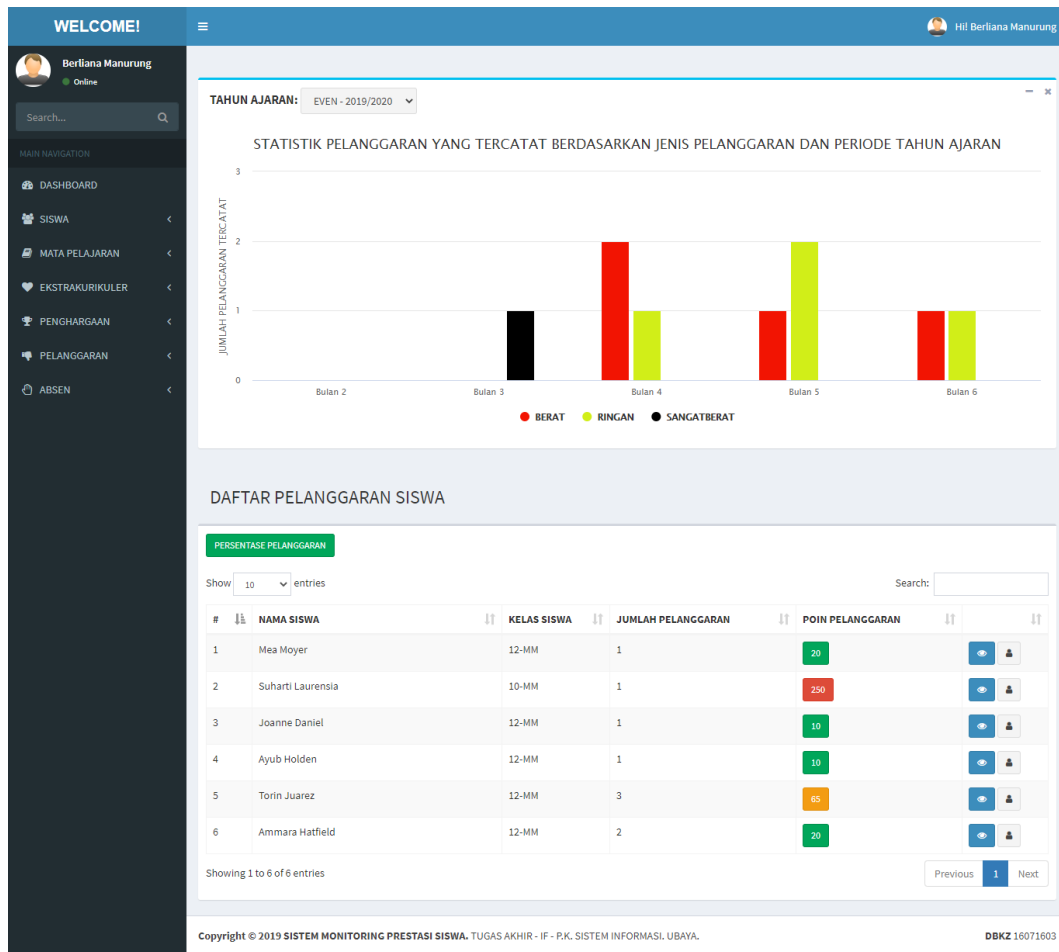
Copyright © 2019 SISTEM MONITORING PRESTASI SISWA. TUGAS AKHIR - IF - P.K. SISTEM INFORMASI, UBAYA. DBKZ 16071603

**Gambar 5.41 Tampilan halaman daftar pelanggaran kepala sekolah**

- **Halaman Daftar Catatan Pelanggaran**

Halaman daftar catatan pelanggaran adalah halaman yang berisi *record* pelanggaran yang dimiliki siswa. Kepala sekolah memiliki akses penuh untuk menerima informasi daftar catatan pelanggaran yang seluruhnya tercatat dalam *database* berdasarkan tahun ajaran.

Detail informasi yang ditampilkan adalah kolom grafik yang merupakan informasi statistik pelanggaran yang tercatat berdasarkan jenis pelanggaran dan periode tahun ajaran. Sedangkan tabel daftar pelanggaran siswa adalah rincian berupa nama siswa, kelas, jumlah pelanggaran dan poin pelanggaran masing-masing siswa. Kepala sekolah juga dapat melihat detail catatan pelanggaran yang dimiliki siswa dengan menekan tombol *icon* mata yang akan ditampilkan dalam bentuk modal. Halaman catatan pelanggaran dapat dilihat seperti pada gambar 5.42.

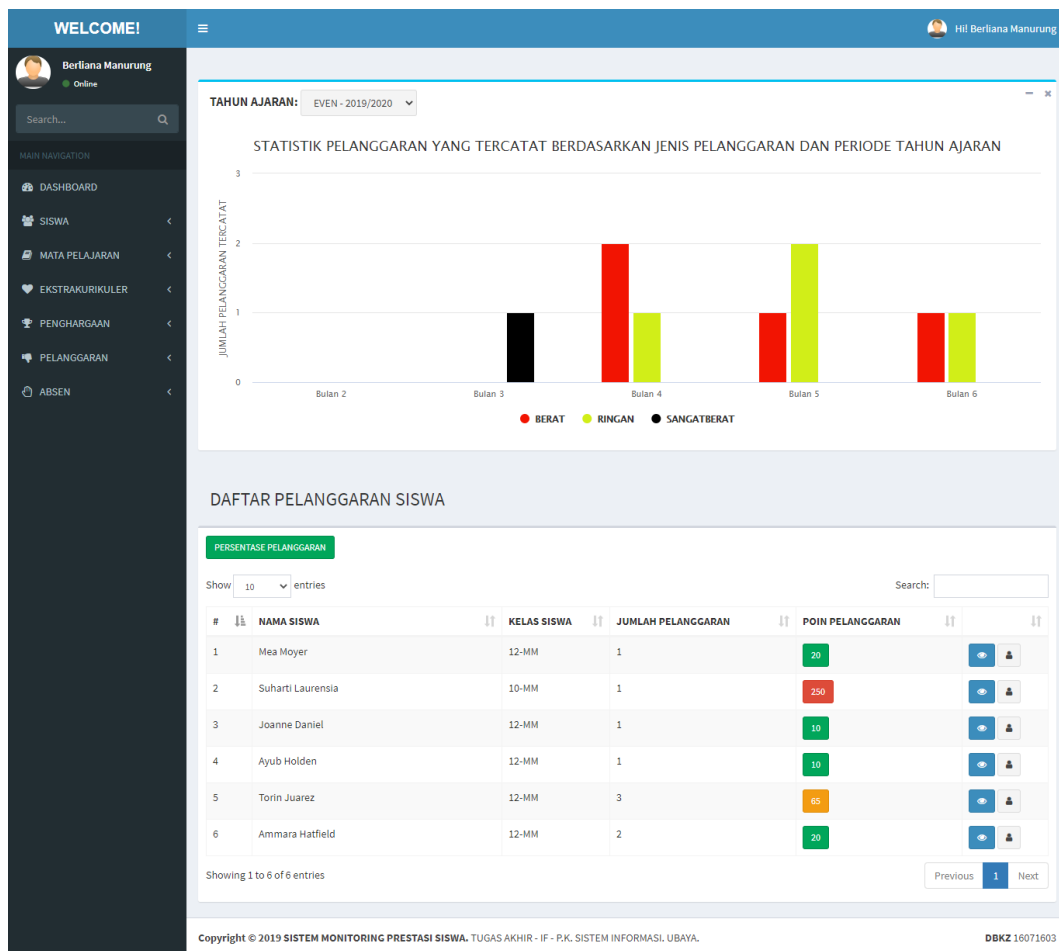


**Gambar 5.42 Tampilan halaman daftar catatan pelanggaran kepala sekolah**

Terdapat detail informasi lain yang berada pada halaman ini. Jika kepala sekolah menekan tombol hijau persentase pelanggaran akan muncul sebuah tabel informasi persentase pelanggaran. Tabel ini memberikan data persentase pelanggaran setiap kategorinya hasil akumulasi jumlah catatan pelanggaran untuk tiap-tiap tingkat yang dibagi dengan jumlah master data pelanggaran.

Kemudian dalam tabel persentase pelanggaran tersebut terdapat satu tombol aksi lagi yang berguna untuk melihat detail pelanggaran apa sajakah yang tercatat untuk setiap kategori. Tabel ini memberikan informasi untuk detail tiap-tiap





**Gambar 5.42 Tampilan halaman daftar catatan pelanggaran kepala sekolah**

Terdapat detail informasi lain yang berada pada halaman ini. Jika kepala sekolah menekan tombol hijau persentase pelanggaran akan muncul sebuah tabel informasi persentase pelanggaran. Tabel ini memberikan data persentase pelanggaran setiap kategorinya hasil akumulasi jumlah catatan pelanggaran untuk tiap-tiap tingkat yang dibagi dengan jumlah master data pelanggaran.

Kemudian dalam tabel persentase pelanggaran tersebut terdapat satu tombol aksi lagi yang berguna untuk melihat detail pelanggaran apa sajakah yang tercatat untuk setiap kategori. Tabel ini memberikan informasi untuk detail tiap-tiap

kategori. Tampilan tabel persentase pelanggaran dan detail pelanggaran berdasarkan kategori dapat dilihat seperti pada gambar 5.43.

PERSENTASE PELANGGARAN			
#	KATEGORI	JUMLAH TERCATAT	PERSENTASE
1	SANGATBERAT	1	20 %
2	RINGAN	4	80 %
3	BERAT	4	80 %
*Keterangan: Persentase pelanggaran adalah berdasarkan jumlah pelanggaran tercatat untuk setiap kategorinya yang dibagi dengan jumlah daftar pelanggaran yang berlaku			

PELANGGARAN BERDASARKAN KATEGORI			
KODE PELANGGARAN	NAMA PELANGGARAN	JUMLAH TERJADI	
R-01	Tidak memakai ikat pinggan	1	
R-04	Tidak memakai sepatu hitam sesuai ketentuan	2	
R-05	Memakai kaos kaki dengan warna bukan putih/tidak sesuai ketentuan	1	

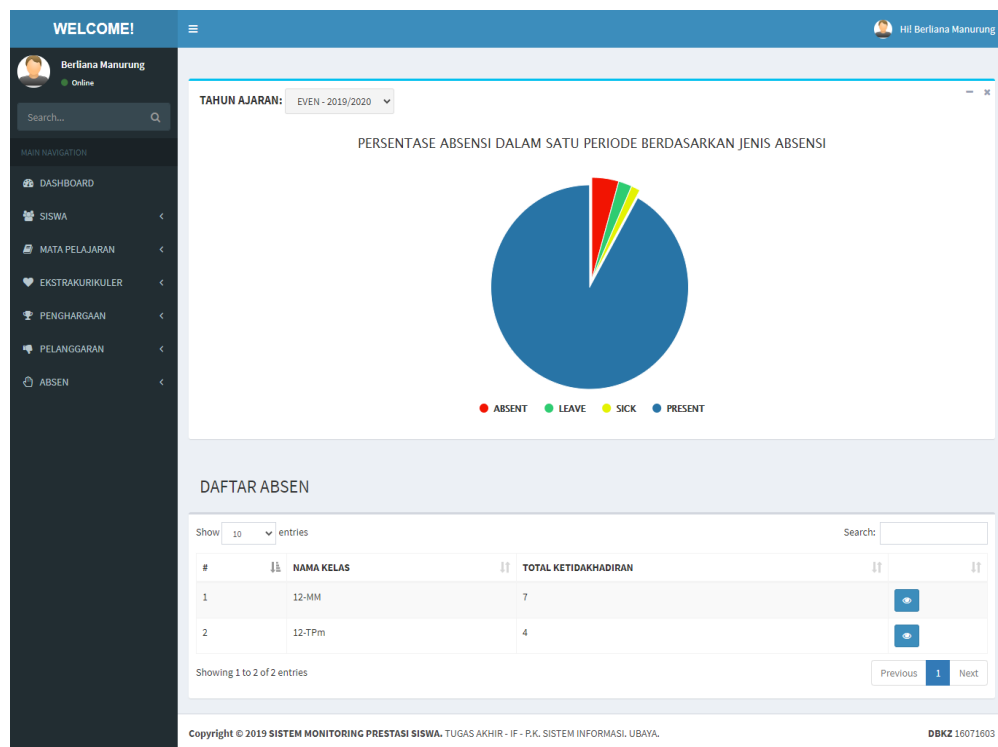
**Gambar 5.43 Tabel persentase pelanggaran dan detail tiap tingkat**

- **Halaman Daftar Absen**

Halaman daftar absen adalah halaman informasi catatan-catatan ketidakhadiran siswa. Halaman ini memberikan informasi tentang grafik persentase setiap jenis absen dan daftar absen dalam tabel. Halaman ini juga adalah halaman yang dapat diakses oleh guru/wali kelas. Namun terdapat perbedaan yaitu isi informasi yang disampaikan. Kepala sekolah dapat melihat catatan absen seluruh kelas sedangkan guru/wali kelas sebatas kelasnya saja.

Daftar tabel yang menjadi penanda perbedaan isi informasi yang disampaikan. Kepala sekolah akan dimulai dengan sebuah tabel rincian pertama yaitu tabel daftar absen berisi nama kelas, dan total kehadiran tiap kelas. Tabel ini dapat dilihat pada gambar 5.44. Kemudian kepala sekolah dapat menekan tombol aksi pada kolom paling kanan yang akan menampilkan modal seperti pada gambar 5.45. Tahap terakhir yang dimiliki kepala sekolah ketika modal daftar absen kelas muncul, kepala sekolah masih dapat melihat rinciannya dengan menekan tombol

aksi kembali. Rincian terakhir dalam bentuk modal yang dapat dilihat adalah daftar siswa yang tidak hadir berdasarkan tipe absensi. Modal detail ketidak hadiran dapat dilihat seperti pada gambar 5.46.



**Gambar 5.44 Tampilan halaman daftar absen kepala sekolah**

DAFTAR ABSEN KELAS		
TIPE ABSENSI	JUMLAH KETIDAKHADIRAN/TIPE	
ABSENT	5	
LEAVE	1	
SICK	1	

**Gambar 5.45 Modal detail daftar absen per kelas**



The image shows a modal window titled "DETAIL ABSEN" with a close button (X) in the top right corner. Inside the modal is a table with the following data:

ID	NAMA SISWA	TIPE ABSENSI	TAHUN AJARAN	TANGGAL AWAL	TANGGAL AKHIR	KETERANGAN
18	Mea Moyer	ABSENT	EVEN-2019/2020	2020-02-13	2020-02-13	Tidak hadir tanpa alasan
19	Maggie Singh	ABSENT	EVEN-2019/2020	2020-02-13	2020-02-13	Tidak hadir tanpa alasan
21	Torin Juarez	ABSENT	EVEN-2019/2020	2020-02-20	2020-02-20	Tidak hadir tanpa alasan
22	Isma Higgs	ABSENT	EVEN-2019/2020	2020-03-20	2020-03-20	Tidak hadir tanpa alasan
24	Rianne Kane	ABSENT	EVEN-2019/2020	2020-04-11	2019-04-11	Tidak hadir tanpa alasan

**Gambar 5.46** Modal detail absen berdasarkan tipe absen kepala sekolah

Ketiga rincian gambar sebelumnya adalah informasi yang berhak dimiliki kepala sekolah. Kalau kepala sekolah bisa memiliki 2 tahap untuk melihat informasi tentang absen sedangkan guru/wali kelas hanya 1 tahap. Hal ini dikarenakan isi informasi absen untuk guru/wali kelas hanya sebatas *scope* kelasnya. Tampilan awal yang diterima langsung pada daftar absen perkelas dengan tipe absen dan jumlah ketidakhadirannya. Namun hal ini akan dijelaskan lebih rinci lagi pada poin berikutnya yang membahas tampilan halaman guru/wali kelas.

### 5.3.2 Tampilan User Guru/Wali Kelas

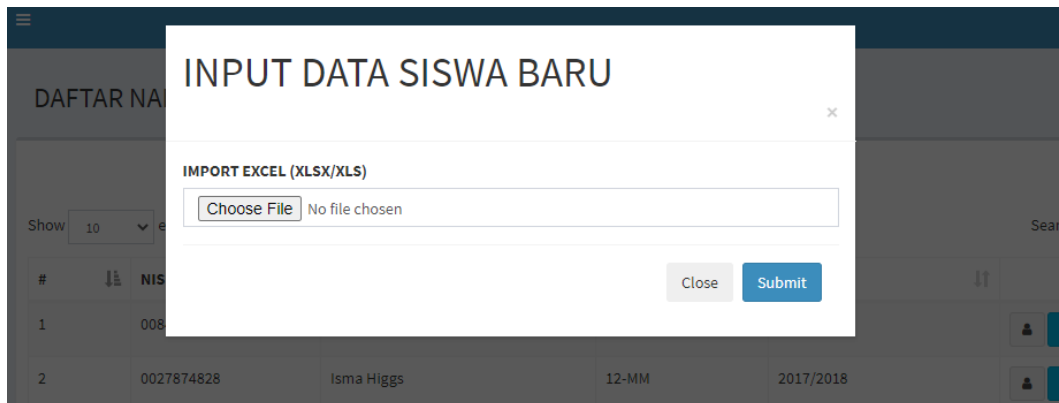
Secara umum, tampilan untuk *user role* guru/wali kelas sama dengan tampilan yang dimiliki kepala sekolah. Tampilan halaman yang dimiliki guru/wali kelas melingkupi halaman seperti halaman *dashboard*, halaman daftar siswa, halaman profil siswa, halaman daftar mata pelajaran siswa, halaman daftar detail mata pelajaran siswa, halaman daftar laporan ketidaktuntasan siswa, halaman master ekskul, halaman catatan nilai ekskul siswa, halaman master penghargaan, halaman catatan penghargaan, halaman master pelanggaran, halaman catatan

pelanggaran, dan halaman daftar absen. Namun, dari ke-13 halaman yang disebutkan hanya ada perbedaan isi informasi yang ditampilkan. Jika setiap halaman yang dimiliki kepala sekolah memberikan informasi keseluruhan data-data siswa ataupun catatan lainnya yang tersimpan dalam *database*, maka untuk guru/wali kelas isi informasi untuk ke-13 halaman yang sama tersebut hanya menampilkan data siswa atau catatan-catatan siswa untuk *scope* siswa yang terdaftar dikelasnya.

Oleh karena itu poin dari sub-bab kali ini akan menjelaskan daftar halaman lain yang dimiliki guru/wali kelas yang tidak dimiliki kepala sekolah. Adapun halaman lain tersebut secara garis besar adalah halaman untuk melakukan proses input data. Halaman yang akan dibahas adalah halaman-halaman *input* dan *update* data. Maka dari itu penjelasan dibawah ini akan menjelaskan halaman proses untuk modul siswa, modul mata pelajaran, modul ketidaktuntasan, modul ekskul, modul penghargaan dan modul pelanggaran.

- **Halaman proses modul siswa**

Hanya terdapat 1 buah proses untuk modul siswa dalam sistem *monitoring* yang dibuat. Halaman proses modul siswa bagi guru/wali kelas adalah halaman input daftar siswa. Halaman ini sebenarnya berupa modal yang menimpa halaman daftar siswa. Modal input daftar siswa memberikan satu buah input berupa input file yang mengimport sebuah file excel kedalam sistem. Modal input daftar siswa dapat dilihat seperti pada gambar 5.47.



**Gambar 5.47 Modal input data siswa**

- **Halaman proses modul mata pelajaran**

Terdapat 2 halaman informasi tambahan pada modul mata pelajaran untuk guru/wali kelas yaitu halaman daftar mata pelajaran guru dan halaman detail mata pelajaran. Halaman daftar mata pelajaran guru dapat dilihat pada gambar 5.48 yang memberi informasi mata pelajaran yang diajar dan dimiliki guru. Sedangkan halaman detail mata pelajaran guru dapat dilihat pada gambar 5.49.

Halaman daftar mata pelajaran guru berisi informasi kode mapel dan nama mapel serta 2 tombol aksi yang diberikan dalam bentuk tabel. Sedangkan halaman detail mata pelajaran guru berisi informasi detail rincian nilai mata pelajaran siswa untuk mapel tertentu dan grafik sebagai statistic nilai akhir. Grafik memberikan informasi dengan 3 indikator yaitu data nilai rata-rata kelas, data jumlah siswa yang nilai akhirnya dibawah rata kelas dan nilai ketuntasan minimal.

**WELCOME!** Hii Nyai Sooman

**MATA PELAJARAN-GURU**

Show 10 entries Search:

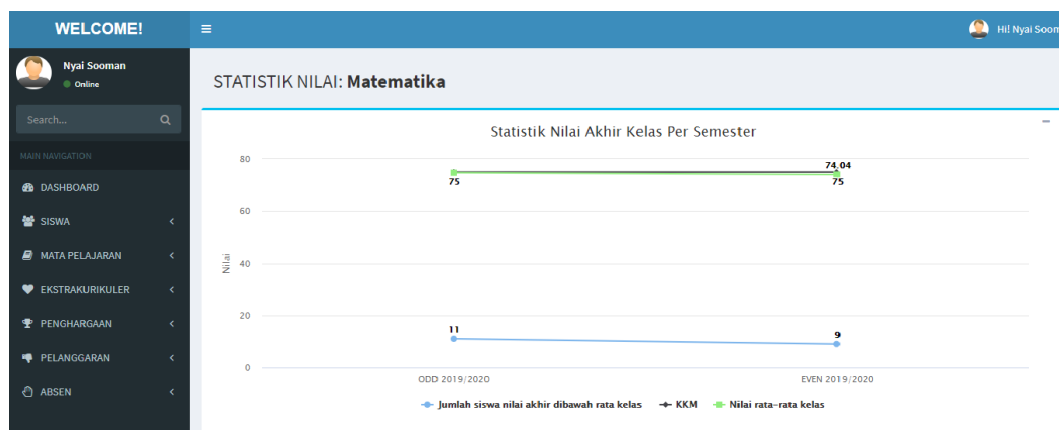
#	KODE MAPEL	NAMA MAPEL	
1	12-MM-A-1	Pendidikan Agama dan Budi Pekerti	
2	12-MM-A-2	Pendidikan Pancasila dan Kewarganegaraan	
3	12-MM-A-3	Bahasa Indonesia	
4	12-MM-A-4	Matematika	
5	12-MM-A-6	Bahasa Inggris	
6	12-MM-B-3	Bahasa Jawa	
7	12-MM-C3-2	Teknik Pengolahan Audio Video	
8	12-MM-C3-4	Design Media Interaktif	
9	12-MM-C3-5	Produk Kreatif dan Kewirausahaan	

Showing 1 to 9 of 9 entries

Previous 1 Next

Copyright © 2019 SISTEM MONITORING PRESTASI SISWA. TUGAS AKHIR - IF - P.K. SISTEM INFORMASI UBAYA. DBKZ 16071603

Gambar 5.48 Tampilan halaman daftar mata pelajaran guru



Gambar 5.49 Tampilan halaman detail mata pelajaran guru

DETAIL NILAI MATA PELAJARAN: Matematika									
TAHUN AJARAN: <span>EVEN - 2019/2020</span>			<span>BAIK</span> <span>CUKUP</span> <span>KURANG</span>						
Show <span>10</span> entries			Search: <input type="text"/>						
#	NISN	NAMA SISWA	TUGAS	ULANGAN HARIAN	UJIAN TENGAH SEMESTER	UJIAN AKHIR SEMESTER	NILAI AKHIR		
1	0084892394	Mea Moyer	93   71   86   54   93   62   82   92	93   97   62   92   81   89   67   100	53	83	78.34		
2	0027874828	Isma Higgs	73   84   49   83   95   91   95   71	89   70   98   55   67   93   76   61	85	92	74.68		
3	0038589893	Joanne Daniel	77   46   74   99   52   95   63   91	68   90   91   63   93   90   95   95	73	86	73.67		
4	009582938	Ammara Hatfield	56   81   47   47   52   63   59   91	70   96   48   61   48   51   54   95	100	65	72.97		
5	0016673789	Safwan Simons	66   87   48   83   60   63   49   69	85   85   53   45   46   71   95   50	80	93	75.68		
6	0026748721	Keelan Gonzalez	73   45   94   85   72   55   70   69	72   81   45   57   79   60   85   53	66	69	74.23		
7	0012723413	Lyndon Bowden	99   64   70   62   64   89   56   91	45   61   87   70   83   47   90   70	86	76	76.44		
8	0098903285	Torin Juarez	93   73   60   57   66   79   75   83	87   53   86   80   58   77   52   92	81	78	71.25		
9	0026735929	Sidrah Christian	55   74   70   48   86   80   66   66	49   66   89   76   59   98   91   51	92	68	76.09		
10	0023724734	Tiah Lucero	46   45   69   81   67   45   86   59	58   69   68   52   63   59   66   60	72	45	73.85		
Showing 1 to 10 of 19 entries							<span>Previous</span> <span>1</span> <span>2</span> <span>Next</span>		

Copyright © 2019 SISTEM MONITORING PRESTASI SISWA. TUGAS AKHIR - IF - P.K. SISTEM INFORMASI UBAYA.

DBKZ 16071603

**Gambar 5.49 Tampilan halaman detail mata pelajaran guru (lanjutan)**

Kemudian terdapat 2 halaman proses untuk modul mata pelajaran bagi guru/wali kelas yaitu halaman input nilai dan halaman ubah nilai. Halama input nilai pun menjadi 2 bagian karena menjadi halama proses. Bagian pertama yaitu tampilan input nilai pada gambar 5.50 dan tampilan daftar nilai yang diinput pada gambar 5.51 dalam sebuah halaman proses yang sama.

Bagian input nilai berupa satu input bertipe file untuk mengimport sebuah file excel nilai keadalam sistem. Sedangkan bagian kedua yaitu daftar inputan mata pelajaran yang masih diproses.



WELCOME! Hil Sonyawati Tjan

Sonyawati Tjan  
Online

Search...

MAIN NAVIGATION

- DASHBOARD
- SISWA
- MATA PELAJARAN

IMPORT FILE

IMPORT EXCEL (XLSX/XLS)

Choose File No file chosen

Submit

**Gambar 5.50** Tampilan input penilaian dari halaman proses mata pelajaran

DAFTAR NILAI SISWA

KELAS: 10-MM

VERIFIKASI NILAI

Show 10 entries

#	NISN	NAMA SISWA	NAMA MAPEL	TUGAS	ULANGAN HARIAN	UJIAN TENGAH SEMESTER	UJIAN AKHIR SEMESTER	NILAI AKHIR
1	0084930204	Tri Magdalena	Matematika	90   80   54   88   85   55   75   87	64   49   52   87   65   64   63   81	65	95	78.3
2	0027383342	Purwanti Asidik	Matematika	73   72   73   99   63   63   54   99	81   54   93   94   89   81   53   46	84	99	79.9
3	0082738949	Kirana Larasati	Matematika	68   96   67   93   53   47   71   99	67   85   69   47   83   85   88   51	77	58	66.14
4	0017377288	Wilson Alfonso	Matematika	94   52   92   75   97   84   45   71	97   45   62   64   58   60   75   94	66	47	73.2
5	0093720083	Micky Lay	Matematika	69   81   83   57   78   82   71   52	56   50   46   100   47   57   69   94	98	48	80.21
6	0011777288	Borman Napitupulu	Matematika	71   55   49   76   54   68   88   78	96   68   98   74   81   45   91   51	58	97	80.06
7	0030019938	Robert Kristanto	Matematika	60   65   98   48   99   81   69   95	60   67   48   60   87   68   47   57	65	53	72.92
8	0045593923	Suharti Laurensia	Matematika	51   100   67   64   79   87   99   60	55   85   63   99   83   76   87   70	69	49	70.18
9	0088590020	Kornelius Saputra	Matematika	68   62   53   98   69   76   71   62	84   94   66   70   93   99   80   82	98	87	78.28
10	0069949274	Monica Clay	Matematika	64   51   78   74   73   86   81   100	79   92   48   67   65   74   79   81	69	94	71.43

Showing 1 to 10 of 19 entries

Previous 1 2 Next

Copyright © 2019 SISTEM MONITORING PRESTASI SISWA. TUGAS AKHIR - IF - P.K. SISTEM INFORMASI. UBAYA.

DBKZ 16071603

**Gambar 5.51** Tampilan daftar nilai dari halaman proses mata pelajaran

Halaman ini hanya akan menampilkan daftar nilai yang belum terverifikasi. Jika guru/wali kelas sudah memverifikasi nilai maka halaman proses hanya akan ada satu bagian yaitu bagian input nilai. Selain kedua gambar diatas, modul mata pelajaran guru juga memiliki satu halaman ubah mata pelajaran yang dapat dilihat

pada gambar 4.52. Gambar dibawah ini hanya berupa potongan dari tampilan ubah nilai sebelum di verifikasi.



WELCOME! Hil Sonyawati Tjan

Sonyawati Tjan  
Online

Search...

MAIN NAVIGATION

- DASHBOARD
- SISWA
- MATA PELAJARAN
- EKSTRAKURIKULER
- PENGHARGAAN
- PELANGGARAN
- ABSEN

UBAH DATA NILAI SISWA

Mata Pelajaran : Matematika  
Nama Siswa : Tri Magdalena

Nilai Tugas

90
80
54
88
85
55
75
87

Nilai PH

64
49
52

**Gambar 5.52 Tampilan halaman ubah data nilai**

- **Halaman proses modul ketidaktuntasan**

Terdapat 2 halaman proses untuk modul ketidaktuntasan yaitu halaman buat laporan ketidaktuntasan dan ubah laporan ketidaktuntasan. Kedua halaman tersebut berupa *form* dimana guru/wali kelas wajib isi yaitu tanggal, nama siswa, nama pelanggaran yang sudah di set secara *default* sebagai ketidaktuntasan dan tidak dapat diubah dan deskripsi. Halaman input ketidaktuntasan dapat dilihat pada gambar 5.53 sedangkan halaman ubahnya dapat dilihat pada gambar 5.54

**WELCOME!** Hil Nyal Sooman

**BUAT LAPORAN KETIDAKTUNTASAN SISWA**

Tanggal: dd/mm/yyyy

Nama Siswa: Mea Moyer

Nama Pelanggaran: TTS - Tidak tuntas dalam mata pelajaran

Deskripsi:

Keluar Submit

Copyright © 2019 SISTEM MONITORING PRESTASI SISWA. TUGAS AKHIR - IF - P.K. SISTEM INFORMASI UBAYA. DBKZ 16071603

**Gambar 5.53 Tampilan halaman buat laporan ketidaktuntasan**

**WELCOME!** Hil Nyal Sooman

**UBAH LAPORAN KETIDAKTUNTASAN SISWA**

Tanggal: 19/02/2020

Nama Siswa: Ammara Hatfield

Nama Pelanggaran: TTS - Tidak tuntas dalam mata pelajaran

Deskripsi: Mengerjakan PR tidak tuntas

Ubah

Copyright © 2019 SISTEM MONITORING PRESTASI SISWA. TUGAS AKHIR - IF - P.K. SISTEM INFORMASI UBAYA. DBKZ 16071603

**Gambar 5.54 Tampilan halaman ubah laporan ketidaktuntasan**

- **Halaman input nilai ekstrakurikuler**

Halaman input nilai ekstrakurikuler adalah halaman untuk melakukan pencatatan terhadap nilai ekstrakurikuler seorang siswa. Halaman ini berupa *form* berupa *field* nama ekstrakurikuler, nama siswa, nilai dan deskripsi. Tampilan halaman input nilai ekstrakurikuler dapat dilihat pada gambar 5.55.

**Gambar 5.55 Tampilan halaman input nilai ekstrakurikuler**

- **Halaman proses modul penghargaan**

Terdapat 2 halaman proses untuk modul penghargaan yaitu halaman buat catatan penghargaan dan ubah catatan penghargaan. Kedua halaman tersebut berupa *form* dimana guru/wali kelas wajib isi yaitu tanggal, nama siswa, nama penghargaan dan deskripsi. Halaman input penghargaan dapat dilihat pada gambar 5.56 sedangkan halaman ubah catatan penghargaan dapat dilihat pada gambar 5.57.

**Gambar 5.56 Tampilan halaman input catatan penghargaan siswa**

**Gambar 5.57 Tampilan halaman ubah catatan penghargaan siswa**

- **Halaman proses modul pelanggaran**

Terdapat 2 halaman proses untuk modul pelanggaran yaitu halaman buat catatan pelanggaran dan ubah catatan pelanggaran. Kedua halaman tersebut berupa *form* dimana guru/wali kelas wajib isi yaitu tanggal, nama siswa, nama pelanggaran deskripsi, dan hukuman. Halaman input pelanggaran dapat dilihat pada gambar 5.58 sedangkan halaman ubah catatan pelanggaran dapat dilihat pada gambar 5.59.

**Gambar 5.58 Tampilan halaman input catatan pelanggaran siswa**

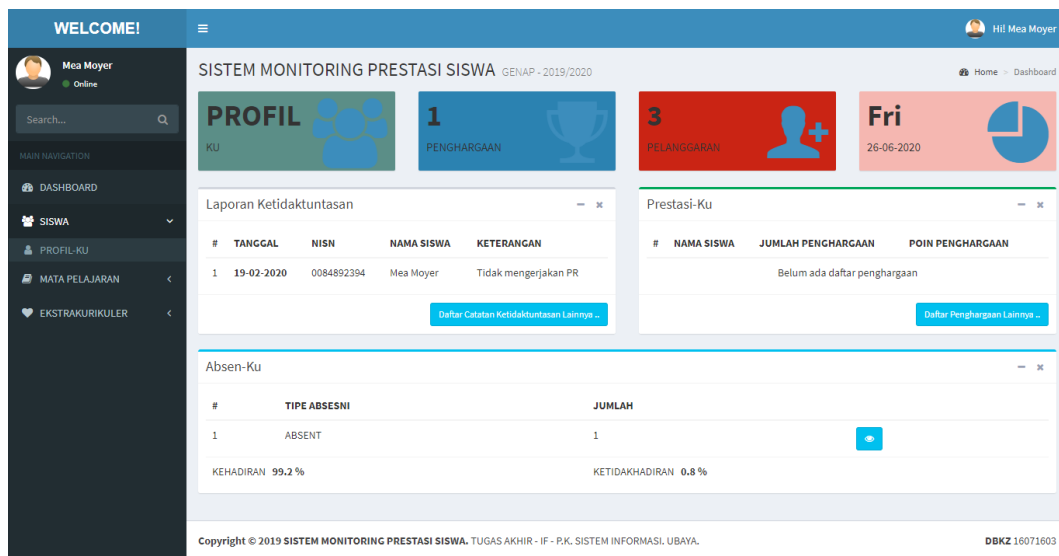
**Gambar 5.59 Tampilan halaman ubah catatan pelanggaran siswa**

### 5.3.3 Tampilan User Siswa

Secara umum, tampilan untuk *user role* siswa adalah bagian dari keseluruhan halaman yang dimiliki kepala sekolah dan guru/wali kelas. Tampilan halaman yang dimiliki siswa hanyalah halaman *dashboard*, halaman profil, halaman daftar mata pelajaran dan detail mata pelajaran, halaman ketidaktuntasan dan daftar ekstrakurikuler.

- **Halaman Dashboard Siswa**

Halaman *dashboard* yang dimiliki siswa adalah *highlights*/cuplikan setiap data yang tercatat dalam sistem. Pada halaman ini diberikan informasi jumlah profil, jumlah total penghargaan siswa, jumlah total pelanggaran siswa. Kemudian terdapat 3 kolom lagi yaitu informasi daftar laporan ketidaktuntasan, informasi daftar prestasi siswa, dan informasi daftar absen siswa. Gambar 5.60 menunjukkan tampilan halaman *dashboard* bagi kepala sekolah.



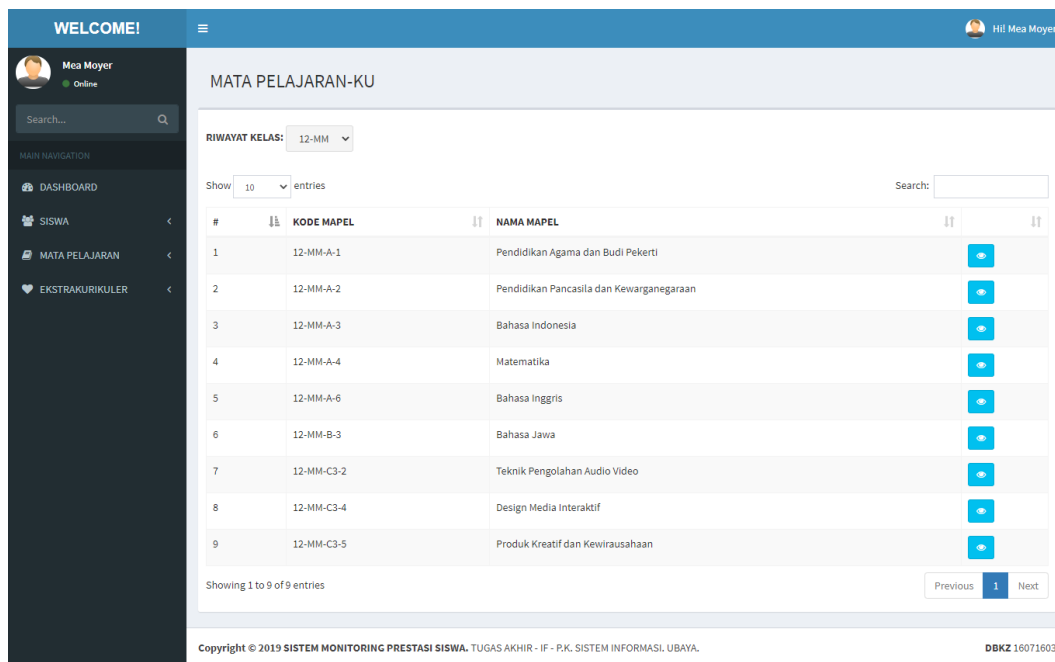
**Gambar 5.60 Tampilan halaman dashboard siswa**

- **Halaman Profil Siswa**

Halaman profil siswa yang adalah halaman yang menampilkan informasi mengenai catatan penghargaan, catatan pelanggaran, dan catatan absen siswa yang merupakan halaman untuk memantau nilai karakter siswa. Halaman ini sama seperti yang sudah dijelaskan pada tampilan halaman untuk kepala sekolah. Hanya saja data yang ditampilkan miliki siswa itu sendiri. Tampilan halaman profil siswa sudah digambarkan sebelumnya yaitu tampilan halaman profil siswa pada gambar 5.31.

- **Halaman Mata Pelajaran-Ku**

Halaman daftar mata pelajaran guru dapat dilihat pada gambar 5.61 yang memberi informasi mata pelajaran yang dimiliki siswa. Halaman daftar mata pelajaran-ku berisi informasi kode mapel dan nama mapel serta 1 tombol aksi yang diberikan dalam bentuk tabel.

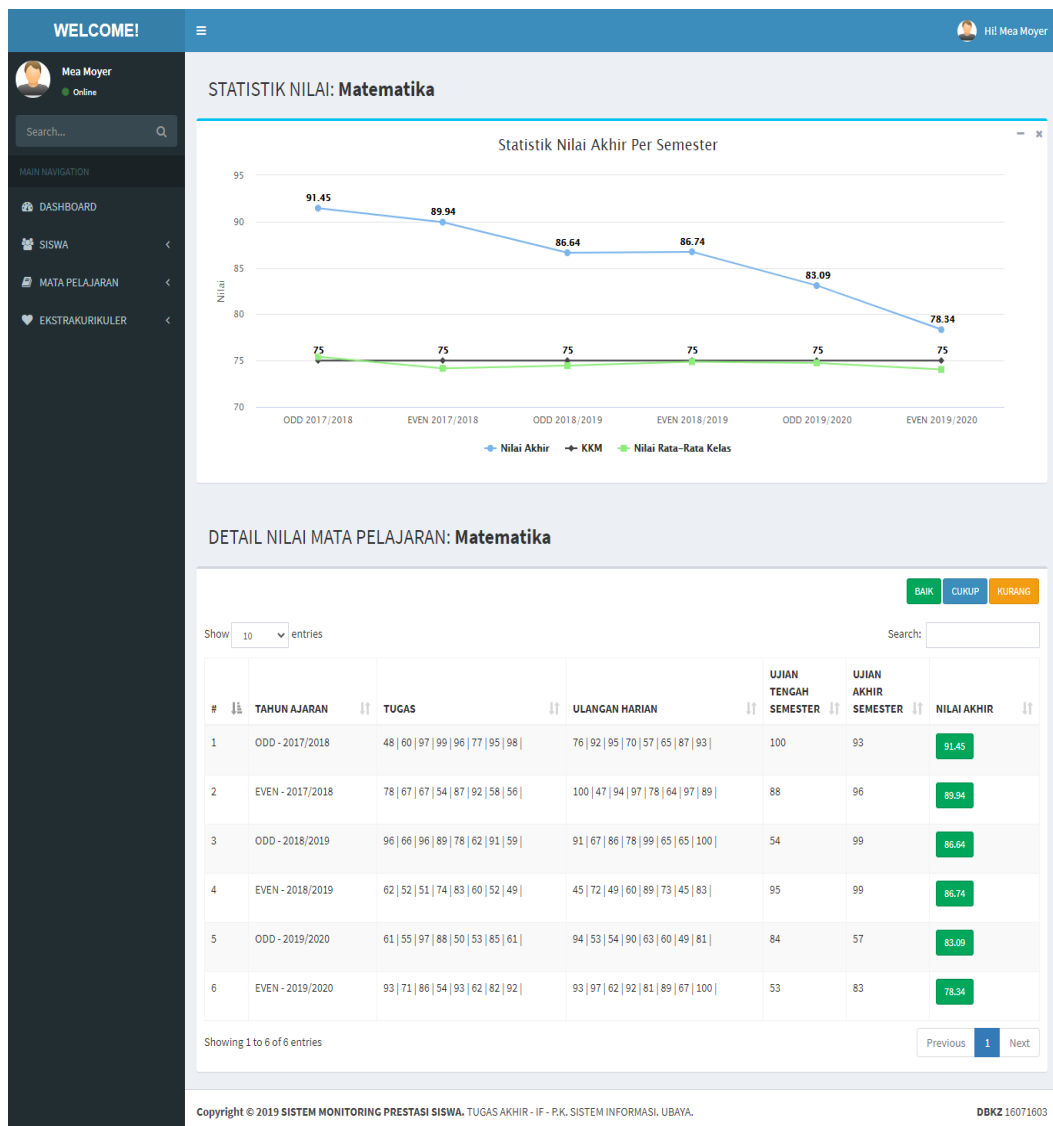


**Gambar 5.61 Tampilan halaman daftar mata pelajaran-ku**

- **Halaman Detail Mata Pelajaran-Ku**

Halaman detail mata pelajaran siswa dapat dilihat pada gambar 5.62. Sedangkan halaman detail mata pelajaran guru berisi informasi detail rincian nilai mata pelajaran siswa untuk mapel tertentu dan grafik sebagai statistic nilai akhir. Grafik memberikan informasi dengan 3 indikator yaitu data nilai rata-rata kelas, data nilai akhir dan nilai ketuntasan minimal.





**Gambar 5.62 Tampilan halaman daftar mata pelajaran-ku**

- **Halaman Daftar Catatan Ekskul-Ku**

Halaman daftar catatan ekskul-ku adalah *report* nilai dan kegiatan ekskul yang dimiliki siswa. Halaman ini memberikan informasi, nama siswa, nama ekskul, tahun ajaran dan nilainya. Halaman daftar catatan ekskul-ku dapat dilihat pada gambar 5.63.

#	NAMA SISWA	EKSUL	TAHUN AJARAN	NILAI
1	Mea Moyer	Pramuka	EVEN - 2019/2020	90

Showing 1 to 1 of 1 entries

Previous 1 Next

Copyright © 2019 SISTEM MONITORING PRESTASI SISWA. TUGAS AKHIR - IF - P.K. SISTEM INFORMASI UBAYA. DBKZ 16071603

**Gambar 5.63 Tampilan halaman catatan ekstrakurikuler**

Demikian penjelasan detail mengenai implementasi tampilan yang berada pada sistem *monitoring* prestasi siswa. Implementasi tampilan dijelaskan berdasarkan *role user* yang memiliki batasan-batasan halaman berdasarkan hak aksesnya. Tampilan yang dimiliki oleh guru/wali kelas dan kepala sekolah adalah melingkupi semua fitur yang ada pada sistem, sedangkan tampilan halaman yang dimiliki oleh siswa terbatas. Implementasi tampilan juga didampingi dengan adanya proses-proses. Proses-proses tersebut akan dibahas pada sub berikutnya yaitu implementasi program.

### 5.3 IMPLEMENTASI PROGRAM

Subbab implementasi program akan menjelaskan potongan kode untuk fungsi utama pada sistem. Tampilan sistem ini menggunakan *framework* yang berbasis HTML dan CSS. Pada bagian pengolahan data, sistem ini menggunakan bantuan *framework* PHP yaitu Laravel untuk memudahkan dalam membuat, mengubah, dan memproses data. Demikian juga pada implementasi program diperlukan potongan kode JQuery dan Javascript dalam menerima data.

### 5.3.1 Halaman Dashboard

Halaman *dashboard* adalah halaman utama ketika *user* masuk ke dalam sistem setelah melakukan login. Terdapat perbedaan data yang ditampilkan pada halaman ini antara *role* kepala sekolah, guru/wali kelas, dan siswa.

Kepala sekolah akan mendapatkan informasi mengenai keseluruhan data yang tercatat di database seperti total jumlah siswa, total jumlah penghargaan, total pelanggaran, daftar siswa bermasalah, daftar laporan ketidaktuntasan, daftar prestasi siswa, dan daftar absen siswa. Sementara untuk guru/wali kelas akan mendapatkan informasi data-data siswa yang dimiliki oleh kelasnya saja. Terakhir, untuk siswa hanya akan mendapatkan informasi mengenai datanya sendiri yang tercatat dalam *database*. Pada halaman ini terjadi banyak proses pengolahan data, seperti dibawah ini.

#### 1. Daftar Siswa Bermasalah

Proses pemberian informasi daftar siswa bermasalah kepada kepala sekolah dapat dilihat pada potongan baris program listing 5.1. Pengolahan data untuk menampilkan siswa bermasalah dijalankan dengan *function* “getTroubleStudent”. Fungsi ini akan menghitung banyaknya jumlah pelanggaran tiap-tiap siswa, dan menjumlahkan poin pelanggaran. Hasil dari fungsi ini akan dilemparkan ke halaman *dashboard* oleh fungsi index oleh *controller dashboard*. Demikian pula untuk role guru/wali kelas, baris program dapat dilihat pada listing 5.2.

**Listing 5.1 Baris program daftar siswa bermasalah role kepala sekolah**

```

public function getTroubleStudent($idUser, $selected_student){
    $siswa = ViolationRecord::join('students', 'violation_records.ST
UDENTS_ID', 'students.id')
        ->select('students.*',
                DB::raw('COUNT(*) as BANYAKPELANGGARAN'),
                DB::raw('SUM(violation_records.TOTAL) as TOTALP
OIN'))
        ->groupBy('students.id')
        ->orderBy('violation_records.id', 'DESC')->take(5)
        ->having('TOTALPOIN', ">=", 50 )
        ->get(); }

```

**Listing 5.2 Baris program daftar siswa bermasalah role guru/wali kelas**

```

public function getTroubleStudent($idUser, $selected_student)
{
    $kelas_guru = Grade::where('STAFFS_ID', $idUser)->first()
        ->id;
    $siswa = GradeStudent::where('GRADES_ID', $kelas_guru)
        ->where('ACADEMIC_YEAR_ID', $selected_student)
        ->get();
    $arr_siswa = [];
    foreach($siswa as $s){
        array_push($arr_siswa, $s->STUDENTS_ID);
    }
    $siswa = ViolationRecord::join('students', 'violation_records.
STUDENTS_ID', 'students.id')
        ->select('students.*',
                DB::raw('COUNT(*) as BANYAKPELANGGARAN'),
                DB::raw('SUM(violation_records.TOTAL) as TOTALPOIN'))
        ->whereIn('violation_records.STUDENTS_ID', $arr_siswa)
        ->groupBy('students.id')
        ->orderBy('violation_records.id', 'DESC')->take(5)
        ->having('TOTALPOIN', ">=", 50 )
        ->get();
}

```

**2. Daftar Laporan Ketidaktuntasan**

Pada halaman *dashboard* akan diberikan informasi mengenai laporan ketidaktuntasan. Daftar laporan ini akan ditampilkan dengan menjalankan fungsi “showStudentIncompleteness” dan “myIncompleteness” yang membutuhkan parameter *id user*. Fungsi ini akan berjalan apabila di dalam tabel *database* terdapat

daftar ketidaktuntasan dan akan membatasi jumlah daftar yang ditampilkan pada halaman ini. Fungsi ini dapat dilihat pada listing 5.3 dan 5.4 untuk fungsi bagi *role* siswa.

**Listing 5.3 Baris program daftar laporan ketidaktuntasan**

```
public function showStudentIncompleteness($idUser, $selected_student)
{
    if (Auth::guard('web')->user()->staff->ROLE == "TEACHER") {
        $kelas_guru = Grade::where('STAFFS_ID', $idUser)->first()->id;
        $siswa = GradeStudent::where('GRADES_ID', $kelas_guru)
            ->where('ACADEMIC_YEAR_ID', $selected_student)
            ->get();

        $arr_siswa = [];
        foreach($siswa as $s){
            array_push($arr_siswa, $s->STUDENTS_ID);
        }

        $ketidaktuntasan = ViolationRecord::join('violations',
            'violation_records.VIOLATIONS_ID', '=', 'violations.id')
            ->select('violation_records.*')
            ->where('violations.NAME', 'TTS')
            ->whereIn('violation_records.STUDENTS_ID', $arr_siswa)
            ->orderBy('violation_records.id', 'DESC')->take(5)
            ->get();
    }
    elseif (Auth::guard('web')->user()->staff
        ->ROLE == "HEADMASTER")
    {
        $ketidaktuntasan = ViolationRecord::join('violations',
            'violation_records.VIOLATIONS_ID', '=', 'violations.id')
            ->select('violation_records.*')
            ->where('violations.NAME', 'TTS')
            ->orderBy('violation_records.id', 'DESC')->take(5)
            ->get();
    }
    return $ketidaktuntasan;
}
```

**Listing 5.4 Baris program daftar laporan ketidaktuntasan role siswa**

```
public function myIncompleteness($idsiswa)
{
    $academic_year_id = AcademicYear::select(DB::raw('MAX(id) as
        id'))->get()[0]->id;
```

**Listing 5.4 Baris program daftar laporan ketidaktuntasan role siswa (lanjutan)**

```
$ketidaktuntasan = ViolationRecord::join('violations',  
'violation_records.VIOLATIONS_ID','=', 'violations.id')  
->select('violation_records.*')  
->where('violations.NAME', 'TTS')  
->where('violation_records.STUDENTS_ID', $idsiswa)  
->  
>where('violation_records.ACADEMIC_YEAR_ID', $academic_year_id)  
->orderBy('violation_records.id', 'DESC')->take(5)  
->get();  
  
return $ketidaktuntasan;  
}
```

**3. Daftar Prestasi Siswa**

Halaman *dashboard* memberikan informasi mengenai prestasi/penghargaan yang dimiliki siswa. Daftar prestasi yang ditampilkan berupa nama siswa beserta dengan jumlah penghargaan yang tercatat dalam *database* dengan menjalankan fungsi “studentAchievements” dan “myAchievement” untuk *role* siswa. Potongan baris program fungsi daftar prestasi siswa dapat dilihat pada listing 5.5 dan 5.6.

**Listing 5.5 Baris program daftar prestasi siswa**

```
public function studentAchievements($iduser)  
{  
    if(Auth::guard('web')->user()->staff->ROLE == "TEACHER")  
    {  
        $selected_student = GradeStudent::select(DB::raw('MAX(ACADEMIC_YEAR_ID) AS id'))->limit(1)->first()->id;  
  
        $kelas_guru = Grade::where('STAFFS_ID', $iduser)->first()->id;  
  
        $siswa = GradeStudent::where('GRADES_ID', $kelas_guru)  
        ->where('ACADEMIC_YEAR_ID', $selected_student)->get();  
  
        $arr_siswa = [];  
        foreach($siswa as $s){  
            array_push($arr_siswa, $s->STUDENTS_ID);  
        }  
    }  
}
```

**Listing 5.5 Baris program daftar prestasi siswa (lanjutan)**

```

    $penghargaan_siswa = AchievementRecord::join('achievements',
    'achievement_records.ACHIEVEMENTS_ID', 'achievements.id')
    ->select('achievement_records.*',
            DB::raw('COUNT(*) AS BANYAKPENGHARGAAN'),
            DB::raw('SUM(POINT) AS POINPENGHARGAAN'))
    ->whereIn('achievement_records.STUDENTS_ID', $arr_siswa)
    ->groupBy('achievement_records.STUDENTS_ID')
    ->orderBy('achievement_records.id', 'DESC')->take(5)
    ->get();
}
elseif(Auth::guard('web')->user()->staff->ROLE == "HEADMASTER")
{
    $penghargaan_siswa = AchievementRecord::join('achievements',
    'achievement_records.ACHIEVEMENTS_ID', 'achievements.id')
    ->select('achievement_records.*',
            DB::raw('COUNT(*) AS BANYAKPENGHARGAAN'),
            DB::raw('SUM(POINT) AS POINPENGHARGAAN'))
    ->groupBy('achievement_records.STUDENTS_ID')
    ->orderBy('achievement_records.id', 'DESC')->take(5)
    ->get();
}
return $penghargaan_siswa;
}

```

**Listing 5.6 Baris program daftar prestasi siswa role siswa**

```

public function myAchievement($idsiswa)
{
    $academic_year_id = AcademicYear::select(
        DB::raw('MAX(id) as id'))->get()[0]->id;

    $penghargaan_siswa = AchievementRecord::join('achievements',
    'achievement_records.ACHIEVEMENTS_ID', 'achievements.id')
    ->select('achievement_records.*',
            DB::raw('COUNT(*) AS BANYAKPENGHARGAAN'),
            DB::raw('SUM(POINT) AS POINPENGHARGAAN'))
    ->where('achievement_records.STUDENTS_ID', $idsiswa)
    ->where('achievement_records.ACADEMIC_YEAR_ID',
        $academic_year_id)
    ->groupBy('achievement_records.STUDENTS_ID')
    ->orderBy('achievement_records.id', 'DESC')->take(5)
    ->get();

    return $penghargaan_siswa;
}

```

#### 4. Daftar Absen

User role kepala sekolah, wali kelas, dan siswa akan mendapatkan informasi tentang daftar absen. Daftar absen ini menjalankan perintah fungsi yang diatur pada *controller dashboard* yaitu “showAbsent” untuk *role* kepala sekolah dan guru/wali kelas, sedangkan fungsi “showAbsentKu” untuk *role* siswa. Potongan baris program fungsi yang menampilkan daftar absen dapat dilihat pada listing 5.7.

##### Listing 5.7 Baris program daftar absen

```
public function showAbsent($iduser)
{
    $academic_year_id = AcademicYear::select(DB::raw('MAX(id) as
    Id'))->get()[0]->id;

    $count_total_day_each_ay = AcademicYear::select(
    DB::raw('DATEDIFF(END_DATE, START_DATE) AS TOTALHARI'))
        ->where('id', $academic_year_id)->first()->TOTALHARI;

    if(Auth::guard('web')->user()->staff->ROLE == "TEACHER")
    {
        $type = Absent::select(DB::raw('TYPE AS TIPE'))
            ->groupBy('TYPE')->get();

        $selected_student = GradeStudent::select(DB::raw('MAX(
        ACADEMIC_YEAR_ID) AS id'))->limit(1)->first()->id;

        $kelas_guru = Grade::where('STAFFS_ID', $iduser)
            ->first()->id;

        $siswa = GradeStudent::join('students', 'grades_students
            .STUDENTS_ID', 'students.id')
            ->select('students.id')
            ->where('grades_students.GRADES_ID', $kelas_guru)
            ->where('grades_students.ACADEMIC_YEAR_ID',
                $selected_student)
            ->get();

        $arr_siswa = [];
        foreach($siswa as $s){
            array_push($arr_siswa, $s->id);
        }
    }
}
```



**Listing 5.7 Baris program daftar absen (lanjutan)**

```
$data = Absent::select(DB::raw('TYPE AS TIPE'),
    DB::raw('ACADEMIC_YEAR_ID AS TAHUNAJARAN'),
    DB::raw('COUNT(*) AS JUMLAH')
    ->where('ACADEMIC_YEAR_ID', $academic_year_id)
    ->whereIn('STUDENTS_ID', $arr_siswa)
    ->groupBy('TIPE', 'TAHUNAJARAN')
    ->get();

$absen = Absent::where('ACADEMIC_YEAR_ID',
    $academic_year_id)
    ->whereIn('STUDENTS_ID', $arr_siswa)->get();
}
elseif(Auth::guard('web')->user()->staff->ROLE ===
"HEADMASTER")
{
    $data = Absent::select(DB::raw('TYPE AS TIPE'),
        DB::raw('ACADEMIC_YEAR_ID AS TAHUNAJARAN'),
        DB::raw('COUNT(*) AS JUMLAH'))
        ->where('ACADEMIC_YEAR_ID',
            $academic_year_id)
        ->groupBy('TIPE', 'TAHUNAJARAN')
        ->get();
}

$value["data"] = $data;

return $value;
}
```

**5.3.2 Daftar Siswa**

Daftar siswa merupakan halaman pada hak akses untuk kepala sekolah dan guru/wali kelas yang menampilkan daftar siswa yang terdaftar di sekolah. Jika *user* login sebagai kepala sekolah maka dapat melihat keseluruhan daftar siswa di sekolah. Jika *user* login sebagai guru/wali kelas dapat melihat daftar siswa kelasnya saja. Baris program daftar siswa dapat dilihat pada listing 5.8.

**Listing 5.8 Baris program function daftar siswa**

```

public function index(Request $request)
{
    $kelas = Grade::all();
    $min_grade_id = Grade::select(DB::raw('MIN(id) as id'))
    ->get()[0]->id;
    if($request->has('gradeId')){
        $grade_id = $request->gradeId;
    }
    else{
        $grade_id = $min_grade_id;
    }
    $selected_student = GradeStudent::select(
        DB::raw('MAX(ACADEMIC_YEAR_ID) AS id'))
    ->limit(1)->first()->id;
    if(Auth::guard('web')->user()->staff->ROLE === "TEACHER"){
        $kelas_guru = Grade::where('STAFFS_ID',
            $request->session()->get('session_user_id'))
        ->first()->id;
        $siswa = GradeStudent::where('GRADES_ID',$kelas_guru)
        ->where('ACADEMIC_YEAR_ID',
            $selected_student)->get();
    }
    else{
        $siswa = GradeStudent::where('GRADES_ID', $grade_id)
        ->where('ACADEMIC_YEAR_ID', $selected_student)->get();
    }
    return view('student.index', compact('kelas', 'siswa',
        'grade_id'));
}

```

**5.3.3 Input Daftar Siswa**

Untuk dapat melakukan input daftar siswa hanya bisa dilakukan oleh user dengan *role* sebagai teacher atau wali kelas. Fitur input daftar siswa terdapat di halaman daftar siswa dengan menekan tombol dibagian *header* table. Proses menginputkan data siswa dilakukan pada *function* “importStudent” pada import *controller* yang kemudian melemparkan data ke *function* berupa collection di dalam *student import collection* tersendiri. Listing 5.9 menunjukkan baris program import *controller*, sedangkan listing 5.10 menunjukkan baris program *function* collection.

**Listing 5.9 Baris program melempar data ke function collection**

```
public function importStudent(Request $request)
{
    $this->validate($request, [
        'student_import' => 'required|mimes:csv,xls,xlsx'
    ]);

    Excel::import(new StudentImport($request), $request-
    >file('student_import'));
    Session::flash('sukses', 'Import data siswa berhasil
    ditambahkan');
    return redirect('student');
}
```

**Listing 5.10 Baris program function collection input data ke database**

```
public function collection(Collection $collection)
{
    foreach($collection as $key => $row)
    {
        if($key >= 3)
        {
            $grade_id = $row[4];
            $tmp_id_grade = $this->selectGradeId($grade_id);

            foreach($tmp_id_grade as $ig)
            {
                $id_kelas = $ig->id;
            }

            if($key >= 5)
            {
                $student_full_name = $row[5];
                $studentName = explode(" ", $row[5], 2);

                $siswa = new Student();
                $siswa->NISN = $row[4];
                $siswa->FNAME = $studentName[0];
                $siswa->LNAME = $studentName[1];
                $siswa->ACADEMIC_YEAR_ID = $this->request
                ->session()
                ->get("session_academic_year_id");

                $siswa->save();
                $kelas_siswa = GradeStudent::create([
                    'STUDENTS_ID' => $siswa->id,
                    'GRADES_ID' => $id_kelas,
                    'ACADEMIC_YEAR_ID' => $this->request
                    ->session()->get("session_academic_year_id")
                ]);
            }
        }
    }
}
```

**Listing 5.10** Baris program function collection input data ke database (lanjutan)

```

        $user = User::create([
            'name' => $student_full_name,
            'email' => "s".$row[4]."@student.ac.id",
            'password' =>
                Hash::make($studentName[0]),
            'ROLE' => 'STUDENT',
            'STUDENTS_ID' => $siswa->id
        ]);
    }
}
}}
```

### 5.3.4 Halaman Profil Siswa

Halaman profile siswa bisa diakses dari *button* yang terdapat di kolom tabel aksi tabel daftar siswa. Pada halaman profile, user bisa melihat informasi data yang menjadi tolak ukur dalam sistem ini yaitu catatan pelanggaran, catatan penghargaan, dan catatan absen masing-masing siswa. Untuk menampilkan halaman ini, data diproses pada *controller* siswa menggunakan *function* “profile”. Baris program halaman siswa dapat dilihat pada listing 5.11.

**Listing 5.11** Baris program function profile

```

public function profile(Request $request, $id)
{
    $siswa = Student::find($id);
    $student_grade_name = $siswa->getGradeName();
    $siswaku = Student::where('STUDENTS_ID', $id);
    $absen = Absent::where('STUDENTS_ID', $id)
        ->orderBy('START_DATE', 'END_DATE', 'ASC')
        ->get();
    $catatan_absen = Absent::select('TYPE', DB::raw('COUNT(TYPE)
        AS TOTAL'))
        ->where('STUDENTS_ID', $id)
        ->where('ACADEMIC_YEAR_ID',
            $request->session()->get('session_academic_year_id'))
        ->groupBy('TYPE')
        ->get();
}
```

**Listing 5.11 Baris program function profile (lanjutan)**

```
$x = new DashboardController();
$kehadiran = $x->showAbsentKu($siswa->id);

$selectd_student_ay = Student::select('ACADEMIC_YEAR_ID AS
AY_ID')->where('id', $siswa->id) ->first()
->AY_ID;

$tahun_ajaran = AcademicYear::where('id', '>=', $selectd_stu
dent_ay)->get();

$catatan_pelanggaran = ViolationRecord::join('violations',
'violation_records.VIOLATIONS_ID', 'violations.id')
->select('violation_records.*', 'violations.*')
->where('STUDENTS_ID', '=', $id)
->where('ACADEMIC_YEAR_ID', '=', $request->session()
->get('session_academic_year_id'))
->where('violations.NAME', 'NOT LIKE', 'TTS%')
->get();

$violation_point = ViolationRecord::select(DB::raw('SUM(TOTAL
) AS POINT'))
->where('STUDENTS_ID', $id)
->first()->POINT;

$sachievement_point = AchievementRecord::join('achievements',
'achievement_records.ACHIEVEMENTS_ID', 'achievements.id')
->select(DB::raw('SUM(POINT) AS POINT'))
->where('achievement_records.STUDENTS_ID', $id)
->first()->POINT;

$catatan_penghargaan = AchievementRecord::join('achievements',
'achievement_records.ACHIEVEMENTS_ID', 'achievements.id')
->select('achievements.*', 'achievement_records.*')
->where('STUDENTS_ID', '=', $id)
->where('ACADEMIC_YEAR_ID', '=', $request->session()
->get('session_academic_year_id'))
->get();

$maxId = AcademicYear::select(DB::raw('MAX(id) as id'))
->get()[0]->id;
if($request->has('academicYearId')){
    $academic_year_id = $request->academicYearId;
}
else{
    $academic_year_id = $maxId;
}

$selectd_tahun_ajaran = AcademicYear::select(DB::raw('MONTH(
START_DATE) AS STARTMONTH'),
DB::raw('MONTH(END_DATE) AS ENDMONTH'))
->where('id', $academic_year_id)->get()[0];
```

**Listing 5.11 Baris program function profile (lanjutan)**

```

// GRAFIK TAB VIOLATION
$kategori = DB::select("SELECT (CASE
    WHEN v.NAME LIKE 'R%' THEN 'RINGAN'
    WHEN v.NAME LIKE 'B%' THEN 'BERAT'
    WHEN v.NAME LIKE 'SB%' THEN 'SANGATBERAT'
    END) AS KATEGORI
FROM violation_records vr INNER JOIN violations v ON vr.VIOLA
TIONS_ID = v.id
WHERE v.NAME NOT LIKE 'TTS%'
GROUP BY KATEGORI ");

$data = ViolationRecord::join('violations', 'violation_record
s.VIOLATIONS_ID', 'violations.id')
->select(DB::raw('(CASE
    WHEN violations.NAME LIKE "R%" THEN "RINGAN"
    WHEN violations.NAME LIKE "B%" THEN "BERAT"
    WHEN violations.NAME LIKE "SB%" THEN "SANGATBERAT"
END) AS KATEGORI'),
DB::raw('MONTH(violation_records.DATE) AS BULAN'),
DB::raw('COUNT(*) AS JUMLAH'))
-
where('violation_records.ACADEMIC_YEAR_ID', $academic_year_id
)
->where('violation_records.STUDENTS_ID', $id)
->where('violations.NAME', 'NOT LIKE', 'TTS%')
->groupBy('KATEGORI')->groupBy('BULAN')
->orderBy('violation_records.id', 'DESC')->get();

// GRAFIK TAB ACHIEVEMENT
$type = Achievement::select(DB::raw('GRADE AS TINGKAT'))
->groupBy('TINGKAT')
->get();

$dataAchievement = DB::select("SELECT a.GRADE AS TINGKAT, MON
TH (ass.DATE) AS BULAN , COUNT(*) AS JUMLAH
FROM achievements a INNER JOIN achievement_records ass ON a.
id = ass.ACHIEVEMENTS_ID
WHERE ACADEMIC_YEAR_ID = " . $academic_year_id . " AND STUDEN
TS_ID = " . $id . "
GROUP BY TINGKAT, BULAN
ORDER BY BULAN ASC");

// GRAFIK TAB ABSENT
$typeAbsen = Absent::select(DB::raw('TYPE AS TIPE'))
->groupBy('TIPE')
->get();

$dataAbsen = Absent::select(DB::raw('TYPE AS TIPE'), DB::raw(
'ACADEMIC_YEAR_ID AS TAHUNAJARAN'), DB::raw('COUNT(*) AS JUML
AH'))
->where('ACADEMIC_YEAR_ID', $academic_year_id)
->where('STUDENTS_ID', $id)

```

**Listing 5.11 Baris program function profile (lanjutan)**

```

->groupBy('TIPE', 'TAHUNAJARAN')
->get();

$count_total_day_each_ay = AcademicYear::select(DB::raw('DATE
DIFF(END_DATE, START_DATE) AS TOTALHARI'))
->where('id', $academic_year_id)->first()
->TOTALHARI;

// RETURN VIEW
if(Auth::guard('web')->user()->ROLE === "STAFF")
    return view('student.profile', compact('siswa', 'student_grad
    e_name', 'absen', 'catatan_absen', 'kehadiran', 'tahun_ajaran',
    'catatan_pelanggaran', 'violation_point',
    'catatan_penghargaan', 'achievement_point',
    'kategori', 'data', 'selected_tahun_ajaran', 'academic_year
    _id',
    'type', 'dataAchievement', 'tipeAbsen', 'dataAbsen', 'count_
    total_day_each_ay'));
else
    return view('student.profile', compact('siswa', 'absen', 'c
    atatan_absen', 'kehadiran', 'tahun_ajaran',
    'catatan_pelanggaran', 'violation_point',
    'catatan_penghargaan', 'achievement_point',
    'kategori', 'data', 'selected_tahun_ajaran', 'academic_year
    _id', 'type', 'dataAchievement',
    'tipeAbsen', 'dataAbsen', 'count_total_day_each_ay'));

```

**5.3.5 Halaman Mapel-Guru**

Halaman mapel guru adalah salah satu sub-menu dari menu mata pelajaran.

Halaman ini bisa diakses hanya oleh *user* sebagai wali kelas. Untuk menampilkan halaman ini diperlukan session id user sebagai parameter dari *function* “mapelguru”. Fungsi ini memproses data user terlebih dahulu *user* berada pada departemen id berapa kemudian mencari id kelas yang diajar sehingga baru di proses untuk menemukan kode-kode mata pelajaran yang berada dalam kelas *user*.

Baris program daftar mapel-guru dapat dilihat pada listing 5.12.

**Listing 5.12 Baris program function mapelguru**

```

public function mapelguru(Request $request)
{
    $get_department_id = DepartmentStaff::select(
        'DEPARTMENTS_ID')
    ->where('STAFFS_ID', $request->session())
    ->get('session_user_id'))
    ->get();

    $get_grade_id = Grade::select('NAME')
    ->where('STAFFS_ID', $request->session())
    ->get('session_user_id'))
    ->get();

    $a = array($get_department_id, $get_grade_id);
    $tmp_subject = [];
    foreach($get_department_id as $deparment_id){
        foreach($get_grade_id as $grade_id){
            $selected_mapel = Subject::select('id', 'CODE'
                , 'DESCRIPTION')
            ->where('DEPARTMENTS_ID', $deparment_id->DEPARTMENTS_ID)
            ->where('CODE', 'LIKE', '%' . $grade_id->NAME . '%')

            ->get();

            array_push($tmp_subject, $selected_mapel);
        }
    }
    $kelas = Grade::all();
    $gname = Grade::select('NAME')
    ->where('id', Grade::select(DB::raw('MIN(id) as id'))
    ->first()->id)->get()[0]->NAME;

    $gid = Grade::select(DB::raw('MIN(id) as id'))->first()->id;
    if($request->has('gradeName')){
        $grade_name = $request->gradeName;
    }
    else{
        $grade_name = $gname;
    }
    if(Auth::guard('web')->user()->staff->ROLE === "TEACHER")
        $subject = $tmp_subject[0];
    elseif(Auth::guard('web')->user()->staff->ROLE
    === "HEADMASTER")
        $subject = Subject::where('CODE', 'LIKE', '%' .
        $grade_name . '%')->get();

    return view('student.mapel-guru', compact('subject', 'kelas'
        , 'grade_name', 'gid'));
}

```



### 5.3.5 Halaman Detail Mata Pelajaran

Halaman detail mata pelajaran adalah halaman dimana *user* bisa melihat rincian setiap nilai yang tercatat dalam *database*. Dalam halaman ini terdapat 2 buah kolom informasi yang diberikan kepada user yaitu kolom grafik yang sebagai informasi yang menunjukkan ada atau tidaknya kenaikan/menurunnya nilai siswa, dan tabel rinciannya. Data pada halaman ini diproses menggunakan *function* “subjectDetail” yang dibedakan cara memproses data untuk setiap *role*.

#### 1. Halaman detail mata pelajaran untuk *user role* wali kelas

Halaman detail mata pelajaran untuk *user role* wali kelas akan menampilkan data nilai seluruh siswanya pada tabel. Sedangkan informasi pada bagian grafik memberikan data nilai rata-rata kelas, jumlah siswa yang nilai akhirnya berada dibawah nilai ketuntasan minimal/KKM dan KKM itu sendiri. Baris program detail mata pelajaran pada tabel dapat dilihat pada listing 5.13. Sedangkan baris program data nilai dalam bentuk grafik dapat dilihat pada listing 5.14.

#### Listing 5.13 Baris program detail mata pelajaran pada tabel

```
public function subjectDetail(Request $request, $id)
{
    $mapel = Subject::find($id);
    $max_academic_year_id = AcademicYear::select(DB::raw(
        ('MAX(id) as id'))->get()[0]->id;

    if($request->has('academicYearId')){
        $academic_year_id = $request->academicYearId;
    }
    else{
        $academic_year_id = $max_academic_year_id;
    }

    $kkm = Subject::select('MINIMALPOIN')->where('id', $mapel
        ->id)->get();

    $x = explode("-", $mapel->CODE, 3);
    $y = $x[0] . "-" . $x[1];
```

**Listing 5.13 Baris program detail mata pelajaran pada tabel (lanjutan)**

```

$url = parse_url(url()->previous(), PHP_URL_QUERY);
$get_url_param = explode("=", $url);
$siswa_dibawah_rata = [];

if(Auth::guard('web')->user()->ROLE === "STAFF"){
    $select_grade_id = Grade::select('id')
    ->where('NAME', $y)->first()->id;

    $selected_student_in_ay = GradeStudent::select(
    DB::raw('MAX(ACADEMIC_YEAR_ID) AS id'))->limit(1)
    ->first()->id;

    $selected_student_ay = Student::select('students
    .ACADEMIC_YEAR_ID AS AY_ID')
    ->join('grades_students', 'students.id',
    'grades_students.STUDENTS_ID')

    ->where('grades_students.GRADES_ID', $select_grade_id)
    ->limit(1)->first()->AY_ID;

    $tahun_ajaran = AcademicYear::where('id', '>=',
    $selected_student_ay)->get();

    $detail_mapel = SubjectReport::join('subject_records',
    'subject_reports.SUBJECT_RECORD_ID',
    'subject_records.id')
    ->join('subjects', 'subject_reports.SUBJECTS_ID',
    'subjects.id')
    ->join('students', 'subject_records.STUDENTS_ID',
    'students.id')
    ->join('grades_students', 'students.id',
    'grades_students.STUDENTS_ID')
    ->select('subjects.*', 'subject_records.*',
    'subject_reports.*')
    ->where('subject_records.ACADEMIC_YEAR_ID',
    $academic_year_id)
    ->where('subject_reports.SUBJECTS_ID', $mapel->id)
    ->where('grades_students.GRADES_ID', '=',
    $select_grade_id)
    ->where('grades_students.ACADEMIC_YEAR_ID',
    $selected_student_in_ay)
    ->get();

    $rata_kelas = SubjectReport::join('subject_records',
    'subject_reports.SUBJECT_RECORD_ID',
    'subject_records.id')
    ->join('subjects', 'subject_reports.SUBJECTS_ID',
    'subjects.id')
    ->join('students', 'subject_records.STUDENTS_ID',
    'students.id')
    ->join('grades_students', 'students.id',
    'grades_students.STUDENTS_ID')

```

**Listing 5.13 Baris program detail mata pelajaran pada tabel (lanjutan)**

```

->select(DB::raw('ROUND(SUM(subject_reports.FINAL_SCORE)/
COUNT(subject_records.STUDENTS_ID), 2) AS RATAKELAS'),
'subject_records.*')
->where('subject_reports.SUBJECTS_ID', $mapel->id)
->where('grades_students.GRADES_ID', '=',
$select_grade_id)
->where('grades_students.ACADEMIC_YEAR_ID',
$select_student_in_ay)
->groupBy('subject_records.ACADEMIC_YEAR_ID')
->get();

foreach($rata_kelas as $rk)
{
    if(!empty($rk))
        $siswa_dibawah_rata = SubjectReport::join
            ('subject_records',
            'subject_reports.SUBJECT_RECORD_ID',
            'subject_records.id')
        ->join('students', 'subject_records.STUDENTS_ID',
            'students.id')
        ->join('subjects', 'subject_reports.SUBJECTS_ID',
            'subjects.id')
        ->join('grades_students', 'students.id',
            'grades_students.STUDENTS_ID')
        ->select('subject_records.*', DB::raw('COUNT(*)
            AS JUMLAH_SISWA_DIBAWAH_RATA'))
        ->where('subject_reports.SUBJECTS_ID', $mapel->id)
        ->where('subject_reports.FINAL_SCORE', '<',
            $rk->RATAKELAS)
        ->where('grades_students.GRADES_ID', '=',
            $select_grade_id)
        ->where('grades_students.ACADEMIC_YEAR_ID',
            $select_student_in_ay)
        ->groupBy('subject_records.ACADEMIC_YEAR_ID')
        ->get();
    else
        $siswa_dibawah_rata = [];
}
}

```

**Listing 5.4 Baris program data nilai dalam bentuk grafik**

```

var categories = {!! json_encode($tahun_ajaran) !!}
var kkm = {!! json_encode($kkm) !!}
var averageScorePerClass = {!! json_encode($rata_kelas) !!}
var badStudent = {!! json_encode($siswa_dibawah_rata) !!}
var category = []
var series1 = []
var series2 = []
var series3 = []

```

**Listing 5.14 Baris program data nilai dalam bentuk grafik (lanjutan)**

```

categories.forEach(function(item){
    badStudent.forEach(function(obj_x){
        if(obj_x.ACADEMIC_YEAR_ID == item.id){
            var catName = item.TYPE + " " + item.NAME
            category.push(catName)
            averageScorePerClass.forEach(function(obj_rata){
                if(obj_rata.ACADEMIC_YEAR_ID == item.id){
                    var tmp_rata = obj_rata.RATAKELAS
                    series3.push(tmp_rata)
                    kkm.forEach(function(obj_kkm){
                        var tmp_kkm = obj_kkm.MINIMALPOIN
                        series2.push(tmp_kkm)
                    })
                }
            })
            var tmp_jumlah = obj_x.JUMLAH_SISWA_DIBAWAH_RATA
            series1.push(tmp_jumlah)
        }
    })
})
Highcharts.chart('gradeOfSubjectChart', {
    chart: {
        type: 'line'
    },
    title: {
        text: 'Statistik Nilai Akhir Kelas Per Semester'
    },
    xAxis: {
        categories: category
    },
    yAxis: {
        title: {
            text: 'Nilai'
        }
    },
    plotOptions: {
        line: {
            dataLabels: {
                enabled: true
            },
            enableMouseTracking: true
        }
    },
    credits: {
        enabled: false
    },
    series: [{
        name: 'Jumlah siswa nilai akhir dibawah rata kelas',
        data: series1
    },

```

**Listing 5.14 Baris program data nilai dalam bentuk grafik (lanjutan)**

```
{
    name: 'KKM',
    data: series2
}, {
    name: 'Nilai rata-rata kelas',
    data: series3 }]]]);
```

**2. Filter tahun ajaran untuk tabel detail mata pelajaran**

Pada halaman detail mata pelajaran bagi wali kelas juga disediakan satu *combobox* yang berisikan daftar tahun ajaran. *Combobox* filter tahun ajaran ini berguna bagi wali kelas melihat/mencari detail-detail nilai mata pelajaran pada tahun ajaran tertentu. Adapun baris program filter tahun ajaran untuk tabel detail mata pelajaran dapat dilihat pada listing 5.15.

**Listing 5.15 Baris program filter tahun ajaran untuk tabel detail mata pelajaran**

```
$('#dropdown-detail-subject-academicyear').val(
  {{$academic_year_id}})
$('#dropdown-detail-subject-academic-year').change(function(){
  var academicYearId = $(this).val()
  var route = "{{ route('subject.detail', ':id') }}"
  route = route.replace(':id', '{{ $mapel->id }}')
  window.location = route+"?academicYearId="+academicYearId;
})
```

**3. Halaman detail mata pelajaran untuk *user role* siswa**

Halaman detail mata pelajaran untuk *user role* siswa akan menampilkan data nilai siswa itu sendiri. Rincian nilai setiap tahun ajaran permata pelajaran akan ditampilkan pada bagian grafik maupun tabel. Baris program detail mata pelajaran

pada tabel dapat dilihat pada listing 5.16. Sedangkan baris program data nilai dalam bentuk grafik dapat dilihat pada listing 5.14 sama seperti untuk bagian wali kelas.

**Listing 5.16 Baris program detail nilai mata pelajaran siswa**

```
public function subjectDetail(Request $request, $id)
{
    $siswa = Student::find($request->session()
                           ->get('session_student_id'));
    $selected_student_ay = Student::select('ACADEMIC_YEAR_ID AS
                                         AY_ID')
    ->where('id', $request->session()->get('session_student_id'))
    ->first()->AY_ID;

    $tahun_ajaran = AcademicYear::where('id', '>=',
                                         $selected_student_ay)->get();

    $detail_mapel_ku = SubjectReport::join('subject_records',
                                           'subject_reports.SUBJECT_RECORD_ID',
                                           'subject_records.id')
    ->join('subjects', 'subject_reports.SUBJECTS_ID',
          'subjects.id')
    ->select('subjects.*', 'subject_records.*',
            'subject_reports.*')
    ->where('subjects.DESCRPTION', $mapel->DESCRIPTION)
    ->where('subject_records.ACADEMIC_YEAR_ID', $academic_year_id)
    ->where('subject_records.STUDENTS_ID', $request->session()
        ->get('session_student_id'))
    ->get();

    $data_final_score = SubjectReport::join('subject_records',
                                           'subject_reports.SUBJECT_RECORD_ID', 'subject_records.id')
    ->join('subjects', 'subject_reports.SUBJECTS_ID',
          'subjects.id')
    ->select('subject_records.*', 'subject_reports.*')
    ->where('subjects.DESCRPTION', $mapel-
    >DESCRIPTION)
    ->where('subject_records.STUDENTS_ID', $request->session()-
    >get('session_student_id'))->get();

    if(isset($url)){
        $select_grade_id = Grade::select('id')
        ->where('NAME', $get_url_param[1])->first()->id;
        $selected_student_in_ay = GradeStudent::select
            ('ACADEMIC_YEAR_ID')
        ->where('STUDENTS_ID', $request->session()
            ->get('session_student_id'))
        ->where('GRADES_ID', $select_grade_id)
        ->first()->ACADEMIC_YEAR_ID;
    }
}
```

pada tabel dapat dilihat pada listing 5.16. Sedangkan baris program data nilai dalam bentuk grafik dapat dilihat pada listing 5.14 sama seperti untuk bagian wali kelas.

**Listing 5.16 Baris program detail nilai mata pelajaran siswa**

```
public function subjectDetail(Request $request, $id)
{
    $siswa = Student::find($request->session()
        ->get('session_student_id'));
    $selected_student_ay = Student::select('ACADEMIC_YEAR_ID AS
        AY_ID')
    ->where('id', $request->session()->get('session_student_id'))
    ->first()->AY_ID;

    $tahun_ajaran = AcademicYear::where('id', '>=',
        $selected_student_ay)->get();

    $detail_mapel_ku = SubjectReport::join('subject_records',
        'subject_reports.SUBJECT_RECORD_ID',
        'subject_records.id')
    ->join('subjects', 'subject_reports.SUBJECTS_ID',
        'subjects.id')
    ->select('subjects.*', 'subject_records.*',
        'subject_reports.*')
    ->where('subjects.DESCRPTION', $mapel->DESCRIPTION)
    ->where('subject_records.ACADEMIC_YEAR_ID', $academic_year_id)
    ->where('subject_records.STUDENTS_ID', $request->session()
        ->get('session_student_id'))
    ->get();

    $data_final_score = SubjectReport::join('subject_records',
        'subject_reports.SUBJECT_RECORD_ID', 'subject_records.id')
    ->join('subjects', 'subject_reports.SUBJECTS_ID',
        'subjects.id')
    ->select('subject_records.*', 'subject_reports.*')
    ->where('subjects.DESCRPTION', $mapel-
    >DESCRIPTION)
    ->where('subject_records.STUDENTS_ID', $request->session()-
    >get('session_student_id'))->get();

    if(isset($url)){
        $select_grade_id = Grade::select('id')
        ->where('NAME', $get_url_param[1])->first()->id;
        $selected_student_in_ay = GradeStudent::select
            ('ACADEMIC_YEAR_ID')
        ->where('STUDENTS_ID', $request->session()
            ->get('session_student_id'))
        ->where('GRADES_ID', $select_grade_id)
        ->first()->ACADEMIC_YEAR_ID;
    }
}
```

**Listing 5.16 Baris program detail nilai mata pelajaran siswa (lanjutan)**

```

else{
    $select_grade_id = Grade::select('id')
                        ->where('NAME', $y)->first()->id;
    $selected_student_in_ay = GradeStudent::select(DB::raw(
        ('MAX(ACADEMIC_YEAR_ID) AS id'))->limit(1)
        ->first()->id;
    }

    $rata_kelas = SubjectReport::join('subject_records',
        'subject_reports.SUBJECT_RECORD_ID',
        'subject_records.id')
    ->join('subjects', 'subject_reports.SUBJECTS_ID',
        'subjects.id')
    ->join('students', 'subject_records.STUDENTS_ID',
        'students.id')
    ->join('grades_students', 'students.id',
        'grades_students.STUDENTS_ID')
    ->select(DB::raw('ROUND(SUM(subject_reports.FINAL_SCORE)/
        COUNT(subject_records.STUDENTS_ID), 2) AS RATAKELAS'),
        'subject_records.*')
    ->where('subjects.DESCRPTION', $mapel->DESCRIPTION)
    ->where('grades_students.GRADES_ID', '=', $select_grade_id)
    ->where('grades_students.ACADEMIC_YEAR_ID',
        $selected_student_in_ay)
    ->groupBy('subject_records.ACADEMIC_YEAR_ID')
    ->get();
}

```

**5.3.6 Halaman Daftar Laporan Ketidaktuntasan**

Halaman daftar laporan ketidaktuntasan adalah halaman yang bisa diakses oleh *user* dengan *role* sebagai kepala sekolah, guru, dan siswa. Jika *user* sebagai kepala sekolah maka hanya bisa melihat isi data yang ditampilkan. Sedangkan *user* sebagai guru/wali kelas bisa melakukan penambahan, pengubahan, atau menghapus daftar laporan. Begitu juga siswa, siswa hanya dapat laporan mengenai dirinya sendiri. Adapun proses yang terdapat pada halaman ini seperti berikut.



## 1. Menginputkan daftar laporan ketidaktuntasan

Proses menginputkan daftar laporan ketidaktuntasan hanya bisa dilakukan oleh guru/wali kelas. Proses ini terjadi ketika *user* menekan tombol “buat daftar ketidaktuntasan” pada bagian atas dari daftar tabel yang disediakan. Untuk melakukan proses penginputan catatan tersebut diperlukan beberapa *field* yang sebagai *request* data dari form input yang kemudian ditangkap oleh *controller* dan menciptakan baris baru dalam tabel *database*. Listing 5.17 dibawah ini menunjukkan bari program get *request* data dari form input pada halaman daftar laporan ketidaktuntasan.

### Listing 5.17 Baris program input daftar laporan ketidaktuntasan

```
public function storeIncomplete(Request $request)
{
    $point = 0;
    $hukuman = "-";
    $date = $request->get('vr_date');
    $request_date = Carbon::parse($date);
    $session_start_ay = $request->session()
        ->get('session_start_ay');
    $start_ay = Carbon::parse($session_start_ay);
    $session_end_ay = $request->session()->get('session_end_ay');
    $end_ay = Carbon::parse($session_end_ay);
    $check = $request_date->between($start_ay,$end_ay);
    if($check == false)
    {
        return redirect('incomplete')->with('error',
            'Input tanggal tidak sesuai dengan tahun ajaran
            yang berlaku');
    }
    else
    {
        $ketidaktuntasan = new ViolationRecord([
            'DATE' => $request->get('vr_date'),
            'STUDENTS_ID' => $request
                ->get('vr_student_name'),
            'DESCRIPTION' => $request
                ->get('vr_description'),
            'PUNISHMENT' => $hukuman,
            'VIOLATIONS_ID' => $request
                ->get('vr_violation_name'),
```

**Listing 5.17 Baris program input daftar laporan ketidaktuntasan (lanjutan)**

```

        'STAFFS_ID' => $request->session()
                    ->get('session_user_id'),
        'TOTAL' => $point,
        'ACADEMIC_YEAR_ID' => $request->session()
                    ->get('session_academic_year_id')
    ]);

    $ketidaktuntasan->save();
    return redirect('incomplete')
    ->with('sukses',
        'Incomplete Report has been created');
    }
}

```

**2. Menginputkan daftar laporan ketidaktuntasan**

Proses mengubah daftar laporan ketidaktuntasan hanya bisa dilakukan oleh *user* sebagai guru/wali kelas. Proses ini terjadi ketika guru menekan tombol “edit” pada kolom aksi pada tabel tersedia. Pengubahan data ini berdasarkan setiap id dari daftar ketidaktuntasan. Setiap id yang ada memiliki isi data yang akan ditampilkan dalam form halaman edit laporan ketidaktuntasan. Baris program edit laporan ketidaktuntasan seperti pada listing 5.18.

**Listing 5.18 Baris program ubah daftar laporan ketidaktuntasan**

```

public function updateIncomplete(Request $request, $id){
    $ketidaktuntasan = ViolationRecord::find($id);
    $point = 0;
    $hukuman = "-";
    $date = $request->get('vr_date');
    $request_date = Carbon::parse($date);
    $session_start_ay = $request->session()
                        ->get('session_start_ay');
    $start_ay = Carbon::parse($session_start_ay);
    $session_end_ay = $request->session()
                    ->get('session_end_ay');
    $end_ay = Carbon::parse($session_end_ay);
    $check = $request_date->between($start_ay,$end_ay);
    if($check == false){

```

**Listing 5.18 Baris program ubah daftar laporan ketidaktuntasan (lanjutan)**

```
        return redirect(action('SubjectController@editIncomplete',
                                $ketidaktuntasan->id))
        ->with('error',
            'Input tanggal tidak sesuai dengan tahun ajaran yang berla
            ku');
    }
    else{
        $ketidaktuntasan->DATE = $request->get('vr_date');
        $ketidaktuntasan->DESCRIPTION = $request
        ->get('vr_desc');
        $ketidaktuntasan->STUDENTS_ID = $request
        ->get('vr_student_name');
        $ketidaktuntasan->VIOLATIONS_ID = $request
        ->get('vr_violation_name');
        $ketidaktuntasan->ACADEMIC_YEAR_ID = $request
        ->session()->get('session_academic_year_id');
        $ketidaktuntasan->STAFFS_ID = $request->session()
        ->get('session_user_id');
        $ketidaktuntasan->PUNISHMENT = $hukuman;
        $ketidaktuntasan->TOTAL = $point;
    }
    $ketidaktuntasan->save();
    return redirect(action('SubjectController@incomplete',
        $ketidaktuntasan->id))->with('sukses',
        'Laporan ketidaktuntasan berhasil diubah');
```

**3. Menghapus daftar laporan ketidaktuntasan**

Proses menghapus daftar laporan ketidaktuntasan hanya bisa dilakukan oleh *user* sebagai guru/wali kelas. Proses ini terjadi ketika guru menekan tombol “delete” pada kolom aksi pada tabel tersedia. Menghapus data ini berdasarkan setiap id dari daftar ketidaktuntasan. Baris program hapus laporan ketidaktuntasan seperti pada listing 5.19.

**Listing 5.19 Baris program hapus laporan ketidaktuntasan**

```
public function destroyIncomplete($id)
{
    $ketidaktuntasan = ViolationRecord::whereId($id)
        ->firstOrFail();
    $ketidaktuntasan->delete();
}
```

**Listing 5.19 Baris program hapus laporan ketidaktuntasan (lanjutan)**

```
return redirect(action('SubjectController@incomplete'))
->with('sukses', 'Laporan ketidaktuntasan berhasil dibuat');
}
```

**5.3.7 Input Nilai Mata Pelajaran**

Proses input nilai setiap mata pelajaran hanya dapat dilakukan oleh *user* guru/wali kelas. Penginputan nilai dengan cara mengimport file *excel* yang telah disediakan kedalam sistem. Pengolahan data terjadi ketika *user* menginputkan file yang ditangkap oleh *controller* import dan dilemparkan ke dalam fungsi *collection* pada *controller* import subject. Baris program import file bisa dilihat pada listing 5.20. Sedangkan untuk menampung semua data yang diimport dapat dilihat pada listing 5.27.

Pada tahap pengolahan nilai, sistem akan membaca terlebih dahulu setiap baris pada file *excel* untuk mendapatkan id dari data tersebut seperti nama mata pelajaran, kelas, nisan siswa, jenis aktivitas, nomor kd. Untuk mendapatkan id setiap data maka dibutuhkan fungsi untuk mengambil id sesuai kondisi data. Potongan kecil fungsi akan dipakai dalam fungsi pengolahan nilai. Potongan fungsi-fungsi kecil tersebut dapat dilihat pada listing 5.21 untuk mengambil deskripsi mata pelajaran, 5.22 untuk mendapatkan nama kelas, 5.23 untuk mendapatkan id siswa, 5.24 untuk mendapatkan id aktivitas, 5.25 untuk mendapatkan id nomor kd, dan 5.26 untuk mendapatkan id mata pelajaran. Baris program proses pengolahan nilai dapat dilihat pada listing 5.28.

**Listing 5.20 Baris program import file excel**

```
public function importAssesment(Request $request)
{
    $this->validate($request, [
        'assesment_import' => 'required|mimes:csv,xls,xlsx'
    ]);

    Excel::import(new SubjectImport($request), $request
        ->file('assesment_import'));
    Session::flash('sukses','Import penilaian siswa berhasil
        ditambahkan');
    return redirect('subject/assesment');
}
```

**Listing 5.21 Baris program fungsi get deksripsi mata pelajaran**

```
Public function selectSubjectDescription($value)
{
    $mapel = DB::SELECT('SELECT DESCRIPTION
                        FROM subjects
                        WHERE subjects.DESRIPTION = "' .
                        $value . '"');

    return $mapel;
}
```

**Listing 5.22 Baris program fungsi get nama kelas**

```
public function selectGradeCode($value)
{
    $kelas = DB::SELECT('SELECT NAME
                        FROM grades
                        WHERE grades.NAME = "' . $value . '"')
;
    return $kelas;
}
```

**Listing 5.23 Baris program fungsi get id siswa**

```
public function selectStudentId($value)
{
    $siswa = DB::SELECT('SELECT id
                        FROM students
                        WHERE students.NISN =
                        "' . $value . '"');

    return $siswa;
}
```

**Listing 5.24 Baris program fungsi get id aktivitas**

```
public function selectActivityId($value)
{
    $aktivitas = DB::SELECT('SELECT id
                            FROM activities
                            WHERE activities.MODULE =
                            "' . $value . '"');
    return $aktivitas;
}
```

**Listing 5.25 Baris program fungsi get id kompetensi dasar**

```
public function selectKdId($value)
{
    $kd = DB::SELECT('SELECT id
                     FROM KD
                     WHERE NUMBER = "' . $value . '"');
    return $kd;
}
```

**Listing 5.26 Baris program fungsi get id mata pelajaran**

```
public function selectSubjectId($subjectName, $gradeCode)
{
    $mapel = Subject::select('id')
                    ->where('DESCRIPTION', $subjectName)
                    ->where('CODE',
                        'LIKE' , '%'. $gradeCode . '%')
                    ->get();
    return $mapel;
}
```

**Listing 5.27 Baris program fungsi collection pengolahan data import**

```
public function collection(Collection $collection){
    foreach($collection as $key => $row)
    {
        if($key >= 1)
        {
            $subject_name = $row[4];
            $tmp_mapel = $this
            ->selectSubjectDescription($subject_name);
            foreach($tmp_mapel as $tmp)
            {
                $mapel = $tmp->DESCRIPTION;
            }
        }
    }
}
```

**Listing 5.27 Baris program fungsi collection pengolahan data import (lanjutan)**

```
        if($key >= 3)
        {
            $grade_code = $row[4];
            $tmp_grade = $this
            ->selectGradeCode($grade_code);
            foreach($tmp_grade as $tm)
            {
                $kodeKelas = $tm->NAME;
            }
            if($key >= 5)
            {
                $student_id = $row[4];
                $id_siswa = $this
                ->selectStudentId($student_id)[0]->id;
                $activity_id = explode(";", $row[6])[1];
                $id_aktivitas = $this
                ->selectActivityId($activity_id)[0]->id;
                $kd_id = $row[7];
                $id_kd = $this->selectKdId($kd_id)[0]
                    ->id;
                $nilai = $row[8];
                $tmp_id_mapel = $this
                ->selectSubjectId($mapel, $kodeKelas);
                foreach($tmp_id_mapel as $tim)
                {
                    $id_mapel = $tim->id;
                }
                $a = ActivityStudent::create([
                    'STUDENTS_ID' => $id_siswa,
                    'ACTIVITIES_ID' => $id_aktivitas,
                    'SUBJECTS_ID' => $id_mapel,
                    'ACADEMIC_YEAR_ID' => $this->request
                    ->session()
                    ->get("session_academic_year_id"),
                    'SCORE' => $nilai,
                ]);
                ActivityKD::create([
                    'KD_ID' => $id_kd,
                    'ACTIVITIES_ID' => $id_aktivitas,
                ]);
            }
        }
    }
    $this->formula($a->SUBJECTS_ID);
}
```

**Listing 5.28 Baris program pengolahan nilai pada fungsi formula**

```
public function formula($mapel_id)
{
    $tmp_nilai = ActivityStudent::join('students',
        'activities_students.STUDENTS_ID', 'students.id')
    ->select('students.*', 'activities_students.*')
    ->where('activities_students.SUBJECTS_ID', $mapel_id)
    ->where('activities_students.ACADEMIC_YEAR_ID',
        $this->request->session()->get("session_academic_year_id"))
    ->get();

    $arrayOfStudent = ActivityStudent::join('students',
        'activities_students.STUDENTS_ID', 'students.id')
    ->select('students.*', 'activities_students.*')
    ->where('activities_students.SUBJECTS_ID', $mapel_id)
    ->where('activities_students.ACADEMIC_YEAR_ID',
        $this->request->session()->get("session_academic_year_id"))
    ->groupBy('activities_students.STUDENTS_ID')
    ->get();

    foreach ($arrayOfStudent as $student)
    {
        $tmp_nama = $student->FNAME . " ". $student->LNAME;
        $tmp_nisn = $student->NISN;
        $tmp_tugas = [];
        $tmp_ph = [];
        $tmp_pts = [];
        $tmp_pas = [];
        $tmp_us = [];
        $tmp_un = [];

        $subject_record = new SubjectRecord();
        $subject_record->ACADEMIC_YEAR_ID = $this->request
            ->session()->get("session_academic_year_id");
        $subject_record->STUDENTS_ID = $student->STUDENTS_ID;
        $subject_record->save();

        foreach ($tmp_nilai as $nilai)
        {
            if($nilai->STUDENTS_ID === $student->STUDENTS_ID){
                if($nilai->ACTIVITIES_ID === 1){
                    array_push($tmp_tugas, $nilai->SCORE);
                }elseif($nilai->ACTIVITIES_ID === 2){
                    array_push($tmp_ph, $nilai->SCORE);
                }else if($nilai->ACTIVITIES_ID === 3){
                    array_push($tmp_pts, $nilai->SCORE);
                }else if($nilai->ACTIVITIES_ID === 4){
                    array_push($tmp_pas, $nilai->SCORE);
                }else if($nilai->ACTIVITIES_ID === 5){
                    array_push($tmp_us, $nilai->SCORE);
                }else if($nilai->ACTIVITIES_ID === 6){
                    array_push($tmp_un, $nilai->SCORE);
                }
            }
        }
    }
}
```



**Listing 5.28 Baris program pengolahan nilai pada fungsi formula (lanjutan)**

```

$count_tugas = $this->countActivity($student
->STUDENTS_ID, ($student->ACTIVITIES_ID = 1));
$sum_score_tugas = $this->sumTotalScore($student
->STUDENTS_ID, ($student->ACTIVITIES_ID = 1));
$count_ph = $this->countActivity($student
->STUDENTS_ID, ($student->ACTIVITIES_ID = 2));
$sum_score_ph = $this->sumTotalScore($student
->STUDENTS_ID, ($student->ACTIVITIES_ID = 2));
$count_pts = $this->countActivity($student
->STUDENTS_ID, ($student->ACTIVITIES_ID = 3));
$sum_score_pts = $this->sumTotalScore($student
->STUDENTS_ID, ($student->ACTIVITIES_ID = 3));
$count_pas = $this->countActivity($student
->STUDENTS_ID, ($student->ACTIVITIES_ID = 4));
$sum_score_pas = $this->sumTotalScore($student
->STUDENTS_ID, ($student->ACTIVITIES_ID = 4));
$count_us = $this->countActivity($student
->STUDENTS_ID, ($student->ACTIVITIES_ID = 5));
$sum_score_us = $this->sumTotalScore($student
->STUDENTS_ID, ($student->ACTIVITIES_ID = 5));
$count_un = $this->countActivity($student
->STUDENTS_ID, ($student->ACTIVITIES_ID = 6));
$sum_score_un = $this->sumTotalScore($student
->STUDENTS_ID, ($student->ACTIVITIES_ID = 6));

foreach($count_tugas as $ct){
    $tmp_ct = $ct->BANYAKAKTIVITAS;
    foreach($sum_score_tugas as $sct){
        $tmp_sct = $sct->TOTALNILAI;
    }
    $avarage_tugas = ($tmp_sct / $tmp_ct) * 0.1;
}
foreach($count_ph as $cph){
    $tmp_cph = $cph->BANYAKAKTIVITAS;
    foreach($sum_score_ph as $scph){
        $tmp_scph = $scph->TOTALNILAI;
    }
    $avarage_ph = ($tmp_scph / $tmp_cph) * 0.2;
}
foreach($count_pts as $cpts){
    $tmp_cpts = $cpts->BANYAKAKTIVITAS;
    foreach($sum_score_pts as $scpts){
        $tmp_scpts = $scpts->TOTALNILAI;
    }
    $avarage_pts = ($tmp_scpts / $tmp_cpts) * 0.3;
}
foreach($count_pas as $cpas){
    $tmp_cpas = $cpas->BANYAKAKTIVITAS;
    foreach($sum_score_pas as $scpas){
        $tmp_scpas = $scpas->TOTALNILAI;
    }
    $avarage_pas = ($tmp_scpas / $tmp_cpas) * 0.4; }

```

**Listing 5.28 Baris program pengolahan nilai pada fungsi formula (lanjutan)**

```

$total_score_student = $avarage_tugas + $avarage_ph +
    $avarage_pts + $avarage_pas;
$fix_total_score_round = round($total_score_student, 2);

$subject_report = SubjectReport::create([
    'SUBJECTS_ID' => $student->SUBJECTS_ID,
    'SUBJECT_RECORD_ID' => $subject_record->id,
    'FINAL_SCORE' => $fix_total_score_round,
    'IS_VERIFIED' => 0,
    'TUGAS' => json_encode($tmp_tugas),
    'PH' => json_encode($tmp_ph),
    'PTS' => json_encode($tmp_pts),
    'PAS' => json_encode($tmp_pas),
    'US' => json_encode($tmp_us),
    'UN' => json_encode($tmp_un),
]);
}
}

```

**5.3.8 Ubah Nilai Mata Pelajaran**

Setiap data yang telah berhasil diimport maka akan tersimpan. Namun, tidak menutupi adanya kesalahan penginputan data maka bisa dilakukan pengubahan data pada halaman *subject assessment* sebelum nilai verifikasi. Proses pengubahan nilai menjalankan satu fungsi “updateAssesment” yang terjadi pada *controller* mata pelajaran. Baris program pengubahan data nilai yang diinput bisa dilihat pada listing 5.29.

**Listing 5.29 Baris program pengubahan nilai yang diinputkan**

```

public function updateAssesment(Request $request, $id)
{
    $laporan_mapel = SubjectReport::find($id);
    $array_tugas = $request->get('a_nilai_tugas');
    $array_ph = $request->get('a_nilai_ph');
    $array_pts = $request->get('a_nilai_pts');
    $array_pas = $request->get('a_nilai_pas');
    $array_un = array();
    $array_us = array();
    $final_score = 0;
}

```

**Listing 5.29 Baris program pengubahan nilai yang diinputkan (lanjutan)**

```
$sum_nilai_tugas = 0;
$rata_nilai_tugas = 0;
$final_nilai_tugas = 0;
foreach($array_tugas as $tugas){
    $sum_nilai_tugas = $sum_nilai_tugas + $tugas;
}
$rata_nilai_tugas = $sum_nilai_tugas /
                    count($array_tugas);
$final_nilai_tugas = $rata_nilai_tugas * 0.1;
$sum_nilai_ph = 0;
$rata_nilai_ph = 0;
$final_nilai_ph = 0;
foreach($array_ph as $ph){
    $sum_nilai_ph = $sum_nilai_ph + $ph;
}
$rata_nilai_ph = $sum_nilai_ph / count($array_ph);
$final_nilai_ph = $rata_nilai_ph * 0.2;

$sum_nilai_pts = 0;
$rata_nilai_pts = 0;
$final_nilai_pts = 0;
foreach($array_pts as $pts){
    $sum_nilai_pts = $sum_nilai_pts + $pts;
}
$rata_nilai_pts = $sum_nilai_pts / count($array_pts);
$final_nilai_pts = $rata_nilai_pts * 0.3;
$sum_nilai_pas = 0;
$rata_nilai_pas = 0;
$final_nilai_pas = 0;
foreach($array_pas as $pas){
    $sum_nilai_pas = $sum_nilai_pas + $pas;
}
$rata_nilai_pas = $sum_nilai_pas / count($array_pas);
$final_nilai_pas = $rata_nilai_pas * 0.4;

$final_score = $final_nilai_tugas + $final_nilai_ph +
               $final_nilai_pts + $final_nilai_pas;
$fix_final_score = round($final_score, 2);

$laporan_mapel->SUBJECTS_ID = $request
                        ->get('a_subject_id');
$laporan_mapel->SUBJECT_RECORD_ID = $laporan_mapel
                        ->SUBJECT_RECORD_ID;
$laporan_mapel->TUGAS = json_encode($array_tugas);
$laporan_mapel->PH = json_encode($array_ph);
$laporan_mapel->PTS = json_encode($array_pts);
$laporan_mapel->PAS = json_encode($array_pas);
$laporan_mapel->UN = json_encode($array_un);
$laporan_mapel->US = json_encode($array_us);
$laporan_mapel->FINAL_SCORE = $fix_final_score;
$laporan_mapel->save();
```

**Listing 5.29 Baris program pengubahan nilai yang diinputkan (lanjutan)**

```
return redirect(action
    ('SubjectController@assessmentImport',
     $laporan_mapel->id))
    ->with('sukses',
        'Data nilai siswa berhasil diubah');
}
```

**5.3.9 Hapus Nilai yang Diinputkan**

Nilai-nilai mata pelajaran siswa yang telah diinputkan, dapat dihapus sebelum nilai diverifikasi. Dengan menjalankan fungsi “destroyAssesment” untuk setiap id nilai siswa yang dipilih maka data akan terhapus. Baris program hapus nilai dapat dilihat pada listing 5.30.

**Listing 5.30 Baris program hapus nilai yang diinput**

```
public function destroyAssesment($id)
{
    $laporan_mapel = SubjectReport::whereId($id)
        ->firstOrFail();
    $laporan_mapel->delete();
    return redirect(action('SubjectController@assessmentImport'))
        ->with('sukses', 'Data nilai siswa berhasil dihapus');
}
```

**5.3.10 Verifikasi Nilai yang Diinputkan**

Nilai-nilai mata pelajaran siswa yang telah diinputkan, dapat diubah dan dihapus sebelum nilai diverifikasi. Proses verifikasi terjadi ketika *user* menekan tombol hijau pada laman *subject assessment* dan menjalankan fungsi “setStatus” seperti pada potongan kode yang dapat dilihat di listing 5.31.

**Listing 5.31 Baris program verifikasi nilai**

```
public function setStatus(Request $request)
{
    $laporan_mapel = SubjectReport::where('IS_VERIFIED', 0)
        ->get();
    foreach($laporan_mapel as $lm){
        $lm->IS_VERIFIED = $request->status;
        $lm->save();
    }
    return redirect(action('SubjectController@assessmentImport'))
        ->with('sukses', 'Daftar nilai berhasil
            di verifikasi dan tersimpan!');
}
```

**5.3.11 Daftar Catatan Penghargaan**

Daftar catatan penghargaan akan menampilkan daftar catatannya seperti nama siswa, kelas, jumlah penghargaan dan total poin penghargaan dengan menggunakan sebuah fungsi pada *controller* penghargaan yaitu “showAchievement” seperti pada listing 5.32. Daftar catatan penghargaan juga menampilkan grafik jumlah catatan setiap tahun ajaran dengan menggunakan fungsi *javascript* yang mendapat data dari *contoller* untuk data grafik seperti pada listing 5.33.

**Listing 5.32 Baris program fungsi show achievement**

```
public function showAchievement($user_id, $ay)
{
    $selected_student = GradeStudent::select(DB::raw('MAX
        (ACADEMIC_YEAR_ID) AS id'))->limit(1)->first()->id;

    $nasional = Achievement::select('id')
        ->where('GRADE', 'Nasional')->get();
    $propinsi = Achievement::select('id')
        ->where('GRADE', 'Propinsi')->get();
    $kota = Achievement::select('id')->where('GRADE', 'Kota')
        ->get();
    $sekolah = Achievement::select('id')
        ->where('GRADE', 'Sekolah')->get();
    if(Auth::guard('web')->user()->staff->ROLE == "TEACHER"){
```

**Listing 5.32 Baris program fungsi show achievement (lanjutan)**

```

$kelas_guru = Grade::where('STAFFS_ID', $user_id)
->first()->id;
$penghargaan = AchievementRecord::join('achievements',
'achievement_records.ACHIEVEMENTS_ID', 'achievements.id')
->join('students', 'achievement_records.STUDENTS_ID',
'students.id')
->join('grades_students', 'students.id',
'grades_students.STUDENTS_ID')
->select('achievement_records.*',
DB::raw('COUNT(*) AS JUMLAHPENGHARGAAN'),
DB::raw('SUM(achievements.POINT) AS POINPENGHARGAAN'))
->where('grades_students.GRADES_ID', $kelas_guru)
->where('achievement_records.ACADEMIC_YEAR_ID', $ay)
->where('grades_students.ACADEMIC_YEAR_ID',
$selected_student)
->groupBy('achievement_records.STUDENTS_ID')
->get();

$count_nasional = $this->countAchievementPerGrade
($nasional, $ay, $kelas_guru, $selected_student)
->first()->JMLH;
$count_propinsi = $this->countAchievementPerGrade
($propinsi, $ay, $kelas_guru, $selected_student)
->first()->JMLH;
$count_kota = $this->countAchievementPerGrade
($kota, $ay, $kelas_guru, $selected_student)
->first()->JMLH;
$count_sekolah = $this->countAchievementPerGrade
($sekolah, $ay, $kelas_guru, $selected_student)
->first()->JMLH;

$count_achievement_list = Achievement::select
(DB::raw('COUNT(*) AS JUMLAH'))->first()->JUMLAH;

$res_nasional = ROUND(($count_nasional /
$count_achievement_list) * 100);
$res_propinsi = ROUND(($count_propinsi /
$count_achievement_list) * 100);
$res_kota = ROUND(($count_kota /
$count_achievement_list) * 100);
$res_sekolah = ROUND(($count_sekolah /
$count_achievement_list) * 100);

$data = DB::select("SELECT GRADE AS TINGKAT,
COUNT(*) AS JUMLAH,
(SELECT (CASE WHEN TINGKAT = 'Kota' THEN $res_kota
WHEN TINGKAT = 'Nasional' THEN $res_nasional
WHEN TINGKAT = 'Propinsi' THEN $res_propinsi
WHEN TINGKAT = 'Sekolah' THEN $res_sekolah
END))
AS PERSENTASE

```

**Listing 5.32 Baris program fungsi show achievement (lanjutan)**

```

FROM achievement_records ar INNER JOIN achievements a ON
ar.ACHIEVEMENTS_ID = a.id
INNER JOIN students s ON ar.STUDENTS_ID = s.id
INNER JOIN grades_students gs ON gs.STUDENTS_ID = s.id
WHERE ar.ACADEMIC_YEAR_ID = " . $ay . " AND gs.GRADES_ID
= " . $kelas_guru . " AND
gs.ACADEMIC_YEAR_ID = ". $selected_student . "
GROUP BY TINGKAT
ORDER BY TINGKAT DESC");
}
elseif(Auth::guard('web')->user()->staff
->ROLE == "HEADMASTER")
{
    $penghargaan = AchievementRecord::join('achievements',
    'achievement_records.ACHIEVEMENTS_ID', 'achievements.id')
    ->join('students', 'achievement_records.STUDENTS_ID',
    'students.id')
    ->join('grades_students', 'students.id',
    'grades_students.STUDENTS_ID')
    ->select('achievement_records.*',
    DB::raw('COUNT(*) AS JUMLAHPENGHARGAAN'),
    DB::raw('SUM(achievements.POINT) AS POINPENGHARGAAN'))
    ->where('achievement_records.ACADEMIC_YEAR_ID', $ay)
    ->where('grades_students.ACADEMIC_YEAR_ID',
    $selected_student)
    ->groupBy('achievement_records.STUDENTS_ID')
    ->get();

    $count_nasional = $this->countAchievementPerGradeKepsek
    ($nasional, $ay, $selected_student)->first()->JMLH;
    $count_propinsi = $this->countAchievementPerGradeKepsek
    ($propinsi, $ay, $selected_student)->first()->JMLH;
    $count_kota = $this->countAchievementPerGradeKepsek
    ($kota, $ay, $selected_student)->first()->JMLH;
    $count_sekolah = $this->countAchievementPerGradeKepsek
    ($sekolah, $ay, $selected_student)->first()->JMLH;

    $count_achievement_list = Achievement::select
    (DB::raw('COUNT(*) AS JUMLAH'))->first()->JUMLAH;

    $res_nasional = ROUND(($count_nasional /
    $count_achievement_list) * 100);
    $res_propinsi = ROUND(($count_propinsi /
    $count_achievement_list) * 100);
    $res_kota = ROUND(($count_kota /
    $count_achievement_list) * 100);
    $res_sekolah = ROUND(($count_sekolah /
    $count_achievement_list) * 100);

```

**Listing 5.32 Baris program fungsi show achievement (lanjutan)**

```

    $data = DB::select("SELECT GRADE AS TINGKAT,
                        COUNT(*) AS JUMLAH,
                        (SELECT(CASE WHEN TINGKAT = 'Kota' THEN $res_kota
                                    WHEN TINGKAT = 'Nasional' THEN $res_nasional
                                    WHEN TINGKAT = 'Propinsi' THEN $res_propinsi
                                    WHEN TINGKAT = 'Sekolah' THEN $res_sekolah
                                    END))
                        AS PERSENTASE
FROM achievement_records ar INNER JOIN achievements a ON
ar.ACHIEVEMENTS_ID = a.id
INNER JOIN students s ON ar.STUDENTS_ID = s.id
INNER JOIN grades_students gs ON gs.STUDENTS_ID = s.id
WHERE ar.ACADEMIC_YEAR_ID = " . $ay . "
AND gs.ACADEMIC_YEAR_ID = " . $selected_student . "
GROUP BY TINGKAT
ORDER BY TINGKAT DESC");
}
else{
    $penghargaan = AchievementRecord::join('achievements',
    'achievement_records.ACHIEVEMENTS_ID', 'achievements.id')
    ->join('students', 'achievement_records.STUDENTS_ID',
    'students.id')
    ->join('grades_students', 'students.id',
    'grades_students.STUDENTS_ID')
    ->select('achievement_records.*',
    DB::raw('COUNT(*) AS JUMLAHPENGHARGAAN'),
    DB::raw('SUM(achievements.POINT) AS POINPENGHARGAAN'))
    ->where('achievement_records.ACADEMIC_YEAR_ID', $ay)
    ->where('grades_students.ACADEMIC_YEAR_ID',
    $selected_student)
    ->groupBy('achievement_records.STUDENTS_ID')
    ->get();
}

$value["penghargaan"] = $penghargaan;
$value["data"] = $data;
return $value;
}

```

**Listing 5.33 Baris program informasi penghargaan dengan grafik**

```

var types = {!!json_encode($type)!!}
var dataGraph = {!!json_encode($data)!!}
var selectedTahunAjaran = {!!json_encode
($selected_tahun_ajaran)!!}
var startMonth = selectedTahunAjaran.STARTMONTH;
var endMonth = selectedTahunAjaran.ENDMONTH;
var dataSeries = []
var dataCategory = []

```



**Listing 5.33 Baris program informasi penghargaan dengan grafik (lanjutan)**

```
types.forEach(function(item){
    var values = []
    for(var i=startMonth; i <= endMonth; i++){
        values[i-startMonth] = 0
        dataCategory.push('Bulan ' + i)
    }
    dataGraph.forEach(function(obj){
        if(obj.TINGKAT == item.TINGKAT){
            values[obj.BULAN - startMonth] = obj.JUMLAH
        }
    })
    dataSeries.push({
        name: item.TINGKAT,
        data: values
    })
})
Highcharts.chart('achievementChartStatistic', {
    chart: {
        type: 'column'
    },
    title: {
        text: 'STATISTIK PENGHARGAAN YANG TERCATAT
        BERDASARKAN JENIS PENGHARGAAN DAN PERIODE TAHUN AJARAN'
    },
    xAxis: {
        categories: dataCategory,
        crosshair: true
    },
    yAxis: {
        min: 0,
        allowDecimals: false,
        title: {
            text: 'JUMLAH PENGHARGAAN TERCATAT'
        }
    },
    credits: {
        enabled: false,
    },
    plotOptions: {
        bar: {
            dataLabels: {
                enabled: true
            },
            enableMouseTracking: true,
            pointPadding: 0.08
        }
    },
    series: dataSeries
});
```

### 5.3.12 Persentase Penghargaan

Persentase penghargaan adalah tabel yang muncul ketika *user* menekan tombol persentase penghargaan. Data yang akan ditampilkan adalah kategori, jumlah tercatat dan nilai persentase setiap catatan penghargaan per tingkatnya. Data-data yang ditampilkan terproses pada fungsi “showAchievement” seperti pada listing 5.32 diatas. Namun untuk menampilkan tabel ini menggunakan fungsi *jQuery* seperti pada listing 5.34 dibawah ini.

#### Listing 5.34 Baris program menampilkan tabel persentase penghargaan

```
$('#button-persen-penghargaan').on('click', (function(){  
    $('#persentase-penghargaan').show();  
}))
```

### 5.3.13 Detail Penghargaan Berdasarkan Tingkat

Detail penghargaan berdasarkan tingkat menampilkan sebuah tabel ketika menekan tombol dengan icon mata pada tabel persentase penghargaan dimana fungsi tersebut menggunakan fungsi *ajax*. Fungsi *ajax* tersebut melakukan *request* get data dari fungsi “showAchievementItemByGrade” dari *controller*. Baris program fungsi “showAchievementItemByGrade” bisa dilihat pada listing 5.35. Baris program *ajax* catatan penghargaan berdasarkan tingkat dapat dilihat pada listing 5.36

#### Listing 5.35 Baris program fungsi showAchievementItemByGrade

```
public function showAchievementItemByGrade(Request $request)  
{  
    $selected_student = GradeStudent::select(DB::raw('MAX  
    (ACADEMIC_YEAR_ID) AS id'))->limit(1)->first()->id;  
  
    if(Auth::guard('web')->user()->staff->ROLE == "TEACHER"){
```

**Listing 5.35 Baris program fungsi showAchievementItemByGrade (lanjutan)**

```

$kelas_guru = Grade::where('STAFFS_ID', $request
->session()->get('session_user_id'))->first()->id;
$penghargaan = Achievement::join('achievement_records',
'achievements.id', 'achievement_records.ACHIEVEMENTS_ID')
->join('students', 'achievement_records.STUDENTS_ID',
'students.id')
->join('grades_students', 'students.id',
'grades_students.STUDENTS_ID')
->select('achievements.*',
DB::raw('COUNT(*) AS JUMLAHPERPENGHARGAAN'))
->where('GRADE', $request->achievementGrade)
->where('grades_students.GRADES_ID', $kelas_guru)
->where('achievement_records.ACADEMIC_YEAR_ID', $request
->academicYearId)
->where('grades_students.ACADEMIC_YEAR_ID',
$selected_student)
->groupBy('achievements.id')
->get();
}
elseif(Auth::guard('web')->user()->staff
->ROLE == "HEADMASTER")
{
$penghargaan = Achievement::join('achievement_records',
'achievements.id', 'achievement_records.ACHIEVEMENTS_ID')
->join('students', 'achievement_records.STUDENTS_ID',
'students.id')
->join('grades_students', 'students.id',
'grades_students.STUDENTS_ID')
->select('achievements.*',
DB::raw('COUNT(*) AS JUMLAHPERPENGHARGAAN'))
->where('GRADE', $request->achievementGrade)
->where('achievement_records.ACADEMIC_YEAR_ID', $request
->academicYearId)
->where('grades_students.ACADEMIC_YEAR_ID',
$selected_student)
->groupBy('achievements.id')
->get();
}
return $penghargaan;
}

```

**Listing 5.36 Baris program fungsi ajax detail penghargaan berdasar tingkat**

```

$('#tbody-persentase-penghargaan').on('click', '.button-detail
-persentase', (function(){
var achievementGrade = $(this).attr('achievement-grade')
var academicYearId = $('#selector-dropdown
-achievementrecord-year').val();
$('#detail-penghargaan-berdasarkan-tingkat').show();

```

**Listing 5.36 Baris program fungsi ajax detail penghargaan berdasar tingkat (lanjutan)**

```
$.ajax({
  url: '{{route("achievementrecord.achievementItem")}}',
  type: 'get',
  data: {achievementGrade: achievementGrade,
        academicYearId: academicYearId},

  success: function(result) {
    $('#tbody-detail-penghargaan-berdasarkan-tingkat').empty()
    result.forEach(function(obj) {
      $('#tbody-detail-penghargaan-berdasarkan-tingkat').append(
        <tr>
          <td>${obj.GRADE}</td>
          <td>${obj.DESRIPTION}</td>
          <td>${obj.JUMLAHPERPENGHARGAAN}</td>
          <td>
            <button type="button" class="btn btn-primary btn-sm button-detail" achievement-id=${obj.id}><i class="fa fa-eye"></i></button>
          </td>
        </tr>
      )
    })
  },
  error: function(err) {
    console.log(err)
  }
})
})
})
```

### 5.3.14 Detail Penghargaan Setiap Siswa

Detail penghargaan setiap siswa akan menampilkan data penghargaan setiap id siswa. Jika pada halaman daftar catatan penghargaan terdapat nama dan jumlah penghargaan, maka detail penghargaan ini menampilkan data-datanya. Tampilan detail akan berupa modal form. Untuk menampilkan modal tersebut maka perlu menekan tombol yang menjalankan fungsi *ajax* yang melakukan *request* data dari method “studentDetailAchievement” dari *ajax controller*. Baris program fungsi

“studentDetailAchievement” bisa dilihat pada listing 5.37. Baris program ajax catatan penghargaan berdasarkan tingkat dapat dilihat pada listing 5.38.

#### Listing 5.37 Baris program fungsi studentDetailAchievement

```
public function studentDetailAchievement(Request $request)
{
    $penghargaan_tiap_siswa = AchievementRecord::join
    ('achievements', 'achievement_records.ACHIEVEMENTS_ID',
    'achievements.id')
    ->join('academic_years',
    'achievement_records.ACADEMIC_YEAR_ID', 'academic_years.id')
    ->join('students', 'achievement_records.STUDENTS_ID',
    'students.id')
    ->select('achievement_records.id',
    'achievement_records.DATE',
    'achievement_records.DESCRPTION AS DESKRIPSI1',
    'achievements.DESCRPTION AS DESKRIPSI2' ,
    'achievements.GRADE',
    'achievements.POINT', 'students.FNAME',
    'students.LNAME', 'academic_years.TYPE',
    'academic_years.NAME')
    ->where('STUDENTS_ID', $request->studentId)
    ->where('achievement_records.ACADEMIC_YEAR_ID',
    $request->academicYearId)
    ->orderBy('achievement_records.id', 'DESC')
    ->get();

    return $penghargaan_tiap_siswa;
}
```

#### Listing 5.38 Baris program fungsi ajax detail penghargaan setiap siswa

```
$('#tbody-daftar-siswa-yang-ada-penghargaan')
.on('click', '.button-detail-penghargaan', (function(){
    var studentId = $(this).attr('student-id');
    var academicYearId = $('#selector-dropdown
    -achievementrecord-year').val();

    $.ajax({
        url: '{{ route("achievementrecord
        .studentDetailAchievement") }}',
        type: 'get',
        data: {studentId: studentId,
        academicYearId: academicYearId},

        success: function(result){
```

**Listing 5.38 Baris program fungsi ajax detail penghargaan setiap siswa (lanjutan)**

```

$('#tbody-student-achievement-list').empty()

result.forEach(function(obj){
var route = "{{ route('achievementrecord.edit', ':id') }}"
}}"
route = route.replace(':id', `${obj.id}`)
var route_del = "{{ route ('achievementrecord.destroy', ':id') }}"
route_del = route_del.replace(':id', `${obj.id}`)
var $role_teacher = "{{ Auth::guard('web')->user()
->staff->ROLE === 'TEACHER' }}"
var $role_headmaster = "{{ Auth::guard('web')->user()
->staff->ROLE === 'HEADMASTER' }}"

if($role_teacher){
$('#tbody-student-achievement
-list').append(
`
<tr>
<td style="width: 20px">${obj.id}</td>
<td style="width: 90px">${obj.FNAME} ${obj.LNAME}
</td>
<td style="width: 20px">${obj.DATE}</td>
<td style="width: 60px">${obj.TYPE} - ${obj.NAME}
</td>
<td style="width: 40px">${obj.GRADE}</td>
<td style="width: 200px">${obj.DESKRIPSI2}</td>
<td style="width: 200px">${obj.DESKRIPSI1}</td>
<td style="width: 40px">${obj.POINT}</td>
<td style="width: 80px">
<a href=${route} class="btn btn-warning btn-sm">
<i class="fa fa-pencil"></i>
</a>
<form action=${route_del} method="POST" class="in
line">
@method('delete')
@csrf
<button class="btn btn-danger btn
-sm" onclick="return confirm('Are you sure?') "
value="DELETE">
<i class="fa fa-trash"></i>
</button>
</form>
</td>
</tr>
`
)
}
}

```

**Listing 5.38 Baris program fungsi ajax detail penghargaan setiap siswa (lanjutan)**

```

        else if ($role_headmaster) {
            $('#tbody-student-achievement-
            -list').append(
                `
                <tr>
                    <td style="width: 20px">${obj.id}</td>
                    <td style="width: 90px">${obj.FNAME} ${obj.LNAM
                    E}</td>
                    <td style="width: 20px">${obj.DATE}</td>
                    <td style="width: 60px">${obj.TYPE} - ${obj.NAM
                    E}</td>
                    <td style="width: 40px">${obj.GRADE}</td>
                    <td style="width: 200px">${obj.DESKRIPSI2}</td>
                    <td style="width: 200px">${obj.DESKRIPSI1}</td>

                    <td style="width: 40px">${obj.POINT}</td>

                </tr>
                `
            )
        }
    })
    $('#modalDetailAchievementEachStudent').modal('show')
},
error: function(err){
    console.log(err)
}
})
});

```

### 5.3.15 Daftar Catatan Pelanggaran

Daftar catatan pelanggaran akan menampilkan daftar catatannya seperti nama siswa, kelas, jumlah pelanggaran dan total poin pelanggaran dengan menggunakan sebuah fungsi pada *controller* pelanggaran yaitu “showViolation” seperti pada listing 5.39. Daftar catatan pelanggaran juga menampilkan grafik jumlah catatan setiap tahun ajaran dengan menggunakan fungsi *javascript* yang mendapat data dari *contoller* untuk data grafik seperti pada listing 5.40.

**Listing 5.39 Baris program fungsi showViolation**

```

public function showViolation($user_id, $ay)
{
    $selected_student = GradeStudent::select(DB::raw('MAX
    (ACADEMIC_YEAR_ID) AS id'))->limit(1)->first()->id;

    $r = Violation::select('id')->where('NAME', 'LIKE', 'R%')
    ->get();
    $b = Violation::select('id')->where('NAME', 'LIKE', 'B%')
    ->get();
    $sb = Violation::select('id')->where('NAME', 'LIKE', 'SB%')
    ->get();

    if(Auth::guard('web')->user()->staff->ROLE == "TEACHER"){
        $kelas_guru = Grade::where('STAFFS_ID', $user_id)->first()
        ->id;

        $pelanggaran = ViolationRecord::join('violations',
        'violation_records.VIOLATIONS_ID', 'violations.id')
        ->join('students', 'violation_records.STUDENTS_ID',
        'students.id')
        ->join('grades_students', 'students.id',
        'grades_students.STUDENTS_ID')
        ->select('violation_records.*',
        DB::raw('COUNT(*) AS JUMLAHPELANGGARAN'),
        DB::raw('SUM(violation_records.TOTAL) AS POINPELANGGARAN'
        ))
        ->where('violations.NAME', 'NOT LIKE', 'TTS%')
        ->where('grades_students.GRADES_ID', $kelas_guru)
        ->where('violation_records.ACADEMIC_YEAR_ID', $ay)
        ->where('grades_students.ACADEMIC_YEAR_ID',
        $selected_student)
        ->groupBy('violation_records.STUDENTS_ID')
        ->orderBy('JUMLAHPELANGGARAN', 'DESC')
        ->get();

        $count_r = $this->countViolationPerCategory
        ($r, $ay, $kelas_guru, $selected_student)->first()->JMLH;
        $count_b = $this->countViolationPerCategory
        ($b, $ay, $kelas_guru, $selected_student)->first()->JMLH;
        $count_sb = $this->countViolationPerCategory
        ($sb, $ay, $kelas_guru, $selected_student)->first()->JMLH;

        $count_violation_list = Violation::select(DB::raw
        ('COUNT(*) AS JUMLAH'))->where('NAME', 'NOT LIKE', 'TTS%')
        ->first()->JUMLAH;

        $data = DB::select("SELECT (CASE WHEN v.NAME LIKE 'R%'
        THEN 'RINGAN'
        WHEN v.NAME LIKE 'B%' THEN 'BERAT'
        WHEN v.NAME LIKE 'SB%' THEN 'SANGATBERAT'
        END) AS KATEGORI,
        COUNT(*) AS JUMLAH,

```



**Listing 5.39 Baris program fungsi showViolation (lanjutan)**

```

        (SELECT (CASE WHEN KATEGORI = 'RINGAN' THEN ROUND
            ($count_r / $count_violation_list * 100)
        WHEN KATEGORI = 'BERAT' THEN ROUND
            ( $count_b / $count_violation_list * 100)
        WHEN KATEGORI = 'SANGATBERAT' THEN ROUND
            ( $count_sb / $count_violation_list * 100)
        END))AS PERSENTASE
    FROM violation_records vr INNER JOIN violations v ON
    vr.VIOLATIONS_ID = v.id
    INNER JOIN students s ON vr.STUDENTS_ID = s.id
    INNER JOIN grades_students gs ON gs.STUDENTS_ID = s.id
    WHERE vr.ACADEMIC_YEAR_ID = " . $ay . " AND
    v.NAME NOT LIKE 'TTS%'
    AND gs.GRADES_ID = " . $kelas_guru . "
    AND gs.ACADEMIC_YEAR_ID = ". $selected_student ."
    GROUP BY KATEGORI
    ORDER BY KATEGORI DESC");
}
elseif(Auth::guard('web')->user()->staff
->ROLE == "HEADMASTER"){

    $pelanggaran = ViolationRecord::join('violations',
    'violation_records.VIOLATIONS_ID', 'violations.id')
    ->join('students', 'violation_records.STUDENTS_ID',
    'students.id')
    ->join('grades_students', 'students.id',
    'grades_students.STUDENTS_ID')
    ->select('violation_records.*',
    DB::raw('COUNT(*) AS JUMLAHPELANGGARAN'),
    DB::raw('SUM(violation_records.TOTAL) AS POINPELANGGARAN'))
    ->where('violations.NAME', 'NOT LIKE', 'TTS%')
    ->where('violation_records.ACADEMIC_YEAR_ID', $ay)
    ->where('grades_students.ACADEMIC_YEAR_ID',
    $selected_student)
    ->groupBy('violation_records.STUDENTS_ID')
    ->orderBy('JUMLAHPELANGGARAN', 'DESC')->get();

    $count_r = $this->countViolationPerCategoryKepsek
    ($r, $ay, $selected_student)->first()->JMLH;
    $count_b = $this->countViolationPerCategoryKepsek
    ($b, $ay, $selected_student)->first()->JMLH;
    $count_sb = $this->countViolationPerCategoryKepsek
    ($sb, $ay, $selected_student)->first()->JMLH;

    $data = DB::select("SELECT (CASE WHEN v.NAME LIKE 'R%'
    THEN 'RINGAN'
    WHEN v.NAME LIKE 'B%' THEN 'BERAT'
    WHEN v.NAME LIKE 'SB%' THEN 'SANGATBERAT'
    END) AS KATEGORI, COUNT(*) AS JUMLAH,

```

**Listing 5.39 Baris program fungsi showViolation (lanjutan)**

```

(SELECT (CASE WHEN KATEGORI = 'RINGAN' THEN ROUND
( $count_r / (SELECT COUNT(*) FROM violations
WHERE violations.NAME LIKE 'R%') * 100)
WHEN KATEGORI = 'BERAT' THEN ROUND( $count_b /
(SELECT COUNT(*) FROM violations
WHERE violations.NAME LIKE 'B%') * 100)
WHEN KATEGORI = 'SANGATBERAT' THEN ROUND( $count_sb /
(SELECT COUNT(*) FROM violations WHERE
violations.NAME LIKE 'S%') * 100)
END)) AS PERSENTASE
FROM violation_records vr INNER JOIN violations v ON
vr.VIOLATIONS_ID = v.id
INNER JOIN students s ON vr.STUDENTS_ID = s.id
INNER JOIN grades_students gs ON gs.STUDENTS_ID = s.id
WHERE vr.ACADEMIC_YEAR_ID = " . $ay . " AND
v.NAME NOT LIKE 'TTS%' AND
gs.ACADEMIC_YEAR_ID = " . $selected_student . "
GROUP BY KATEGORI
ORDER BY KATEGORI DESC");
}
else{
    $pelanggaran = ViolationRecord::where('violation_records
.ACADEMIC_YEAR_ID', $ay)
    ->orderBy('violation_records.id', 'DESC')->get();
}
$value["pelanggaran"] = $pelanggaran;
$value["data"] = $data;
return $value;
}

```

**Listing 5.40 Baris program informasi pelanggaran dengan grafik**

```

var categories = {!!json_encode($kategori)!!}
var dataGraph = {!!json_encode($data)!!}
var selectedTahunAjaran = {!!json_encode
($selected_tahun_ajaran)!!}
var startMonth = selectedTahunAjaran.STARTMONTH;
var endMonth = selectedTahunAjaran.ENDMONTH;
var dataSeries = [];
var dataCategory = []
categories.forEach(function(item) {
    var values = []
    for(var i=startMonth; i <= endMonth; i++){
        values[i-startMonth] = 0
        dataCategory.push('Bulan ' + i)
    }
    dataGraph.forEach(function(obj) {
        if(obj.KATEGORI == item.KATEGORI){
            values[obj.BULAN - startMonth] = obj.JUMLAH

```

**Listing 5.40 Baris program informasi pelanggaran dengan grafik (lanjutan)**

```

    })
    dataSeries.push({
        name: item.KATEGORI,
        data: values
    })
})
Highcharts.setOptions({
    colors: ['#F21402', '#028CF2 ', '#E4F202 ', '028CF2']
});
Highcharts.chart('violationChartStatistic', {
    chart: {
        type: 'column'
    },
    title: {
        text: 'STATISTIK PELANGGARAN YANG TERCATAT
BERDASARKAN JENIS PELANGGARAN DAN PERIODE TAHUN AJARAN'
    },
    xAxis: {
        categories: dataCategory,
        crosshair: true
    },
    yAxis: {
        min: 0,
        allowDecimals: false,
        title: {
            text: 'JUMLAH PELANGGARAN TERCATAT'
        }
    },
    credits: {
        enabled: false,
    },
    plotOptions: {
        column: {
            pointPadding: 0.08,
            borderWidth: 0
        }
    },
    series: dataSeries
});

```

**5.3.16 Persentase Pelanggaran**

Persentase pelanggaran adalah tabel yang muncul ketika *user* menekan tombol persentase pelanggaran. Data yang akan ditampilkann adalah kategori, jumlah tercatat dan nilai persentase setiap catatan pelanggaran per kategorinya. Data-data yang ditampilkan terproses pada fungsi “showViolation” seperti pada

listing 5.39 diatas. Namun untuk menampilkan tabel ini menggunakan fungsi *jQuery* seperti pada listing 5.41 dibawah ini.

**Listing 5.41 Baris program menampilkan tabel persentase pelanggaran**

```
$('#button-persen-pelanggaran').on('click', (function(){  
    $('#persentase-pelanggaran').show();  
}))
```

### 5.3.17 Detail Pelanggaran Berdasarkan Kategori

Detail pelanggaran berdasarkan tingkat menampilkan sebuah tabel ketika menekan tombol dengan icon mata pada tabel persentase pelanggaran dimana fungsi tersebut menggunakan fungsi *ajax*. Fungsi *ajax* tersebut melakukan *request* get data dari fungsi “showViolationItemByCategory” dari *controller*. Baris program fungsi “showViolationItemByCategory” bisa dilihat pada listing 5.42. Baris program *ajax* catatan penghargaan berdasarkan tingkat dapat dilihat pada listing 5.43.

**Listing 5.42 Baris program fungsi showViolationItemByCategory**

```
public function showViolationItemByCategory(Request $request)  
{  
    $selected_student = GradeStudent::select(DB::raw('MAX  
    (ACADEMIC_YEAR_ID) AS id'))->limit(1)->first()->id;  
  
    if(Auth::guard('web')->user()->staff->ROLE == "TEACHER"){  
        $kelas_guru = Grade::where('STAFFS_ID',  
        $request->session()  
        ->get('session_user_id'))->first()->id;  
  
        $pelanggaran = Violation::join('violation_records',  
        'violations.id', 'violation_records.VIOLATIONS_ID')  
        ->join('students', 'violation_records.STUDENTS_ID',  
        'students.id')  
        ->join('grades_students', 'students.id',  
        'grades_students.STUDENTS_ID')
```

**Listing 5.42 Baris program fungsi showViolationItemByCategory (lanjutan)**

```

->select('violations.*',
DB::raw('COUNT(*) AS JUMLAHPERPELANGGARAN'))
->where('NAME', 'LIKE', $request->violationName . '%' )
->where('grades_students.GRADES_ID', $kelas_guru)
->where('violation_records.ACADEMIC_YEAR_ID',
        $request->academicYearId)
->where('grades_students.ACADEMIC_YEAR_ID',
        $selected_student)
->groupBy('violations.id')
->get();
}
elseif(Auth::guard('web')->user()->staff
->ROLE == "HEADMASTER")
{
    $pelanggaran = Violation::join('violation_records',
    'violations.id', 'violation_records.VIOLATIONS_ID')
->join('students', 'violation_records.STUDENTS_ID',
        'students.id')
->join('grades_students', 'students.id',
        'grades_students.STUDENTS_ID')
->select('violations.*',
        DB::raw('COUNT(*) AS JUMLAHPERPELANGGARAN'))
->where('NAME', 'LIKE', $request->violationName . '%' )
->where('violation_records.ACADEMIC_YEAR_ID',
        $request->academicYearId)
->where('grades_students.ACADEMIC_YEAR_ID',
        $selected_student)
->groupBy('violations.id')
->get();
}
return $pelanggaran;
}

```

**Listing 5.43 Baris program fungsi ajax detail pelanggaran berdasar kategori**

```

$('#tbody-persentase-pelanggaran').on('click', '.button-detail-
persentase', (function(){
    var x = $(this).attr('violation-category')
    var res = x.split("")
    var violationName = res[0]
    var academicYearId = $('#selector-dropdown
-violationrecord-year').val()

    $('#detail-pelanggaran-berdasarkan
-kategori').show();

    $.ajax({
        url: '{{route("violationrecord.violationItem")}}',
        type: 'get',

```

**Listing 5.43 Baris program fungsi ajax detail pelanggaran berdasar kategori (lanjutan)**

```

        data: {violationName: violationName,
                academicYearId: academicYearId},

        success: function(result) {
            $('#tbody-detail-pelanggaran-berdasarkan-
            -kategori').empty()

            result.forEach(function(obj) {
                $('#tbody-detail-pelanggaran-berdasarkan-
                -kategori').append(
                    `
                    <tr>
                        <td>${obj.NAME}</td>
                        <td>${obj.DESRIPTION}</td>
                        <td>${obj.JUMLAHPERPELANGGARAN}</td>
                        <td>
                            <button type="button" class="btn
                            btn-primary btn-sm button
                            -detail"
                            violation-id=${obj.id}>
                                <i class="fa fa-eye"></i>
                            </button>
                        </td>
                    </tr>
                    `
                )
            })
        },
        error: function(err) {
            console.log(err)
        }
    })
}))

```

### 5.3.18 Detail Pelanggaran Setiap Siswa

Detail pelanggaran setiap siswa akan menampilkan data pelanggaran setiap id siswa. Jika pada halaman daftar catatan pelanggaran terdapat nama dan jumlah pelanggaran, maka detail pelanggaran ini menampilkan data-datanya. Tampilan detail akan berupa modal form. Untuk menampilkan modal tersebut maka perlu menekan tombol yang menjalankan fungsi *ajax* yang melakukan *request* data dari

method “studentDetailViolation” dari *ajax controller*. Baris program fungsi “studentDetailViolation” bisa dilihat pada listing 5.44. Baris program ajax catatan pelanggaran berdasarkan tingkat dapat dilihat pada listing 5.45.

**Listing 5.44 Baris program fungsi studentDetailViolation**

```
public function studentDetailViolation(Request $request)
{
    $pelanggaran_tiap_siswa = ViolationRecord::join
    ('violations', 'violation_records.VIOLATIONS_ID',
    'violations.id')
    ->join('academic_years',
    'violation_records.ACADEMIC_YEAR_ID',
    'academic_years.id')
    ->join('students', 'violation_records.STUDENTS_ID',
    'students.id')
    ->select('violation_records.id', 'violations.POINT' ,
    'violation_records.DATE' ,
    'violation_records.PUNISHMENT',
    'violations.DESCRPTION',
    'students.FNAME', 'students.LNAME',
    'academic_years.TYPE', 'academic_years.NAME')
    ->where('STUDENTS_ID', $request->studentId)
    ->where('violation_records.ACADEMIC_YEAR_ID',
    $request->academicYearId)
    ->orderBy('violation_records.id', 'DESC')
    ->get();

    return $pelanggaran_tiap_siswa;
}
```

**Listing 5.45 Baris program fungsi ajax detail pelanggaran setiap siswa**

```
$('#tbody-daftar-siswa-yang-ada-pelanggaran')
.on('click', '.button-detail-pelanggaran', (function(){
    var studentId = $(this).attr('student-id');
    var academicYearId = $('#selector-dropdown
    -violationrecord-year').val()

    $.ajax({
        url: '{{ route("violationrecord
        .studentDetailViolation ") }}',
        type: 'get',
        data: {studentId: studentId,
        academicYearId: academicYearId},
```

**Listing 5.45 Baris program fungsi ajax detail pelanggaran setiap siswa  
(lanjutan)**

```

success: function(result){
$('#tbody-student-violation-list').empty()
result.forEach(function(obj){
    var route = "{{ route('violationrecord.edit', ':id') }}"
    route = route.replace(':id', `${obj.id}`)
    var route_del = "{{ route('violationrecord.destroy', ':id') }}"
    route_del = route_del.replace(':id', `${obj.id}`)
    var $role_teacher = "{{ Auth::guard('web')->user()->staff->ROLE === 'TEACHER' }}"
    var $role_headmaster = "{{ Auth::guard('web')->user()->staff->ROLE === 'HEADMASTER' }}"
    if($role_teacher)
    {
        $('#tbody-student-violation-list').append(
            <tr>
                <td style="width: 20px">${obj.id}</td>
                <td style="width: 90px">${obj.FNAME}
                    ${obj.LNAME}</td>
                <td style="width: 20px">${obj.DATE}</td>
                <td style="width: 60px">${obj.TYPE} -
                    ${obj.NAME}</td>
                <td style="width: 110px">${obj.DESCRPTION}</td>
                <td style="width: 110px">${obj.PUNISHMENT}</td>
                <td style="width: 40px">${obj.POINT}</td>
                <td style="width: 80px">
                    <a href=${route} class="btn btn-warning btn-sm">
                        <i class="fa fa-pencil"></i>
                    </a>
                    <form action=${route_del} method="POST" class="inline">
                        @method('delete')
                        @csrf
                        <button class="btn btn-danger btn-sm" onclick=
                            "return confirm('Are you sure?')"
                            value="DELETE">
                            <i class="fa fa-trash"></i>
                        </button>
                    </form>
                </td>
            </tr>

```



**Listing 5.45 Baris program fungsi ajax detail pelanggaran setiap siswa (lanjutan)**

```

        )
    }
    else if ($role_headmaster)
    {
        $('#tbody-student-violation
        -list').append(
            <tr>
                <td style="width: 20px">${obj.id}</td>
                <td style="width: 90px">${obj.FNAME}
                ${obj.LNAME}</td>
                <td style="width: 20px">${obj.DATE}</td>
                <td style="width: 60px">${obj.TYPE} -
                ${obj.NAME}</td>
                <td style="width: 110px">${obj.DESCRPTION}
                </td>
                <td style="width: 110px">${obj.PUNISHMENT}
                </td>
                <td style="width: 40px">${obj.POINT}</td>

            </tr>

        )
    }
}

$('#modalDetailViolationEachStudent').modal('show')
},
error: function(err){
    console.log(err)
}
})
}));

```

### 5.3.19 Daftar Catatan Absen

Halaman daftar catatan absen adalah halaman yang memproses catatan-catatan absen. Halaman ini hanya bisa diakses oleh *user* sebagai kepala sekolah dan guru/wali kelas.

## 1. Daftar Catatan Absen Kepala Sekolah

Data yang akan diberikan kepada sekolah adalah informasi grafik yaitu persentase setiap absensi dan daftar tabel. Daftar tabel untuk kepala sekolah akan memberikan informasi nama kelas, dan total kehadiran setiap kelas per tahun ajarannya ada berapa. Kepala sekolah bisa melihat detail absen dengan menekan tombol *icon* mata yang akan menampilkan modal daftar absen per kelas. Modal daftar absen per kelas ini menggunakan satu fungsi *request* data menggunakan *ajax*. Fungsi *ajax* akan meminta data dari *controller* pada method “absentEachGrade”.

Kepala sekolah juga bisa melihat detail lebih lagi daftar absen per tipe absennya. Untuk menampilkan daftar absen per tipe absen digunakan suatu fungsi *ajax* untuk lakukan *request* data kepada fungsi “detailAbsentEachType”. Baris program fungsi “absentEachGrade” dapat dilihat pada listing 5.46 dan untuk fungsi *ajax* sendiri meminta data daftar absen per kelas dapat dilihat pada listing 5.47. Sedangkan baris program fungsi “detailAbsentEachType” dapat dilihat pada listing 5.48 dan untuk fungsi *ajax* meminta data detail absen per tipe dapat dilihat pada listing 5.49.

### Listing 5.46 Baris program fungsi “absentEachGrade”

```
public function absentEachGrade(Request $request)
{
    $selected_student = GradeStudent::select(DB::raw('MAX
(ACADEMIC_YEAR_ID) AS id'))->limit(1)->first()->id;

    $absen_per_kelas = Absent::join('students',
    'absents.STUDENTS_ID', 'students.id')
    ->join('grades_students', 'grades_students.STUDENTS_ID',
    'students.id')
    ->join('grades', 'grades_students.GRADES_ID', 'grades.id')
```

**Listing 5.46 Baris program fungsi “absentEachGrade” (lanjutan)**

```

->select('absents.TYPE', 'grades.NAME',
        DB::raw('COUNT(*) ABSENPERTIPE'),
        'grades_students.GRADES_ID')
->where('grades_students.GRADES_ID', $request->gradeId)
->where('grades_students.ACADEMIC_YEAR_ID',
        $selected_student)
->where('absents.ACADEMIC_YEAR_ID', $request->academicYearId)
->groupBy('absents.TYPE')
->get();

return $absen_per_kelas;
}

```

**Listing 5.47 Baris program ajax request daftar absen per kelas**

```

$('#tbody-daftar-absen').on('click', '.button-detail-absen-per-
kelas', (function(){
    var gradeId = $(this).attr('grade-id');
    var academicYearId = $('#selector-dropdown-absentrecord-
year').val()

    $.ajax({
        url: '{{ route("absent.absentEachGrade") }}',
        type: 'get',
        data: {gradeId: gradeId,
                academicYearId: academicYearId},

        success: function(result){
            $('#tbody-absent-each-grade-list').empty()
            result.forEach(function(obj){
                $('#tbody-absent-each-grade-list').append(
                    `
                    <tr>
                        <td style="width: 50px">${obj.TYPE}
                        </td>
                        <td style="width: 50px">
                        ${obj.ABSENPERTIPE}</td>
                        <td style="width: 50px">
                            <button type="button"
                                class="btn
                                btn-primary btn-sm
                                button-detail"
                                absent-type=${obj.TYPE}
                                grade-id=${obj.GRADES_ID}>
                                <i class="fa fa-eye"></i>
                            </button>
                        </td>
                    </tr>
                    `
                )
            })
        }
    })
})

```

**Listing 5.47 Baris program ajax request daftar absen per kelas (lanutan)**

```

        })
        $('#modalAbsentEachGrade').modal('show')
    },
    error: function(err){
        console.log(err)
    }
    })
    })
    )))

```

**Listing 5.48 Baris program fungsi “detailAbsentEachType”**

```

public function detailAbsentEachType(Request $request)
{
    $selected_student = GradeStudent::select(DB::raw('MAX
    (ACADEMIC_YEAR_ID) AS id'))->limit(1)->first()->id;

    $detail_absen_per_tipe = Absent::join('academic_years',
    'absents.ACADEMIC_YEAR_ID', 'academic_years.id')
    ->join('students', 'absents.STUDENTS_ID', 'students.id')
    ->join('grades_students', 'grades_students.STUDENTS_ID',
    'students.id')
    ->select('absents.*', 'students.FNAME',
    'students.LNAME',
    'academic_years.TYPE AS TIPETHNAJARAN',
    'academic_years.NAME')
    ->where('absents.TYPE', $request->absentType)
    ->where('absents.ACADEMIC_YEAR_ID',
    $request->academicYearId)
    ->where('grades_students.GRADES_ID', $request->gradeId)
    ->where('grades_students.ACADEMIC_YEAR_ID',
    $selected_student)
    ->get();

    return $detail_absen_per_tipe;
}

```

**Listing 5.49 Baris program ajax request detail absen per tipe absen**

```

$('#tbody-absent-each-grade-list').on('click', '.button-
detail', (function(){
    var absentType = $(this).attr('absent-type')
    var gradeId = $(this).attr('grade-id')
    var academicYearId = $('#selector-dropdown-absentrecord
-year').val()

```

**Listing 5.49** Baris program ajax request detail absen per tipe absen  
(lanjutan)

```
$.ajax({
  url: '{{ route("absent.detailAbsentEachType") }}',
  type: 'get',
  data: {absentType: absentType, gradeId: gradeId,
        academicYearId: academicYearId},

  success: function(result){
    $('#tbody-detail-absent-each-type-
    -list').empty()

    result.forEach(function(obj){
      $('#tbody-detail-absent-each-type-
      -list').append(
        `
        <tr>
          <td style="width: 30px">${obj.id}
          </td>
          <td style="width: 110px">
            ${obj.FNAME}  ${obj.LNAME}</td>
          <td style="width: 80px">${obj.TYPE}
          </td>
          <td style="width: 100px">
            ${obj.TIPETHNAJARAN}-
            ${obj.NAME}</td>
          <td style="width: 75px">
            ${obj.START_DATE}</td>
          <td style="width: 75px">
            ${obj.END_DATE}</td>
          <td style="width: 150px">
            ${obj.DESCRPTION}</td>
          </tr>
        `
      )
    })
    $('#modalDetailAbsentEachType').modal('show')
  },
  error: function(err){
    console.log(err)
  }
})
})
})
```

## 1. Daftar Catatan Absen Guru/Wali Kelas

Data yang akan diberikan wali kelas adalah informasi grafik yaitu persentase setiap absensi yang mendapat data dari *controller* dan daftar tabel. Daftar absen akan menampilkan catatan ketidakhadiran milik siswa kelasnya sendiri.

Ada proses yang digunakan untuk melihat detail daftar absen per tipe yang dimiliki wali kelas. Detail daftar absen per tipe akan ditampilkan dalam sebuah modal dengan menggunakan fungsi *ajax* yang melakukan request detail absen per tipe absen kepada fungsi “detailAbsentEachType” pada absen *controller*. Subbab ini akan memberikan tiga potongan baris program yaitu, listing 5.50 merupakan baris program data pada grafik, listing 5.51 menjelaskan baris program fungsi “detailAbsentEachType” dan listing 5.52 adalah baris program ajax request detail absen per tipe absen.

### Listing 5.50 Baris program informasi catatan absen dengan grafik

```
var types = {!! json_encode($type) !!}
var dataGraph = {!! json_encode($data) !!}
var totalDayEachAcademicYear =
{!! json_encode($count_total_day_each_ay) !!}
var totalAmountExceptPresent = 0
var present_percentage = 0
var dataSeries = []
var obj_present = {}
types.forEach(function(item) {
    dataGraph.forEach(function(obj) {
        if(obj.TIPE == item.TIPE) {
            dataSeries.push({
                name: item.TIPE,
                y: obj.JUMLAH
            })
            totalAmountExceptPresent =
            totalAmountExceptPresent + obj.JUMLAH
            present_percentage = totalDayEachAcademicYear -
            totalAmountExceptPresent
```

**Listing 5.50 Baris program informasi catatan absen dengan grafik (lanjutan)**

```

    }
  })
})
if(Array.isArray(dataGraph) == dataGraph.length){
  obj_present = {name: 'PRESENT', y: present_percentage,
    sliced: true, selected: true}
  dataSeries.push(obj_present)
}
else {
  obj_present = {name: 'PRESENT', y:
    totalDayEachAcademicYear, sliced: true,
    selected: true}
  dataSeries.push(obj_present)
}
Highcharts.setOptions({
  colors: ['#F21402', '#2ECC71 ', '#E4F202 ', '#2874A6']
});
Highcharts.chart('absentChartStatistic', {
  chart: {
    plotBackgroundColor: null,
    plotBorderWidth: null,
    plotShadow: false,
    type: 'pie'
  },
  credits: {
    enabled: false
  },
  title: {
    text: 'PERSENTASE ABSENSI DALAM SATU PERIODE
      BERDASARKAN JENIS ABSENSI'
  },
  tooltip: {
    pointFormat: '{series.name}: <b>
      {point.percentage:.1f}%</b>'
  },
  accessibility: {
    point: {valueSuffix: '%'}
  },
  plotOptions: {
    pie: {
      allowPointSelect: true,
      cursor: 'pointer',
      dataLabels: {
        enabled: false },
      showInLegend: true}
  },
  series: [{
    name: 'Persentase',
    colorByPoint: true,
    data: dataSeries
  }]
});

```

**Listing 5.51 Baris program fungsi “detailAbsentEachType”**

```

public function detailAbsentEachType(Request $request)
{
    $selected_student = GradeStudent::select(DB::raw('MAX
    (ACADEMIC_YEAR_ID) AS id'))->limit(1)->first()->id;

    $detail_absen_per_tipe = Absent::join('academic_years',
    'absents.ACADEMIC_YEAR_ID', 'academic_years.id')
    ->join('students', 'absents.STUDENTS_ID', 'students.id')
    ->join('grades_students', 'grades_students.STUDENTS_ID',
    'students.id')
    ->select('absents.*', 'students.FNAME',
    'students.LNAME',
    'academic_years.TYPE AS TIPETHNAJARAN',
    'academic_years.NAME')
    ->where('absents.TYPE', $request->absentType)
    ->where('absents.ACADEMIC_YEAR_ID', $request->academicYearId)
    ->where('grades_students.GRADES_ID', $request->gradeId)
    ->where('grades_students.ACADEMIC_YEAR_ID',
    $selected_student)
    ->get();

    return $detail_absen_per_tipe;
}

```

**Listing 5.52 Baris program ajax request detail absent per tipe absen**

```

$('#tbody-daftar-absen').on('click', '.button-detail-tipe-
absen', (function(){
    var absentType = $(this).attr('absent-type')
    var gradeId = $(this).attr('grade-id')
    var academicYearId = $('#selector-dropdown-absentrecord
-year').val()
    $.ajax({
        url: '{{ route("absent.detailAbsentEachType") }}',
        type: 'get',
        data: {absentType: absentType, gradeId: gradeId,
        academicYearId: academicYearId},
        success: function(result){
            $('#tbody-detail-absent-each-type
-list').empty()

            result.forEach(function(obj){
                var route = "{{ route('absent.edit', ':id')
                }}"
                route = route.replace(':id', `${obj.id}`)
                var route_del = "{{ route ('absent.destroy',
                ':id' ) }}"
                route_del = route_del.replace(':id',
                `${obj.id}`)
            })
        }
    })
})

```



**Listing 5.52 Baris program ajax request detail absent per tipe absen (lanjutan)**

```

        $('#tbody-detail-absent-each-type
        -list').append(
            `
            <tr>
                <td style="width: 30px">${obj.id}
                </td>
                <td style="width: 110px">
                    ${obj.FNAME}
                    ${obj.LNAME}</td>
                <td style="width: 80px">${obj.TYPE}
                </td>
                <td style="width: 100px">
                    ${obj.TIPETHNAJARAN}
                    - ${obj.NAME}</td>
                <td style="width: 75px">
                    ${obj.START_DATE}</td>
                <td style="width: 75px">
                    ${obj.END_DATE}</td>
                <td style="width: 150px">
                    ${obj.DESCRPTION}</td>
                <td style="width: 70px">
                    <a href=${route} class="btn
                    btn-warning btn-sm">
                        <i class="fa fa-pencil"></i>
                    </a>
                    <form action=${route_del}
                    method="POST" class="inline">
                        @method('delete')
                        @csrf
                        <button class="btn
                        btn-danger btn-sm"
                        onclick="return
                        confirm('Are you sure?')"
                        value="DELETE">
                            <i class="fa
                            fa-trash"></i>
                        </button>
                    </form>
                    </td>
            </tr>
            `
        )
    })
    $('#modalDetailAbsentEachType').modal('show')
},
error: function(err){
    console.log(err)
}
})
)))

```