**BAB 5**

IMPLEMENTASI SISTEM

**BAB 5**

**IMPLEMENTASI SISTEM**

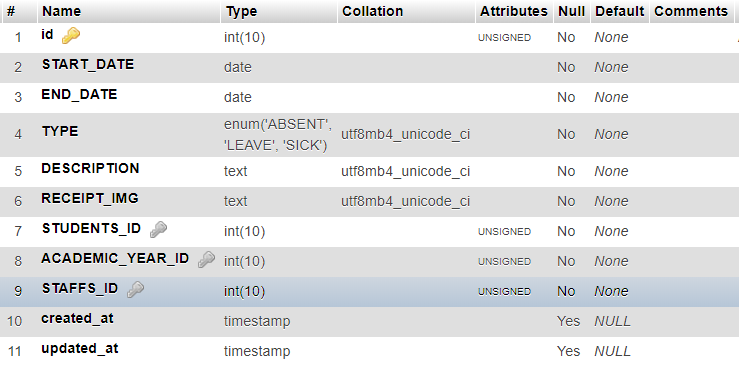
Bab ini membahas tentang implementasi sistem dari rancangan solusi yang sudah direncakan dari bab-bab sebelumnya. Implementasi solusi ini dibangun meliputi database dan bagaimana program ditulis.

**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

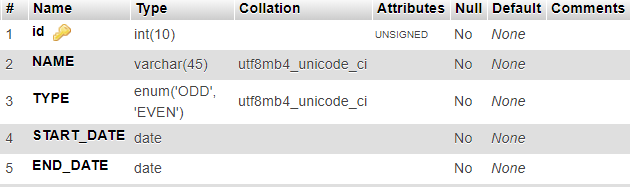
Tabel absents berguna untuk menyimpan catatan absen siswa. Data yang disimpan yaitu tanggal awal absen, tanggal berakhir absen, tipe absen, deskripsi/keterangan absen, bukti keterangan tidak hadir, id siswa, id staff/karyawan yang mencatat, dan akan tersimpan secara otomatis oleh bawaan sistem yaitu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



**Gambar 5.1. Struktur tabel absents**

2. Tabel Academic\_Years

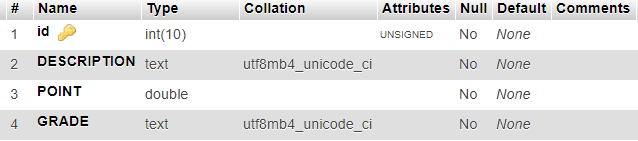
Tabel academic\_years adalah tabel master yang digunakan untuk menyimpan daftar tahun ajaran. Data yang akan disimpan dalam tabel ini adalah nama tahun ajaran, tipe tahun ajaran yaitu ganjil atau genap, tanggal awal dimulai tahun ajaran, tanggal berakhirnya tahun ajaran. Struktur tabel academic\_years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



**Gambar 5.2. Struktur tabel academic years**

3. Table Achievements

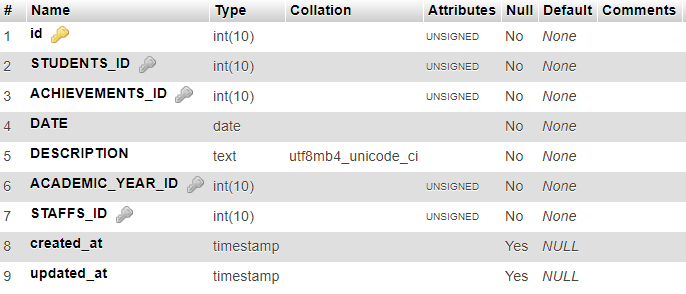
Tabel achievements adalah tabel master yang digunakan untuk menyimpan daftar penghargaan yang berlaku di sekolah. Data yang akan disimpan dalam tabel ini adalah deskripsi, poin, tingkat penghargaan, dan tipe penghargaan. Struktur tabel academic\_years dapat dilihat pada tabel 5.3.



**Gambar 5.3 Struktur tabel achievements**

4. Table Achievement\_Records

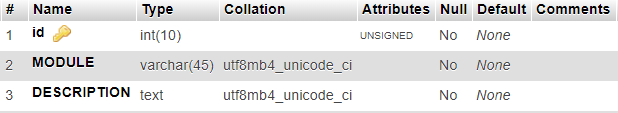
Tabel achievement\_records berguna untuk menyimpan catatan penghargaan yang diperoleh siswa dan detail catatannya. Data yang disimpan pada tabel ini adalah id siswa, id penghargaan, tanggal, deskripsi, id tahun ajaran, id staff/karyawan, *created\_at* dan *updated\_at.* Struktur tabel achievement\_records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



**Gambar 5.4 Struktur tabel achievement records**

5. Tabel Activities

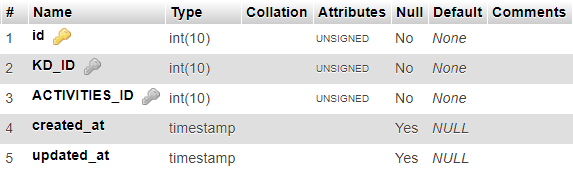
Tabel activities adalah tabel master yang digunakan untuk menyimpan daftar kegiatan/aktivitas pembelajaran yang berlaku di sekolah. Data yang akan disimpan dalam tabel ini adalah module dan deskripsi. Struktur tabel academic\_years dapat dilihat pada tabel 5.4.



**Gambar 5.4 Stuktur tabel activites**

6. Tabel Activities\_KD

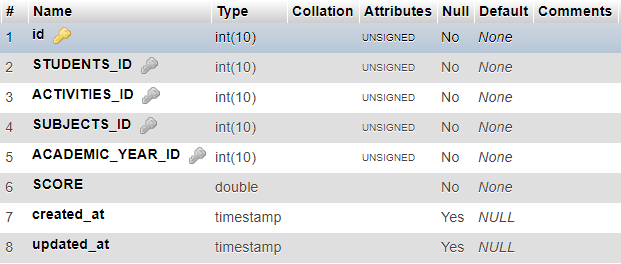
Tabel activites\_kd adalah hubungan antara tabel activities dan tabel kd. Tabel ini berguna untuk menyimpan aktivitas yang dilakukan yang berkaitan dengan satu kompetensi dasar (kd) tertentu. Di dalam tabel ini akan menyimpan data yaitu, id kd, id aktivitas, *created\_at* dan *updated\_at.* Struktur tabel activites kd digambarkan pada gambar 5.5.



**Gambar 5.5 Struktur tabel activities kd**

7. Tabel Activities\_Students

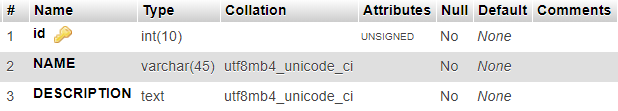
Tabel activites students digunakan untuk menyimpan detail transaksi data pada saat melakukan peng-inputan nilai suatu mata pelajaran oleh *user.* Di dalam tabel ini akan tersimpan data yaitu id siswa sebagai *foreign key,* id aktivitas sebagai *foreign key,* id mata pelajaran sebagai *foreign key,* id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



**Gambar 5.7 Struktur tabel activites students**

8. Tabel Banks

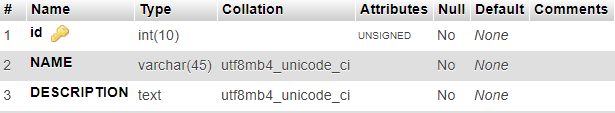
Tabel banks digunakan untuk menampung data master bank. Data yang disimpan dalam tabel ini yaitu nama bank, keterangan bank. Untuk membedakan setiap bank digunakan primary key. Struktur tabel bank dapat dilihat pada tabel 5.8.



**Gambar 5.8 Struktur tabel banks**

9. Tabel Departments

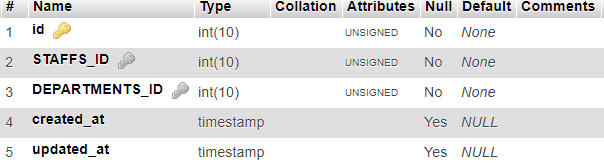
Tabel departments adalah tabel master yang digunakan untuk menyimpan daftar departemen yang berada di sekolah. Tabel ini akan menyimpan data yaitu nama departemen, dan deskripsi dari departemen. Struktur tabel ini bisa dilihat pada gambar 5.9.



**Gambar 5.9 Struktur tabel departments**

10. Tabel Departments\_Staffs

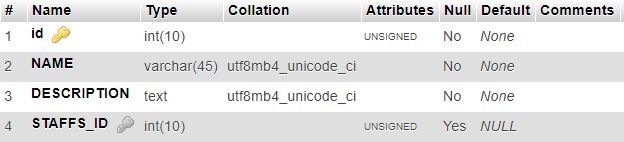
Tabel departments\_staffs adalah tabel hasil relasi *many to many* antara tabel departmens dengan tabel staffs. Tabel ini digunakan sebagai penyimpan data id staff dan id departemen. Selain itu juga disimpan tanggal transaksi kapan hubungan antara satu id staff dan satu id departemen terbentuk yaitu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



**Gambar 5.10 Struktur tabel departments staffs**

11. Tabel Extracurriculars

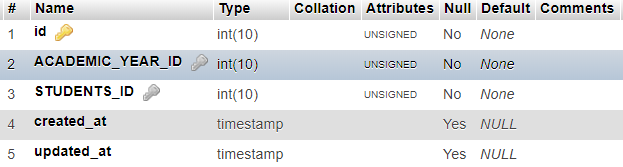
Tabel extracurriculars adalah tabel master yang menyimpan daftar ekstrakurikuler yang berada di sekolah. Tabel ini menyimpan data seperti nama ekskul, deskripsi ekskul, dan id staff sebagai *foreign key* yang menandakan guru pengajar ekskul tersebut. Struktur tabel extracurriculars digambarkan seperti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



**Gambar 5.11 Struktur tabel extracurriculars**

12. Tabel Extracurricular\_Records

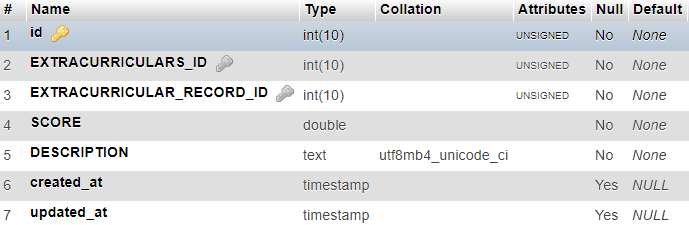
Tabel extracurricular\_records adalah tabel yang digunakan untuk menyimpan catatan ekstrakurikuler setiap siswa pada tahun ajaran tertentu. Tabel ini akan menyimpan data yakni id tahun ajaran sebagai *foreign key¸*dan juga id siswa sebagai *foreign key, created\_at* dan *updated\_at.* Struktur tabel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



**Gambar 5.12 Struktur tabel extracurricular records**

13. Tabel Extracurricular\_Reports

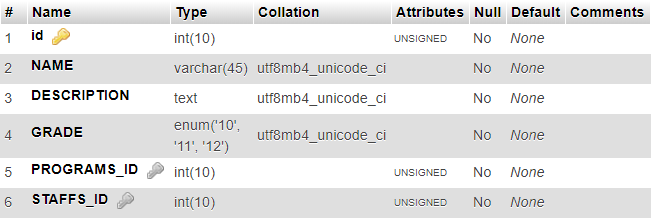
Tabel extracurricular\_reports merupakan tabel hasil relasi *many to many* antara tabel extracurricular\_records dan tabel extracurriculars. Tabel ini digunakan untuk menyimpan transaksi detail setiap mata ekskul yang terdapat dalam satu id catatan ekstrakurikuler pada tabel extracurricular\_records. 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.



**Gambar 5.13 Struktur tabel extracurricular reports**

14. Tabel Grades

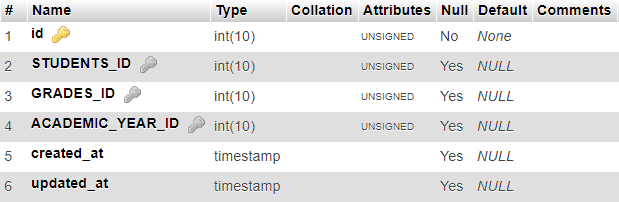
Tabel grades digunakan untuk menyimpan daftar nama kelas. Data yang dicatat yaitu nama kelas, keterangan kelas, tingkat kelas, id unik dari tabel program sebagai *foreign key*, id unik dari tabel staff sebagai *foreign key*. Struktur tabel grades digambarkan seperti pada gambar 5.14.



**Gambar 5.14 Struktur tabel grades**

15. Tabel Grades\_Students

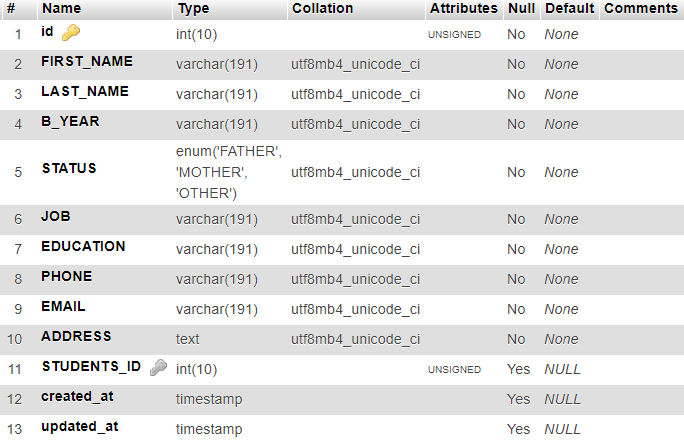
Tabel grades\_students adalah tabel hasil relasi antara tabel grades dan tabel students. Tabel ini untuk mencatat transaksi detail antara siswa dan kelas. Hal ini digunakan untuk memetakan bahwa satu siswa memiliki satu kelas disetiap tahun ajaranya. Artinya siswa tidak terus menerus menetap pada kelas 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.



**Gambar 5.15 Struktur tabel grades students**

16. Tabel Guardians

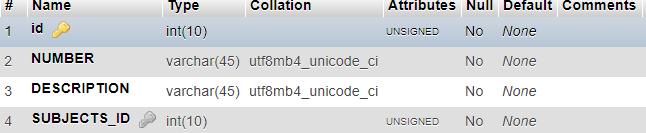
Tabel guardians adalah tabel yang digunakan untuk menyimpan data wali siswa. Setiap siswa memiliki perwakilan yang datanya harus disimpan. Adapun data yang disimpan pada tabel ini adalah nama depan wali siswa, nama akhir wali siswa, tanggal lahir wali siswa, status wali siswa apakah ayah/ibu/lainnya, 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.



**Gambar 5.16 Struktur tabel guardians**

17. Tabel KD

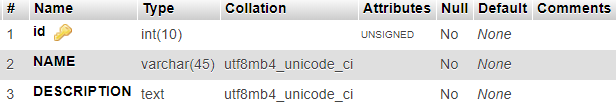
Tabel kd adalah tabel master yang berguna untuk menyimpan daftar kompetensi dasar. Tabel ini akan menyimpan nomor kompetensi dasar, deskripsi kd, dan juga id mata pelajaran sebagai *foreign key.* Struktur tabel kd digambarkan seperti pada gambar 5.17.



**Gambar 5.17 Struktur tabel kd**

18. Tabel Programs

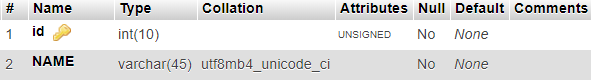
Tabel programs digunakan untuk menampung data mengenai program studi disekolah. Data yang disimpan yaitu nama program studi, keterangan program studi. Struktur tabel programs dapat dilihat pada tabel 5.9.



**Gambar 5.9 Struktur tabel programs**

19. Tabel Religions

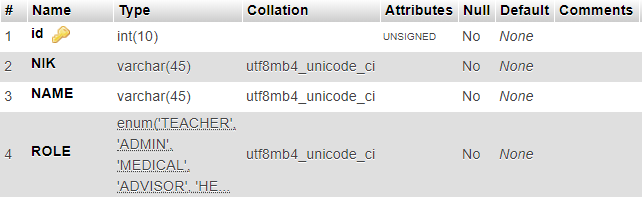
Tabel religion merupakan tabel master untuk menampung data mengenai agama. Data yang akan dicatat dalam tabel ini adalah nama agama. Struktur tabel religions dapat dilihat pada tabel 5.3.



**Gambar 5.19 Struktur tabel religions**

20. Tabel Staffs

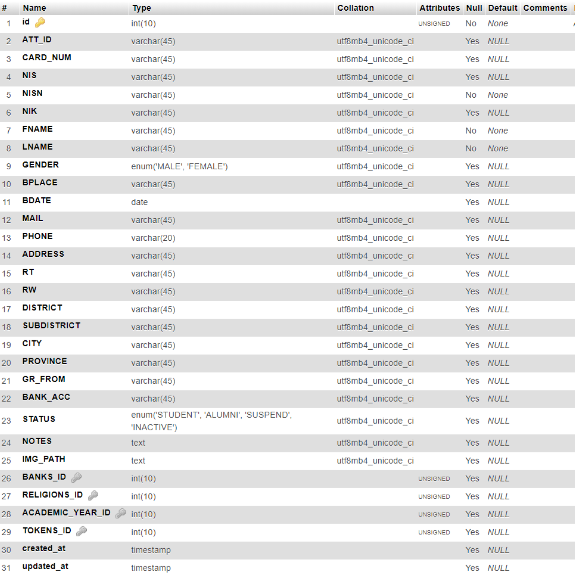
Tabel staffs adalah tabel yang digunakan untuk menyimpan data karyawan/staff yang bekerja di sekolah. Tabel ini akan menyimpan data seperti nomor induk kepegawaian (NIK), nama lengkap staff, hak akses/*role* staff yaitu apakah terdaftar sebagai guru atau admin atau petugas uks atau guru BK atau kepala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



**Gambar 5.20 Struktur tabel staffs**

21. Tabel Students

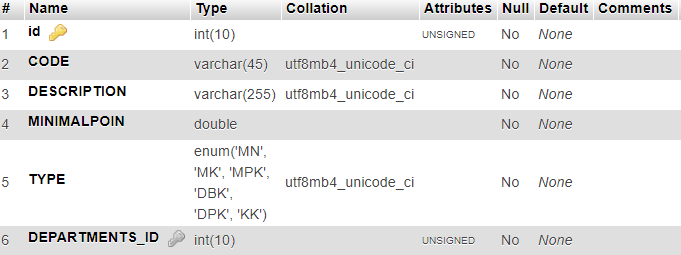
Tabel students adalah tabel yang digunakan untuk menyimpan data siswa. Data siswa yang akan disimpan yaitu atribut id siswa, nomor kartu pelajar, nomor induk sekolah (NIS), nomor induk siswa nasional (NISN), nomor induk kependudukan jika ada (NIK), nama depan siswa, nama belakang siswa, jenis kelamin, 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 keberapa siswa tersebut terdaftar, id token sebagai *foreign key*, *created\_at* dan *updated\_at.* Struktur tabel siswa digambarkan seperti pada gambar 5.21.



**Gambar 5.21 Stuktur tabel students**

22. Tabel Subjects

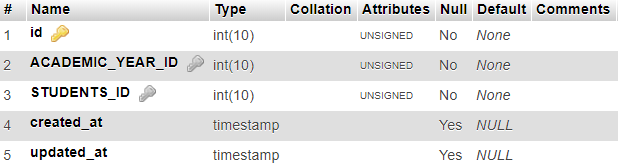
Tabel subjects adalah tabel yang digunakan untuk menyimpan daftar mata pelajaran sesuai departemen yang ada disekolah. Data yang akan dicatat dalam tabel ini yaitu kode mata pelajaran, deskripsi mata pelajaran, nilai kkm, tipe mata pelajaran, id departemen sebagai *foreign key*. Struktur data tabel subjects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



**Gambar 5.22 Struktur tabel subjects**

23. Tabel Subject\_Records

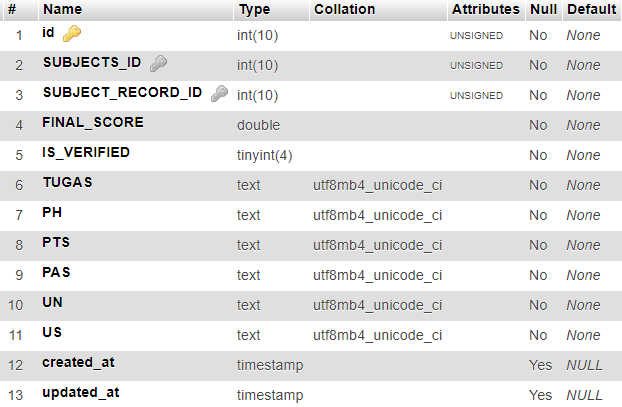
Tabel subject\_records adalah tabel yang menyimpan catatan mata pelajaran setiap satu id siswa dengan id tahun ajaran tertentu. Data yang tersimpan yaitu id tahun ajaran sebagai *foreign key,* id siswa sebagai *foreign key, created at* dan *updated at*. Struktur tabel ini digambarkan seperti pada gambar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



**Gambar 5.23 Stuktur tabel subject records**

24. Tabel Subject\_Reports

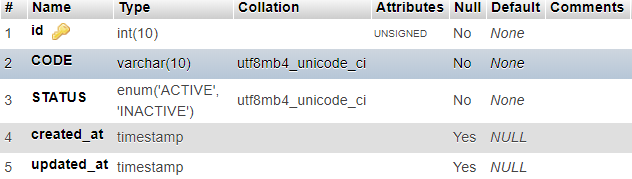
Tabel subject\_reports merupakan tabel hasil relasi *many to many* antara tabel subject\_records dan tabel subjects. Tabel ini digunakan untuk menyimpan transaksi detail setiap mata pelajaran yang terdapat dalam satu id catatan mata pelajaran. Data yang akan dicatat dalam tabel ini adalah id unik dari 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.



**Gambar 5.24 Struktur tabel subject reports**

25. Tabel Tokens

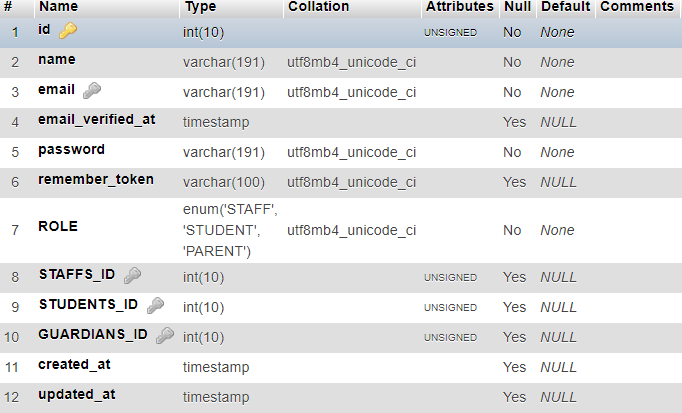
Tabel tokens digunakan untuk menyimpan nomor token setiap kali siswa terdaftar. Data yang disimpan adalah kode token, status token apakah aktif atau tidak, *created\_at* dan *updated\_at.* Struktur tabel tokens digambarkan seperti pada gambar 5.25.



**Gambar 5.25 Struktur tabel tokens**

26. Tabel Users

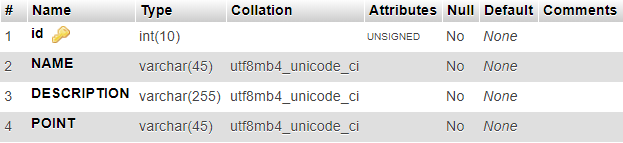
Tabel users berguna untuk menyimpan data-data user yang terdaftar pada sistem. Tabel users menyimpan data nama lengkap user, email user sebagai *key* yang unik agar user satu dengan yang lain tidak bisa memiliki email yang sama untuk login. Kemudian juga menyimpan tanggal email terverifikasi, *password* 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.



**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.

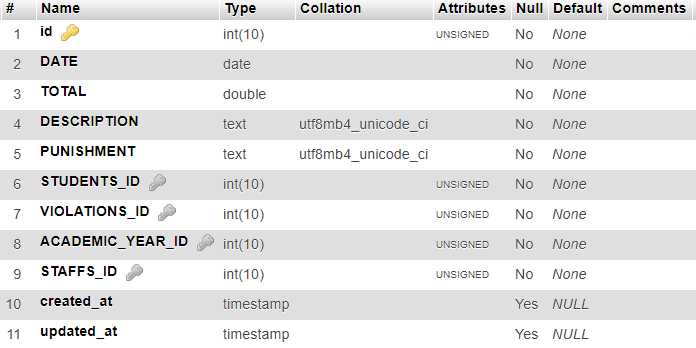


**Gambar 5.27 Struktur tabel violations**

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 sanski, 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.

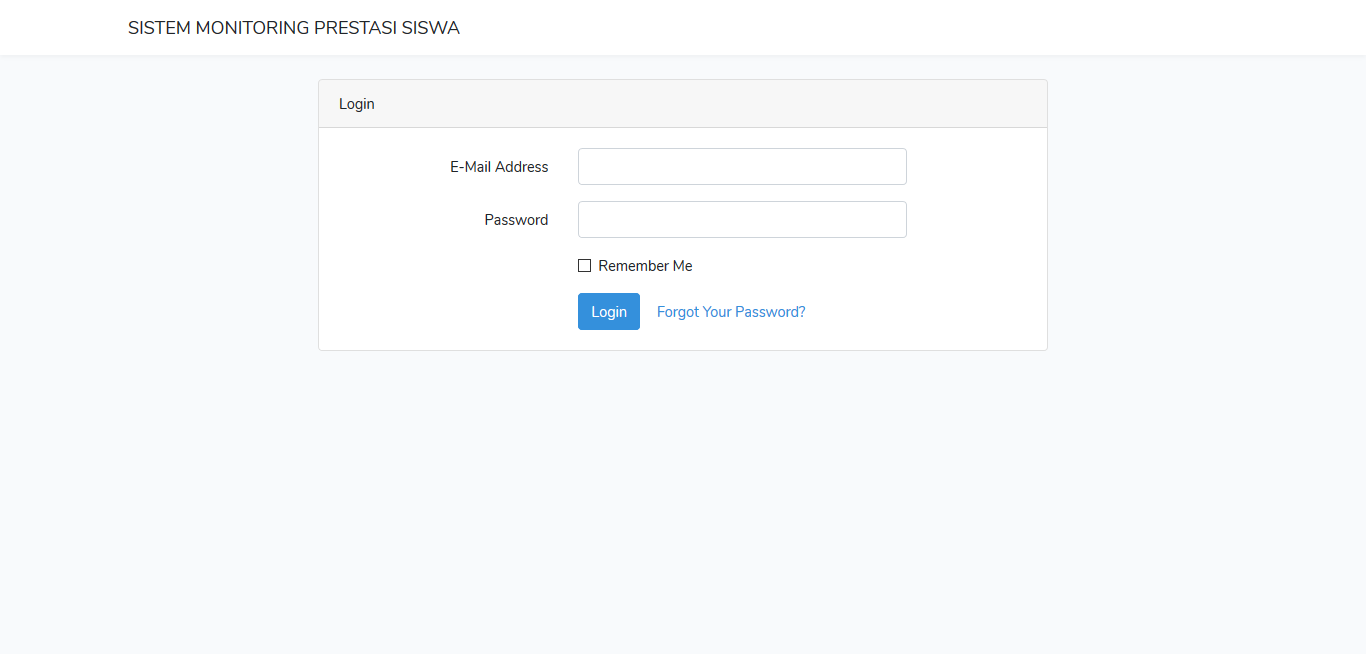


**Gambar 5.28 Struktur tabel violation records**

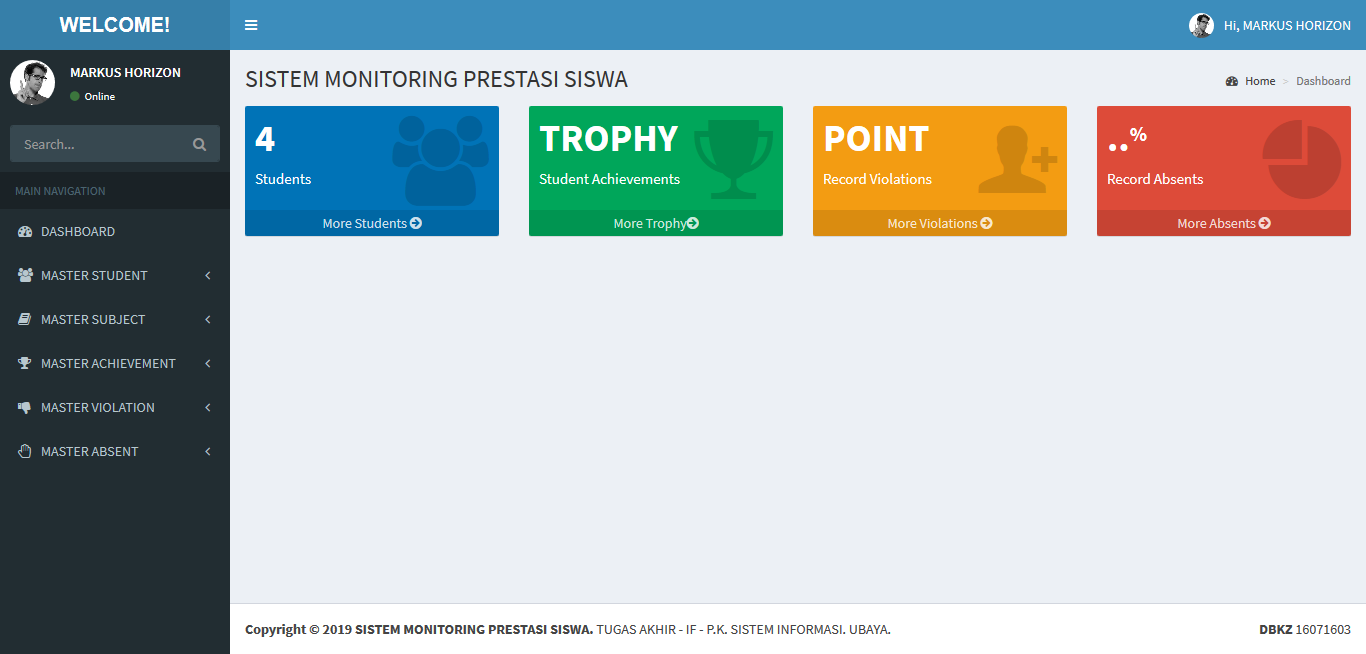
**5.2 Implementasi Tampilan**

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.

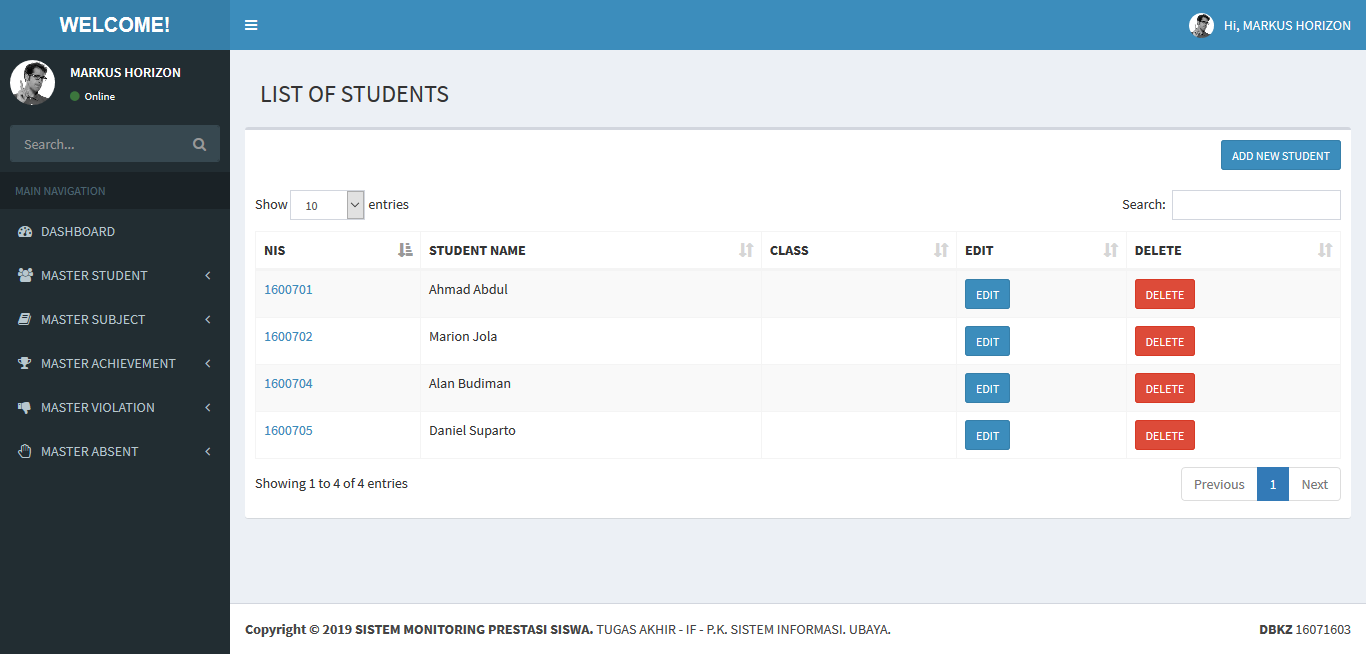
* **Halaman Login**



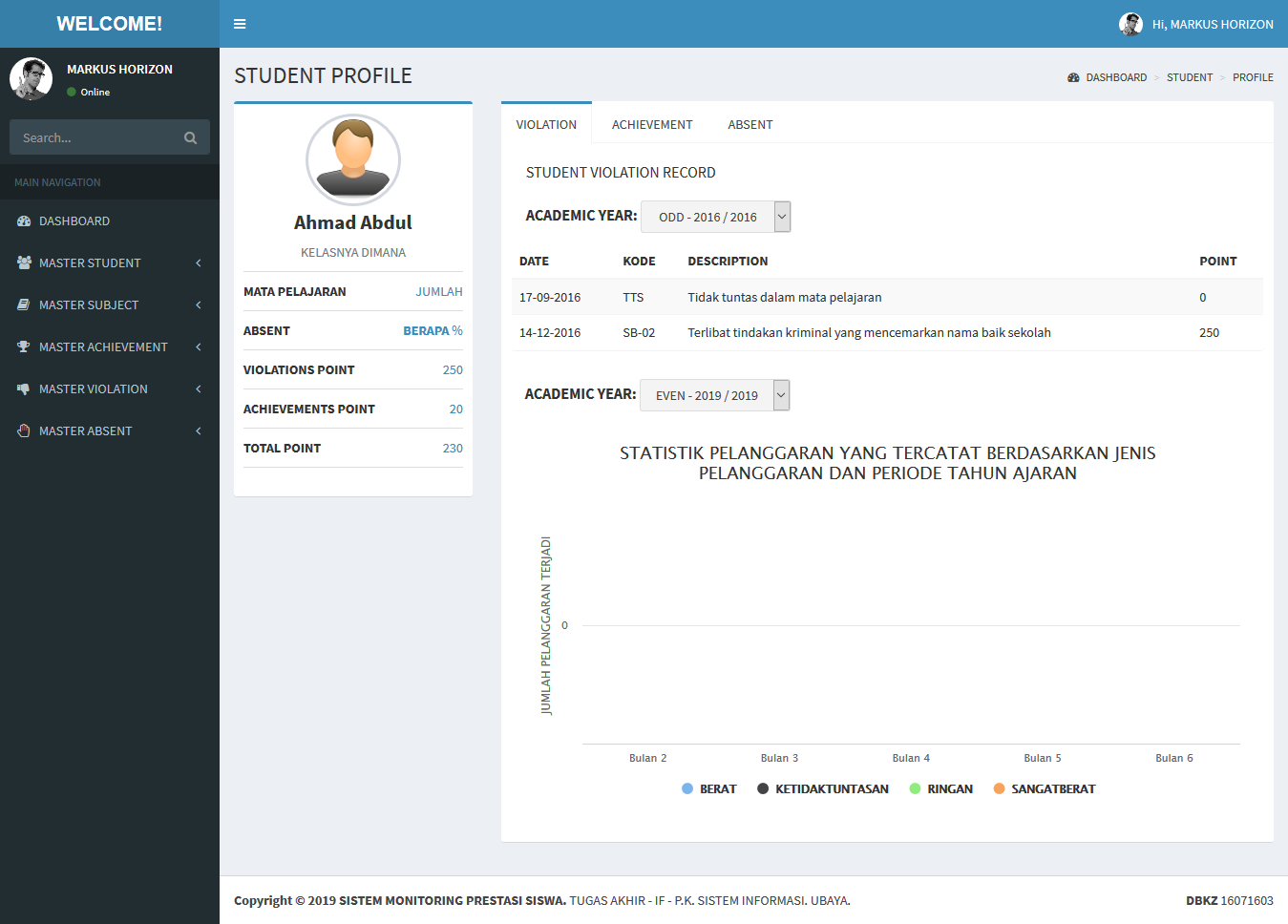
* **Halaman Dashboard**



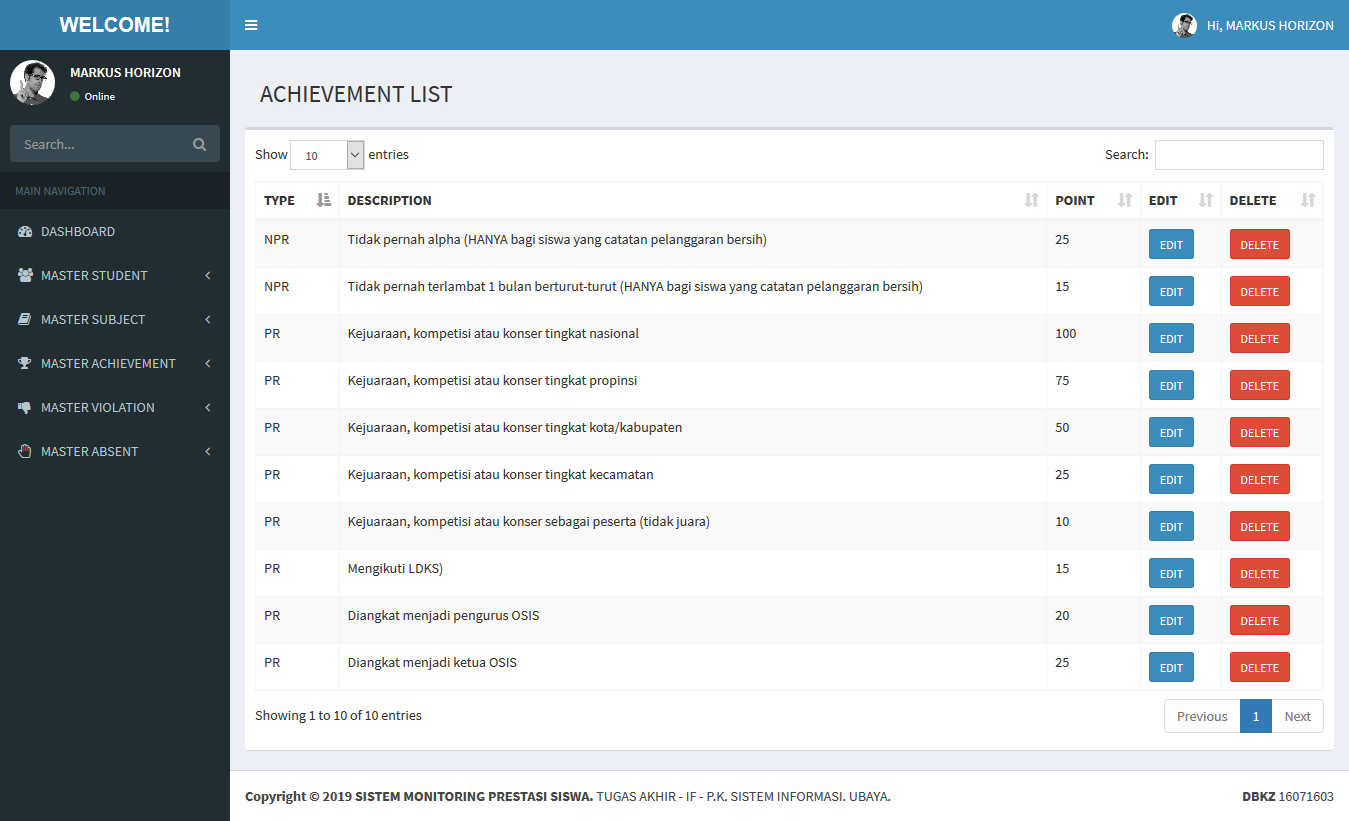
* **Halaman Daftar Siswa**



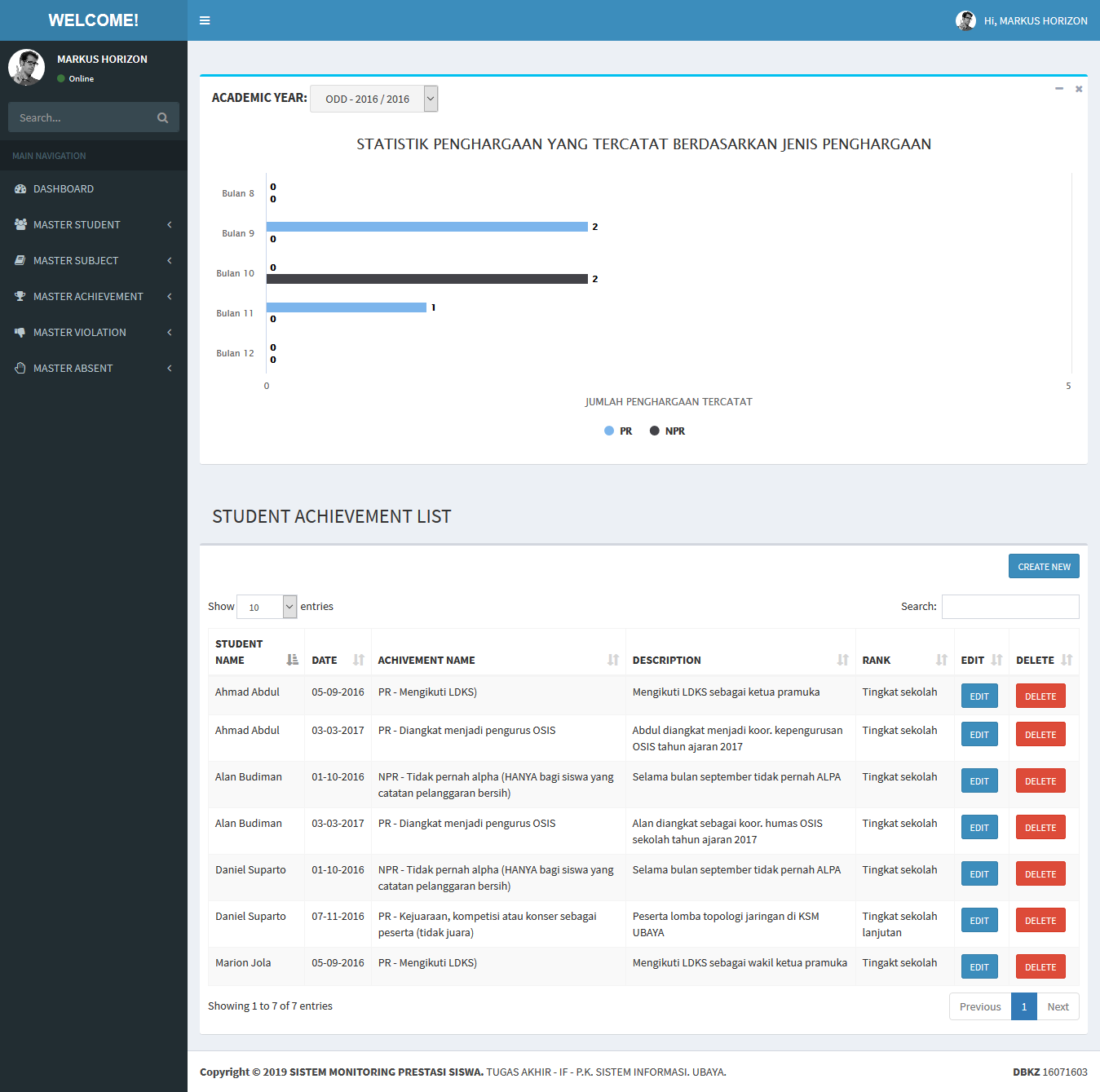
* **Halaman Profil Siswa**



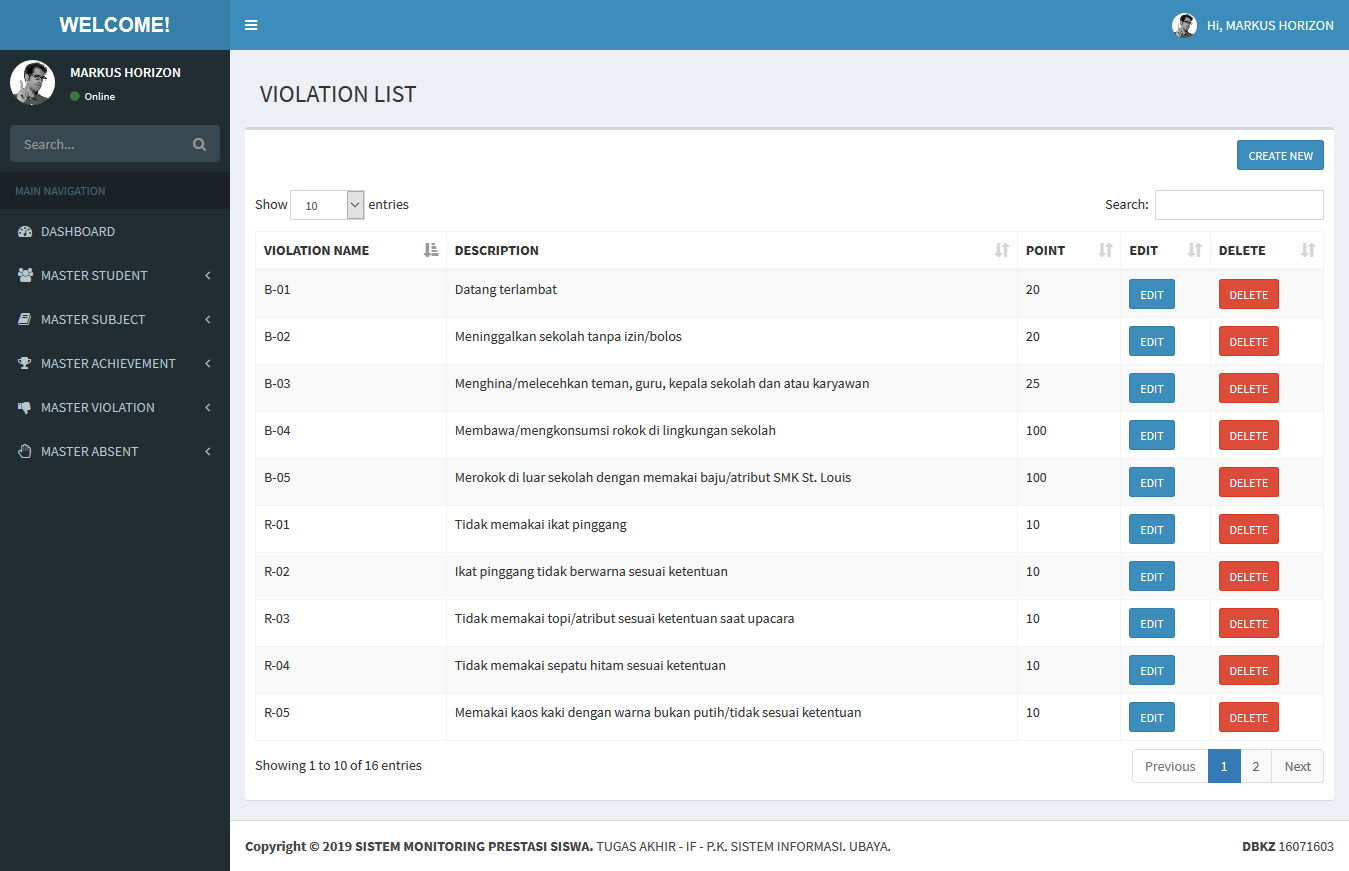
* **Halaman Daftar Penghargaan**



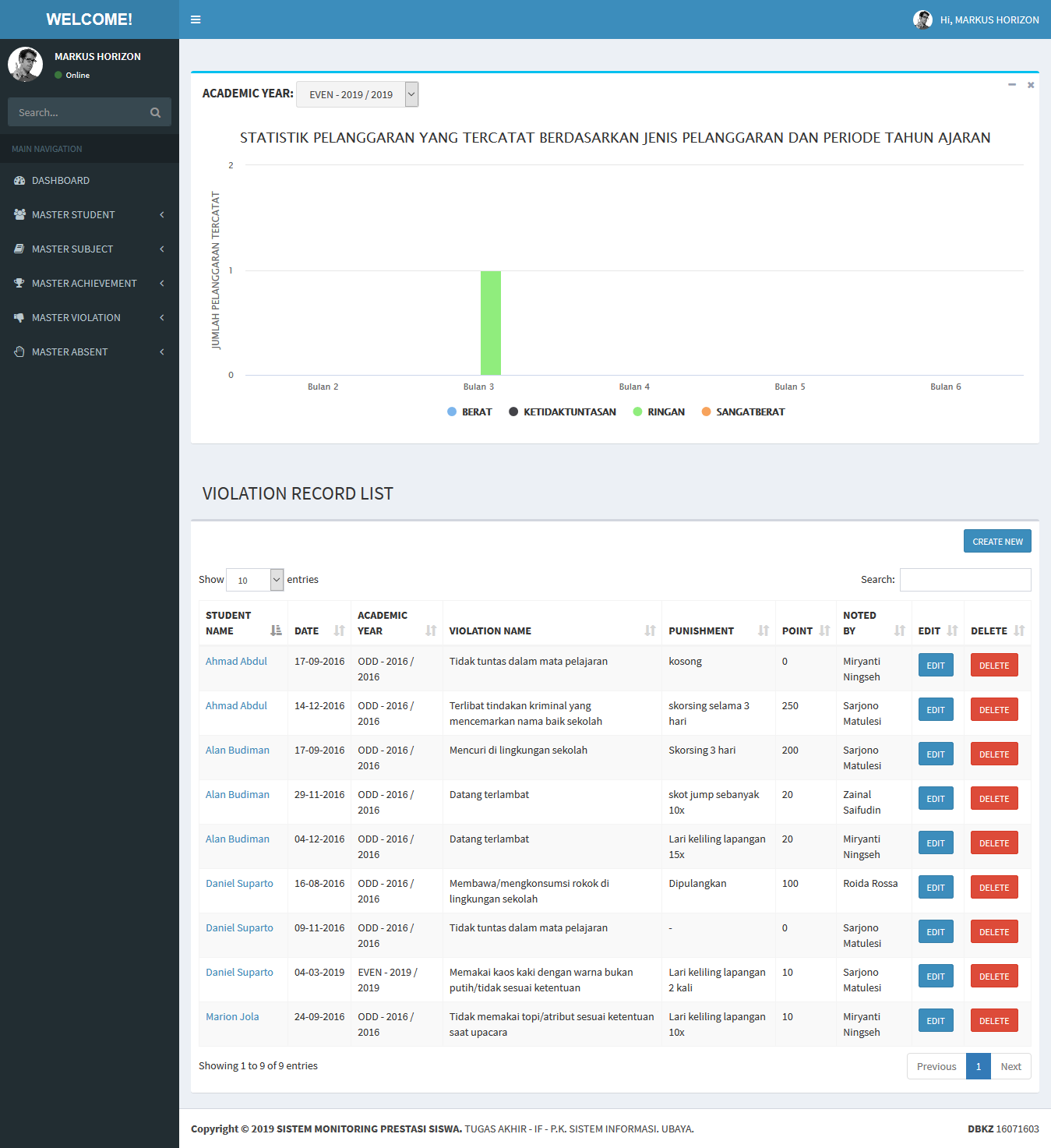
* **Halaman Daftar Catatan Penghargaan**



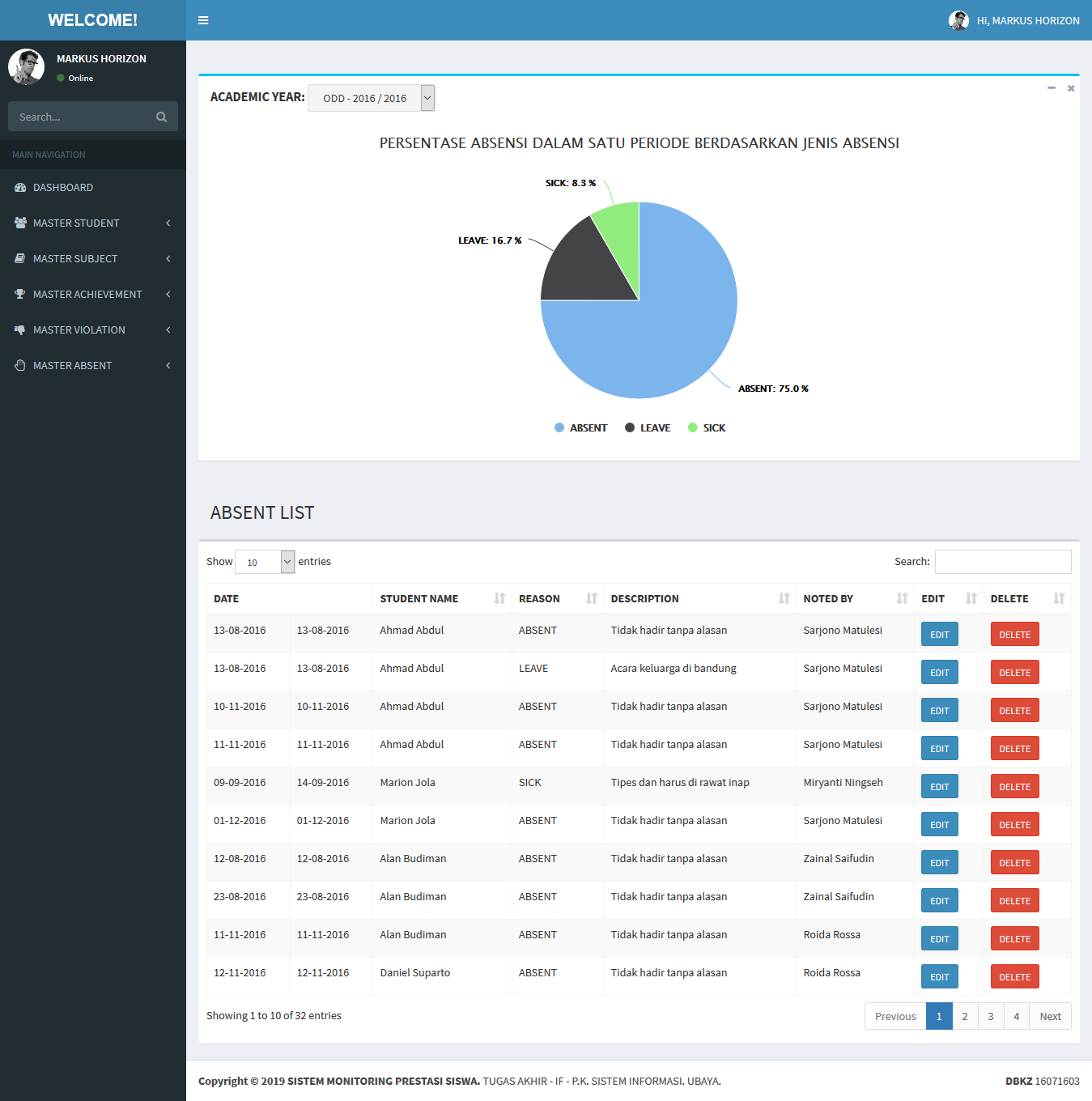
* **Halaman Daftar Pelanggaran**



* **Halaman Daftar Catatan Pelanggaran**



* **Halaman Daftar Absen**



**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 memporses data. Demikian juga pada implementasi program diperlukan potongan kode JQuery dan Javascript dalam menerima data.

**5.3.1 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 disekolah. Jika *user* login sebagai guru/wali kelas dapat melihat daftar siswa kelasnya saja. Baris program daftar siswa dapat dilihat pada listing 5.1.

**Lisitng 5.1 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'));  } |

**5.3.2 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 5.2 menunjukan baris program import *controller*, sedangkan listing 5.3 menunjukan baris program *function* collection.

**Listing 5.2 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 5.3 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] |

**Listing 5.3 Baris program function collection input data ke database (lanjutan)**

|  |
| --- |
| $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")  ]);    $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  ]);    }  }  }  } |

**5.3.3 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 5.4.

**Listing 5.4 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) |

**Listing 5.4 Baris program function profile**  **(lanjutan)**

|  |
| --- |
| ->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();          $x = new DashboardController();          $kehadiran =  $x->showAbsentKu($siswa->id);          $selected\_student\_ay = Student::select('ACADEMIC\_YEAR\_ID AS AY\_ID')->where('id', $siswa->id) ->first()->AY\_ID;          $tahun\_ajaran = AcademicYear::where('id', '>=', $selected\_student\_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;          $achievement\_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(); |

**Listing 5.4 Baris program function profile**  **(lanjutan)**

|  |
| --- |
| $maxId = AcademicYear::select(DB::raw('MAX(id) as id'))->get()[0]->id;          if($request->has('academicYearId')){              $academic\_year\_id = $request->academicYearId;          }          else{              $academic\_year\_id = $maxId;          }          $selected\_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];  *// 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.VIOLATIONS\_ID = v.id                                  WHERE v.NAME NOT LIKE 'TTS%'                                  GROUP BY KATEGORI ");          $data = ViolationRecord::join('violations', 'violation\_records.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(); |

**Listing 5.4 Baris program function profile**  **(lanjutan)**

|  |
| --- |
| *// GRAFIK TAB ACHIEVEMENT*          $type = Achievement::select(DB::raw('GRADE AS TINGKAT'))                          ->groupBy('TINGKAT')                          ->get();          $dataAchievement = DB::select("SELECT a.GRADE AS TINGKAT, MONTH(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 STUDENTS\_ID = " . $id . "                                         GROUP BY TINGKAT, BULAN                                         ORDER BY BULAN ASC");  *// GRAFIK TAB ABSENT*          $tipeAbsen = 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 JUMLAH'))                          ->where('ACADEMIC\_YEAR\_ID', $academic\_year\_id)                          ->where('STUDENTS\_ID', $id)                          ->groupBy('TIPE', 'TAHUNAJARAN')                          ->get();          $count\_total\_day\_each\_ay = AcademicYear::select(DB::raw('DATEDIFF(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\_grade\_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', 'catatan\_absen', 'kehadiran', 'tahun\_ajaran',                                                     'catatan\_pelanggaran', 'violation\_point',                                                     'catatan\_penghargaan', 'achievement\_point', |

**Listing 5.4 Baris program function profile**  **(lanjutan)**

|  |
| --- |
| 'kategori', 'data', 'selected\_tahun\_ajaran', 'academic\_year\_id',                                                     'type', 'dataAchievement',                                                     'tipeAbsen', 'dataAbsen', 'count\_total\_day\_each\_ay')); |

**5.3.4 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 5.5.

**Lisiting 5.5 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'));  } |