

# **SISTEM SKORSING SISWA**

Mata Kuliah: Pemrograman Berorientasi Objek

Dosen Pengampu: Ledy Elsera Astrianty, S.Kom., M.Kom.



## **Kelompok 4:**

**Calvin Darmawan Harun | 5230411006**

**Dhafa Widy Pratama | 5230411007**

**Taufik Nur Ardiansyah | 5230411040**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS TEKNOLOGI YOGYAKARTA  
2024**

## DAFTAR ISI

<b>DAFTAR ISI</b> .....	2
<b>BAB I PENDAHULUAN</b> .....	3
<b>1.1 Deskripsi Sistem</b> .....	3
<b>1.2 Algoritma dan Skenario Sistem</b> .....	4
<b>1.3 Class Diagram Sistem</b> .....	8
<b>BAB II PEMBAHASAN</b> .....	9
<b>2.1 Implementasi dan Hasil</b> .....	9
2.1.1 Menu Utama Sistem .....	9
2.1.2 Menu Tampilkan Siswa .....	11
2.1.3 Menu Tampilkan Hukuman.....	12
2.1.4 Menu Tampilkan Kategori Hukuman.....	14
2.1.5 Menu Tampilkan Histori Skorsing .....	15
2.1.6 Menu Skorsing Siswa .....	17
2.1.7 Menu Tambah Siswa .....	19
2.1.8 Menu Hapus Data Siswa.....	20
2.1.9 Menu Tambah Hukuman .....	21
2.1.10 Menu Tambah Kategori Hukuman .....	22
2.1.11 Menu Ubah Skor Siswa .....	23
2.1.12 Menu Keluar .....	24
<b>2.2 Source Code dan Database Sistem</b> .....	25
2.2.1 Souce Code Sistem .....	25
2.2.2 Database Sistem.....	28
<b>BAB III PENUTUP</b> .....	29
<b>3.1 Kesimpulan</b> .....	29
<b>PEMBAGIAN TUGAS</b> .....	30
<b>DAFTAR PUSTAKA</b> .....	31
<b>LAMPIRAN</b> .....	32

# **BAB I**

## **PENDAHULUAN**

### **1.1 Deskripsi Sistem**

Sistem Skorsing Siswa ini adalah aplikasi berbasis Python yang terintegrasi dengan database MySQL untuk mengelola pelanggaran dan hukuman siswa di lingkungan sekolah. Sistem ini memungkinkan administrator sekolah untuk mencatat, memantau, dan mengelola poin pelanggaran siswa. Sekolah ini menerapkan sistem point pelanggaran yaitu jika ada siswa/siswi yang melanggar peraturan sekolah maka akan terkena sanksi berupa point, dimana jika point sudah mencapai jumlah tertentu, maka akan diberikan sanksi sesuai kriteria yang sudah ditentukan. Adapun jenis sanksi tersebut terdapat 5 jenis sanksi yang diterapkan, yaitu:

1. Jika point siswa sudah mencapai 25 point maka siswa mendapatkan teguran langsung dari bagian Kesiswaan.
2. Jika point siswa sudah mencapai 50 point, siswa akan mendapatkan Surat Peringatan 1 dan hukuman Skorsing Ringan.
3. Jika point siswa sudah mencapai 70 point, siswa akan mendapatkan Surat Peringatan 2 dan hukuman Skorsing Sedang.
4. Jika point siswa sudah mencapai 80 point, siswa akan mendapatkan Surat Peringatan 3 dan hukuman Skorsing Berat.
5. Jika point siswa sudah mencapai 100 point, siswa akan di keluarkan dari Sekolah dan status dari siswa yang semula “aktif” akan berubah menjadi “tidak aktif”

Terdapat 5 kategori pelanggaran, dimana setiap kategorinya akan memiliki poin yang berbeda-beda pula:

#### **1. Kategori Sangat Ringan:**

Pada golongan ini siswa diberikan 5 poin jika melakukan hukuman. Jenis pelanggaran yang termasuk golongan pelanggaran sangat ringan yaitu sebagai berikut:

- Terlambat masuk sekolah > 15 menit
- Tidak memakai seragam sekolah
- Membawa handphone
- Membolos

#### **2. Kategori Ringan:**

Untuk pelanggaran yang tergolong ringan, siswa akan dikenakan penalti sebesar 10 poin. Pelanggaran-pelanggaran yang termasuk dalam kategori ringan adalah sebagai berikut:

- Meninggalkan kelas tanpa izin
- Mengecat Rambut
- Menyontek saat ulangan
- Memakai pakaian tidak sopan

#### **3. Kategori Sedang:**

Siswa yang melakukan pelanggaran sedang, akan mendapatkan penambahan poin sebanyak 25 poin. Berikut adalah daftar pelanggaran yang termasuk dalam kategori sedang:

- Merokok di lingkungan sekolah
- Melakukan vandalisme di lingkungan sekolah

- Berkelahi di lingkungan sekolah
- Merusak fasilitas sekolah

#### 4. Kategori Berat

Siswa akan mendapatkan penambahan poin sebanyak 55 poin jika melakukan pelanggaran pada kategori ini. Berikut beberapa daftar pelanggaran:

- Melakukan Bullying fisik maupun non fisik
- Berjudi dan sejenisnya
- Membuat keributan di lingkungan sekolah
- Kedapatan mencuri
- Membawa senjata tajam ke lingkungan sekolah tanpa izin

#### 5. Kategori Sangat Berat

Siswa yang melakukan pelanggaran pada kategori sangat berat akan langsung mendapatkan penambahan poin sebesar 100 poin. Berikut beberapa pelanggaran yang terdapat pada kategori ini:

- Melakukan tindak asusila
- Membawa atau terlibat penyalahgunaan narkoba
- Membawa atau terlibat penyalahgunaan minuman keras

## 1.2 Algoritma dan Skenario Sistem

Berikut beberapa skenario penggunaan pada sistem ini, yang mencakup berbagai situasi yang mungkin terjadi pada sistem skorsing:

### 1. Tampilkan Siswa: Pengguna sistem dapat melihat ID, nama, NIS, poin, dan status dari siswa yang tersimpan pada sistem.

Skenario penggunaan sistem:

1. Memilih pilihan (1. Tampilkan Siswa), pilihan ini akan menampilkan semua siswa yang terdaftar.
2. Sistem akan menampilkan daftar semua siswa yang tersimpan pada sistem, meliputi ID, nama siswa, NIS, poin, dan status.
3. Jika tidak terdapat data siswa yang tersimpan pada sistem, akan muncul pesan (Tidak ada siswa yang terdaftar)

### 2. Tampilkan Hukuman: Pengguna dapat melihat hukuman apa saja yang ada di dalam sistem, dimana setiap hukuman ini digolongkan berdasarkan kategori nya.

Skenario penggunaan sistem:

1. Memilih pilihan (2. Tampilkan Hukuman), pilihan ini akan menampilkan semua hukuman yang terdapat pada sistem.
2. Sistem akan menampilkan daftar semua hukuman yang tersimpan pada sistem, meliputi ID, nama hukuman, skor dan kategori hukuman.

### 3. Tampilkan Kategori Hukuman: Pengguna dapat melihat setiap kategori hukuman apa saja yang terdapat di dalam sistem beserta skor nya.

Skenario penggunaan sistem:

1. Memilih pilihan (3. Tampilkan Kategori Hukuman), pilihan ini akan menampilkan semua hukuman yang terdapat pada sistem.
2. Sistem akan menampilkan id\_kategori, nama\_kategori, dan skor dari tiap kategori.

### 4. Tampilkan Histori Skorsing: User dapat menampilkan dan melihat histori skorsing yang di miliki oleh siswa berdasarkan ID.

Skenario penggunaan sistem:

1. Memilih pilihan (4. Tampilkan Histori Siswa), pilihan ini akan menampilkan histori skorsing dari siswa yang di pilih berdasarkan ID.
  2. Akan muncul tampilan berupa data semua siswa berikut ID nya.
  3. Selanjutnya user akan diminta memasukan ID siswa yang akan di lihat histori skorsing nya.
  4. Jika ID dari siswa yang akan di lihat histori nya ada di dalam sistem maka akan memberikan output semua histori skorsing yang dimiliki oleh siswa dengan ID yang dipilih.
  5. Jika ID siswa yang akan di lihat histori nya tidak ada di dalam sistem maka akan memberikan output "Siswa dengan ID {siswa\_id} tidak ditemukan."
  6. Jika ID siswa yang di masukan tidak memiliki histori skorsing maka akan memberikan Output "Tidak ada histori skorsing."
- 5. Skorsing Siswa:** Seorang siswa melanggar peraturan sekolah, misalnya, menyontek saat ujian. Administrator akan memilih siswa tersebut dari daftar dan memberikan hukuman yang sesuai. Sistem akan menambahkan skor pelanggaran ke poin siswa, menyimpan riwayat pelanggaran ke dalam histori skorsing.

Skenario Penggunaan Sistem:

1. Memilih pilihan (5. Skorsing Siswa), pilihan ini akan memberikan skorsing pada siswa.
  2. Lalu akan muncul semua data siswa yang terdapat pada sistem beserta ID siswa di dalamnya.
  3. Selanjutnya Admin diminta memasukan ID siswa yang akan di berikan skorsing.
  4. Jika ID siswa yang dipilih ber status "tidak aktif" maka akan muncul pesan (Siswa '{siswa[1]}' sudah tidak aktif)
  5. Namun jika ID siswa yang dipilih tidak terdapat pada sistem akan muncul pesan (Siswa dengan ID {siswa\_id} tidak ditemukan)
  6. Jika pemilihan siswa berhasil dilakukan, sistem akan menampilkan semua hukuman yang terdapat pada sistem beserta kategori dan skor nya, lalu Admin diminta memasukkan ID hukuman yang akan diberikan kepada siswa yang dipilih pada langkah ke-3.
  7. Apabila ID hukuman yang dipilih tidak terdapat pada sistem maka akan muncul pesan (Hukuman dengan ID {hukuman\_id} tidak ditemukan)
  8. Jika proses skorsing gagal dilakukan, sistem akan menampilkan pesan (Kesalahan saat memproses skorsing: {err})
  9. Jika proses skorsing berhasil dilakukan, maka akan muncul pesan (Siswa '{nama\_siswa}' dikenai hukuman '{nama\_hukuman}' dengan skor +{skor\_hukuman}. Poin sekarang: {poin\_baru})
  10. Setelah proses skorsing berhasil dilakukan, sistem akan melakukan pengecekan jumlah skor berdasarkan ID siswa yang dipilih pada langkah ke-3, jika siswa sudah mencapai skor 100 status dari siswa akan berubah menjadi "tidak aktif" dan akan muncul pesan (Siswa '{nama\_siswa}' telah dikeluarkan karena poin mencapai 100)
- 6. Tambah Siswa:** Admin dapat menambahkan siswa yang akan di masukan ke dalam sistem. Admin menambahkan nama dan NIS dari siswa yang akan di tambahkan lalu

poin akan otomatis akan di set=0 dan status dari akan di atur "aktif" pada saat penambahan.

Skenario Penggunaan Sistem:

1. Memilih pilihan (6. Tambah siswa), pilihan ini akan menambahkan siswa.
  2. Admin akan diminta untuk memasukan NIS dan nama dari siswa yang akan di tambahkan.
  3. Jika data yang dimasukan sudah ada di dalam sistem, maka akan sistem akan memberi output (Siswa '{nama}' dengan NIS '{NIS}' sudah ada.)
  4. Jika data yang di masukan tidak ada di dalam sistem(berhasil), maka sistem akan menampilkan pesan (Siswa '{nama}' dengan NIS '{NIS}' berhasil ditambahkan dengan skor awal {poin})
  5. Namun apabila terdapat kesalahan pada saat penambahan siswa, sistem akan menampilkan pesan (Kesalahan saat menambahkan siswa: {err})
- 7. Hapus Data Siswa:** Admin dapat menghapus data siswa beserta histori skorsing nya yang ada di dalam sistem dengan menggunakan ID siswa tersebut. Penghapusan data siswa didasari dengan alasan bahwa siswa tersebut sudah melebihi batas poin sehingga statusnya "tidak aktif", siswa pindah sekolah, ataupun karena siswa tersebut telah lulus.

Skenario Penggunaan Sistem:

1. Memilih pilihan (7. Hapus Data Siswa), pilihan ini akan menghapus data siswa beserta histori nya.
  2. Sistem akan menampilkan data lengkap semua siswa yang tersimpan pada sistem.
  3. Lalu Admin akan diminta untuk memasukan ID siswa yang akan di hapus datanya.
  4. Jika ID siswa yang dipilih Admin tidak terdapat pada sistem, maka akan muncul pesan (Siswa dengan ID {siswa\_id} tidak ditemukan)
  5. Apabila pemilihan ID siswa berhasil dilakukan tanpa kesalahan, maka data dan histori skorsing siswa yang ada di dalam sistem akan di hapus, lalu apabila proses penghapusan data dan histori siswa berhasil dilakukan, sistem akan menampilkan pesan (Siswa '{siswa[0]}' dan histori terkait berhasil dihapus)
  6. Namun apabila terjadi kesalahan pada proses penghapusan siswa, sistem akan menampilkan pesan (Kesalahan saat menghapus siswa: {err})
- 8. Tambah Hukuman:** Admin dapat menambahkan hukuman kedalam sistem sehingga ketika terjadi pelanggaran, Admin hanya perlu memasukan ID hukuman.

Skenario Penggunaan Sistem:

1. Memilih pilihan (8. Tambah Hukuman), pilihan ini akan menambahkan hukuman ke dalam sistem.
2. Admin akan di minta untuk memasukan nama hukuman.
3. Jika nama hukuman sudah ada di dalam sistem maka akan memberikan output (Hukuman '{nama}' sudah ada).
4. Jika nama hukuman belum terdapat di dalam sistem, sistem akan menampilkan nama kategori hukuman dan ID nya yang terdapat pada sistem, lalu Admin akan diminta memilih kategori hukuman yang terdapat pada sistem berdasarkan ID nya.
5. Jika hukuman berhasil di tambahkan ke dalam sistem maka akan muncul pesan (Hukuman '{nama}' berhasil ditambahkan)

6. Namun apabila terdapat kesalahan saat penambahan hukuman, sistem akan memunculkan pesan (Kesalahan saat menambahkan hukuman: {err})

- 9. Tambah Kategori Hukuman:** Admin dapat menambahkan kategori hukuman dan menentukan skor yang berlaku sesuai dengan kategori. Admin akan diminta untuk memasukan nama dan skor dari kategori hukuman yang akan ditambahkan kedalam sistem.

Skenario Penggunaan Sistem:

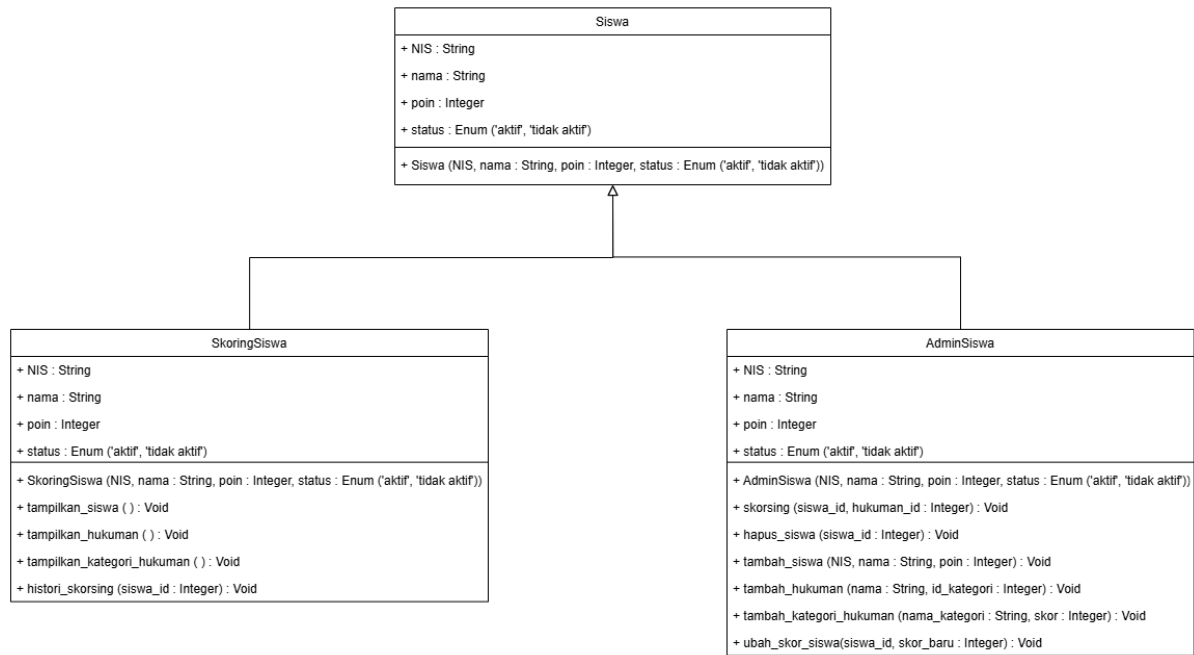
1. Memilih pilihan (9. Tambah Kategori Hukuman), pilihan ini akan menambahkan kategori hukuman ke dalam sistem.
2. Admin akan di minta untuk memasukan nama kategori, dan skor kategori.
3. Jika nama kategori hukuman sudah ada di dalam sistem maka akan memberikan output (Kategori '{nama\_kategori}' sudah ada).
4. Jika nama kategori belum terdapat pada sistem(berhasil), maka kategori akan di tambahkan ke dalam sistem, dan akan menampilkan pesan(Kategori hukuman '{nama\_kategori}' berhasil ditambahkan)
5. Namun jika terjadi kesalahan saat penambahan kategori, sistem akan menampilkan pesan (Kesalahan saat menambahkan kategori hukuman: {err})

- 10. Ubah Skor Siswa:** Admin dapat mengubah skor dari ssiwa yang dipilih berdasarkan ID dari siswa. Admin akan diminta untuk memasukan ID siswa yang akan di kurangi skor nya. Menu ubah skor siswa digunakan sebagai bentuk antisipasi, apabila terdapat kesalahan pada skoring siswa.

Skenario Penggunaan Sistem:

1. Memilih pilihan (10 Ubah Skor Siswa), pilihan ini akan mengubah skor siswa yang dipilih.
2. Sistem akan menampilkan semua data siswa yang terdapat pada sistem.
3. Admin akan di minta untuk memasukan ID siswa.
4. Jika siswa dengan ID yang dituju tidak ada di dalam sistem maka akan memberikan pesan (Siswa dengan ID {siswa\_id} tidak ditemukan)
5. Jika ID siswa terdapat pada sistem dan valid, Admin akan diminta memasukan skor baru. Apabila berhasil mengubah skor siswa, maka sistem akan menampilkan pesan (Skor siswa '{siswa[0]}' berhasil diubah menjadi {skor\_baru})
6. Namun jika terdapat kesalahan pada saat pengubahan skor siswa, sistem akan menampilkan pesan (Kesalahan saat menambahkan kategori hukuman: {err})

### 1.3 Class Diagram Sistem





## BAB II

### PEMBAHASAN

#### 2.1 Implementasi dan Hasil

##### 2.1.1 Menu Utama Sistem

- Source Code

```
1 def menu():
2     admin = AdminSiswa()
3     sistem = SkoringSiswa(None, None)
4     while True:
5         print("\n" + "=" * 40)
6         print("Sistem Skoring Siswa".center(40))
7         print("=" * 40)
8         print("1. Tampilkan Siswa")
9         print("2. Tampilkan Hukuman")
10        print("3. Tampilkan Kategori Hukuman")
11        print("4. Tampilkan Histori Skorsing")
12        print("5. Skorsing Siswa (Admin)")
13        print("6. Tambah Siswa (Admin)")
14        print("7. Hapus Data Siswa (Admin)")
15        print("8. Tambah Hukuman (Admin)")
16        print("9. Tambah Kategori Hukuman (Admin)")
17        print("10. Ubah Skor Siswa (Admin)")
18        print("0. Keluar")
19        print("=" * 40)
20        pilihan = input("Pilih menu: ")
21        if pilihan == "1":
22            sistem.tampilkan_siswa()
23        elif pilihan == "2":
24            sistem.tampilkan_hukuman()
25        elif pilihan == "3":
26            sistem.tampilkan_kategori_hukuman()
27        elif pilihan == "4":
28            sistem.tampilkan_siswa()
29            try:
30                siswa_id = int(input("Masukkan ID siswa untuk melihat histori: "))
31                sistem.histori_skorsing(siswa_id)
32            except ValueError:
33                print("ID harus berupa angka.")
34        elif pilihan == "5":
35            sistem.tampilkan_siswa()
36            try:
37                siswa_id = int(input("Masukkan ID siswa: "))
38                sistem.tampilkan_hukuman()
39                hukuman_id = int(input("Masukkan ID hukuman: "))
40                admin.skorsing(siswa_id, hukuman_id)
41            except ValueError:
42                print("ID harus berupa angka.")
43        elif pilihan == "6":
44            NIS = input("Masukkan NIS siswa: ")
45            nama = input("Masukkan nama siswa: ").strip()
46            if nama and NIS:
47                admin.tambah_siswa(nama, NIS)
48            else:
49                print("Nama siswa tidak boleh kosong.")
50        elif pilihan == "7":
51            sistem.tampilkan_siswa()
52            try:
53                siswa_id = int(input("Masukkan ID siswa yang ingin dihapus: "))
54                admin.hapus_siswa(siswa_id)
55            except ValueError:
56                print("ID harus berupa angka.")
```

```

57     elif pilihan == "8":
58         nama_hukuman = input("Masukkan nama hukuman: ").strip()
59         if not nama_hukuman:
60             print("Nama hukuman tidak boleh kosong.").strip()
61         if not nama_hukuman:
62             print("Nama hukuman tidak boleh kosong.")
63             continue
64         sistem.tampilkan_kategori_hukuman()
65         try:
66             id_kategori = int(input("Masukkan ID kategori hukuman: "))
67             admin.tambah_hukuman(nama_hukuman, id_kategori)
68         except ValueError:
69             print("ID kategori harus berupa angka.")
70     elif pilihan == "9":
71         nama_kategori = input("Masukkan nama kategori hukuman: ").strip()
72         if not nama_kategori:
73             print("Nama kategori hukuman tidak boleh kosong.")
74             continue
75         try:
76             skor = int(input("Masukkan skor untuk kategori hukuman ini: "))
77             admin.tambah_kategori_hukuman(nama_kategori, skor)
78         except ValueError:
79             print("Skor harus berupa angka.")
80     elif pilihan == "10":
81         sistem.tampilkan_siswa()
82         try:
83             siswa_id = int(input("Masukkan ID siswa yang ingin diubah skor: "))
84             skor_baru = int(input("Masukkan skor baru: "))
85             admin.ubah_skor_siswa(siswa_id, skor_baru)
86         except ValueError:
87             print("ID siswa dan skor harus berupa angka.")
88     elif pilihan == "0":
89         print("Terima kasih telah menggunakan Sistem Skoring Siswa.")
90         break
91     else:
92         print("Pilihan tidak valid. Silakan coba lagi.")
93 if __name__ == "__main__":
94     menu()

```

- **Tampilan Menu Utama**

```

=====
                        Sistem Skoring Siswa
=====
1. Tampilkan Siswa
2. Tampilkan Hukuman
3. Tampilkan Kategori Hukuman
4. Tampilkan Histori Skorsing
5. Skorsing Siswa (Admin)
6. Tambah Siswa (Admin)
7. Hapus Data Siswa (Admin)
8. Tambah Hukuman (Admin)
9. Tambah Kategori Hukuman (Admin)
10. Ubah Skor Siswa (Admin)
0. Keluar
=====

```

## 2.1.2 Menu Tampilkan Siswa

- Source Code

```
1 class Siswa:
2     def __init__(self, NIS, nama, poin=0, status='aktif'):
3         self.NIS = NIS
4         self.nama = nama
5         self.poin = poin
6         self.status = status
7
8 class SkoringSiswa(Siswa):
9     def __init__(self, NIS, nama, poin=0, status='aktif'):
10         super().__init__(NIS, nama, poin, status)
11
12     def tampilkan_siswa(self):
13         try:
14             cursor.execute("SELECT id, nama, NIS, poin, status FROM siswa")
15             siswa = cursor.fetchall()
16             if not siswa:
17                 print("✗ Tidak ada siswa yang terdaftar.")
18             else:
19                 headers = ["ID", "Nama", "NIS", "Poin", "Status"]
20                 table = []
21                 for id_siswa, nama, NIS, poin, status in siswa:
22                     status_color = status if status == "aktif" else status
23                     table.append([id_siswa, nama, NIS, poin, status_color])
24
25                 print("\n📋 Daftar Siswa:")
26                 print(tabulate(table, headers=headers, tablefmt="fancy_grid"))
27         except mysql.connector.Error as err:
28             print(f"⚠️ Kesalahan saat mengambil data siswa: {err}")
```

- Tampilan Menu “Tampilkan Siswa”, jika tidak terdapat data siswa pada sistem

```
=====
                        Sistem Skoring Siswa
=====
1. Tampilkan Siswa
2. Tampilkan Hukuman
3. Tampilkan Kategori Hukuman
4. Tampilkan Histori Skorsing
5. Skorsing Siswa (Admin)
6. Tambah Siswa (Admin)
7. Hapus Data Siswa (Admin)
8. Tambah Hukuman (Admin)
9. Tambah Kategori Hukuman (Admin)
10. Ubah Skor Siswa (Admin)
0. Keluar
=====
✗ Tidak ada siswa yang terdaftar.
```

- Tampilan Menu “Tampilkan Siswa”, jika terdapat data yang tersimpan pada sistem, akan menampilkan ID, nama, NIS, poin dan status dari siswa yang tersimpan di sistem

Daftar Siswa:

ID	Nama	NIS	Poin	Status
1	Taufik	5230411040	0	aktif
2	Calvin	5230411006	0	aktif
3	Dhafa	5230411007	0	aktif
4	Putra	5230411026	0	aktif
5	Rum	5230411044	0	aktif
6	Mahes	5230411017	0	aktif

### 2.1.3 Menu Tampilkan Hukuman

- Source Code

```

1 class Siswa:
2     def __init__(self, NIS, nama, poin=0, status='aktif'):
3         self.NIS = NIS
4         self.nama = nama
5         self.poin = poin
6         self.status = status
7
8 class SkoringSiswa(Siswa):
9     def __init__(self, NIS, nama, poin=0, status='aktif'):
10         super().__init__(NIS, nama, poin, status)
11
12     def tampilkan_hukuman(self):
13         try:
14             cursor.execute(
15                 """
16                 SELECT h.id, h.nama, k.skor, kh.nama_kategori
17                 FROM hukuman h
18                 JOIN kategori_hukuman k ON h.id_kategori_hukuman = k.id
19                 JOIN kategori_hukuman kh ON h.id_kategori_hukuman = kh.id
20                 """
21             )
22             hukuman = cursor.fetchall()
23             if not hukuman:
24                 print("✗ Tidak ada hukuman yang terdaftar.")
25             else:
26                 headers = ["ID", "Nama Hukuman", "Skor", "Kategori"]
27                 table = [[h[0], h[1], h[2], h[3]] for h in hukuman]
28
29                 print("\n📄 Daftar Hukuman:")
30                 print(tabulate(table, headers=headers, tablefmt="fancy_grid"))
31         except mysql.connector.Error as err:
32             print(f"⚠️ Kesalahan saat mengambil data hukuman: {err}")

```

- Tampilan Menu “Tampilkan Hukuman”, jika tidak terdapat data hukuman yang tersimpan.

```

=====
                        Sistem Skoring Siswa
=====
1. Tampilkan Siswa
2. Tampilkan Hukuman
3. Tampilkan Kategori Hukuman
4. Tampilkan Histori Skorsing
5. Skorsing Siswa (Admin)
6. Tambah Siswa (Admin)
7. Hapus Data Siswa (Admin)
8. Tambah Hukuman (Admin)
9. Tambah Kategori Hukuman (Admin)
10. Ubah Skor Siswa (Admin)
0. Keluar
=====
✗ Tidak ada hukuman yang terdaftar.

```

- Tampilan Menu “Tampilkan Hukuman”, jika terdapat data hukuman yang tersimpan, sistem akan menampilkan ID, nama hukuman, skor dan kategori nya.

Daftar Hukuman:

ID	Nama Hukuman	Skor	Kategori
1	Terlambat masuk sekolah > 15 menit	5	Sangat Ringan
2	Tidak memakai seragam sekolah	5	Sangat Ringan
3	Membawa handphone	5	Sangat Ringan
4	Membolos	5	Sangat Ringan
5	Meninggalkan kelas tanpa izin	10	Ringan
6	Mengecat rambut	10	Ringan
7	Menyontek saat ulangan	10	Ringan
8	Memakai pakaian tidak sopan di lingkungan sekolah	10	Ringan
9	Merokok di lingkungan sekolah	25	Sedang
10	Melakukan vandalisme di lingkungan sekolah	25	Sedang
11	Berkelahi di lingkungan sekolah	25	Sedang
12	Merusak fasilitas sekolah	25	Sedang
13	Kedapatan mencuri	55	Berat
14	Berjudi dan sejenisnya	55	Berat
15	Melakukan bullying fisik maupun non fisik	55	Berat
16	Membuat keributan di lingkungan sekolah	55	Berat
17	Membawa senjata tajam ke lingkungan sekolah tanpa izin	55	Berat
18	Melakukan tindak asusila	100	Sangat Berat
19	Membawa atau terlibat penyalahgunaan narkoba	100	Sangat Berat

## 2.1.4 Menu Tampilkan Kategori Hukuman

- Source Code

```
1 class Siswa:
2     def __init__(self, NIS, nama, poin=0, status='aktif'):
3         self.NIS = NIS
4         self.nama = nama
5         self.poin = poin
6         self.status = status
7
8 class SkoringSiswa(Siswa):
9     def __init__(self, NIS, nama, poin=0, status='aktif'):
10         super().__init__(NIS, nama, poin, status)
11
12     def tampilkan_kategori_hukuman(self):
13         try:
14             cursor.execute("SELECT id, nama_kategori, skor FROM kategori_hukuman")
15             kategori = cursor.fetchall()
16             if not kategori:
17                 print("✗ Tidak ada kategori hukuman yang terdaftar.")
18             else:
19                 headers = ["ID", "Nama Kategori", "Skor"]
20                 table = [[k[0], k[1], k[2]] for k in kategori]
21
22                 print("\n 📄 Daftar Kategori Hukuman:")
23                 print(tabulate(table, headers=headers, tablefmt="fancy_grid"))
24         except mysql.connector.Error as err:
25             print(f"⚠️ Kesalahan saat mengambil data kategori hukuman: {err}")
```

- Tampilan Menu “Tampilkan Kategori Hukuman”, jika tidak terdapat data hukuman yang tersimpan.

```
=====
                        Sistem Skoring Siswa
=====
1. Tampilkan Siswa
2. Tampilkan Hukuman
3. Tampilkan Kategori Hukuman
4. Tampilkan Histori Skoring
5. Skoring Siswa (Admin)
6. Tambah Siswa (Admin)
7. Hapus Data Siswa (Admin)
8. Tambah Hukuman (Admin)
9. Tambah Kategori Hukuman (Admin)
10. Ubah Skor Siswa (Admin)
0. Keluar
=====
✗ Tidak ada kategori hukuman yang terdaftar.
```

- Tampilan Menu “Tampilkan Kategori Hukuman”, jika terdapat data kategori hukuman yang tersimpan, sistem akan menampilkan ID, nama kategori dan skor.

📄 Daftar Kategori Hukuman:

ID	Nama Kategori	Skor
1	Sangat Ringan	5
2	Ringan	10
3	Sedang	25
4	Berat	55
5	Sangat Berat	100

## 2.1.5 Menu Tampilkan Histori Skorsing

- **Source Code**

```
1 class Siswa:
2     def __init__(self, NIS, nama, poin=0, status='aktif'):
3         self.NIS = NIS
4         self.nama = nama
5         self.poin = poin
6         self.status = status
7
8 class SkoringSiswa(Siswa):
9     def __init__(self, NIS, nama, poin=0, status='aktif'):
10         super().__init__(NIS, nama, poin, status)
11
12     def histori_skorsing(self, siswa_id):
13         try:
14             cursor.execute("SELECT nama FROM siswa WHERE id = %s", (siswa_id,))
15             siswa = cursor.fetchone()
16
17             if not siswa:
18                 print(f"❌ Siswa dengan ID {siswa_id} tidak ditemukan.")
19                 return
20
21             print(f"\n📄 Histori Skorsing untuk '{siswa[0]}':")
22             cursor.execute("""
23                 SELECT h.nama, k.skor
24                 FROM riwayat_hukuman r
25                 JOIN hukuman h ON r.id_hukuman = h.id
26                 JOIN kategori_hukuman k ON h.id_kategori_hukuman = k.id
27                 WHERE r.id_siswa = %s
28                 ORDER BY r.id ASC
29             """, (siswa_id,))
30             riwayat = cursor.fetchall()
31
32             if not riwayat:
33                 print("⚠️ Tidak ada histori skorsing.")
34             else:
35                 headers = ["Keterangan", "Poin"]
36                 table = [[r[0], r[1]] for r in riwayat]
37
38                 print(tabulate(table, headers=headers, tablefmt="fancy_grid"))
39         except mysql.connector.Error as err:
40             print(f"⚠️ Kesalahan saat menampilkan histori skorsing: {err}")
```

- **Tampilan Menu “Tampilkan Histori Skorsing”**, sistem akan menampilkan daftar siswa yang ada di sistem, jika Admin memasukkan ID siswa yang tidak tersimpan dalam daftar siswa

📄 Daftar Siswa:

ID	Nama	NIS	Poin	Status
1	Taufik	5230411040	0	aktif
2	Calvin	5230411006	0	aktif
3	Dhafa	5230411007	0	aktif
4	Putra	5230411026	0	aktif
5	Rum	5230411044	0	aktif
6	Mahes	5230411017	0	aktif

❌ Siswa dengan ID 7 tidak ditemukan.

- **Tampilan Menu “Tampilkan Histori Skorsing”,** jika ID siswa terdapat pada daftar siswa, tapi siswa tidak pernah melakukan pelanggaran, sehingga tidak memiliki histori skorsing

Daftar Siswa:

ID	Nama	NIS	Poin	Status
1	Taufik	5230411040	0	aktif
2	Calvin	5230411006	0	aktif
3	Dhafa	5230411007	0	aktif
4	Putra	5230411026	0	aktif
5	Rum	5230411044	0	aktif
6	Mahes	5230411017	0	aktif

Histori Skorsing untuk 'Taufik':

⚠ Tidak ada histori skorsing.

- **Tampilan Menu “Tampilkan Histori Skorsing”,** jika ID siswa terdapat pada daftar siswa, serta siswa pernah melakukan pelanggaran, sehingga terekam di histori skorsing

Daftar Siswa:

ID	Nama	NIS	Poin	Status
1	Taufik	5230411040	0	aktif
2	Calvin	5230411006	0	aktif
3	Dhafa	5230411007	0	aktif
4	Putra	5230411026	20	aktif
5	Rum	5230411044	0	aktif
6	Mahes	5230411017	0	aktif

Histori Skorsing untuk 'Putra':

Keterangan	Poin
Meninggalkan kelas tanpa izin	10
Membolos	5
Membawa handphone	5



## 2.1.6 Menu Skorsing Siswa

- Source Code

```
1 class Siswa:
2     def __init__(self, NIS, nama, poin=0, status='aktif'):
3         self.NIS = NIS
4         self.nama = nama
5         self.poin = poin
6         self.status = status
7
8 class AdminSiswa(Siswa):
9     def __init__(self, NIS=None, nama=None, poin=0, status='aktif'):
10         super().__init__(NIS, nama, poin, status)
11
12     def skorsing(self, siswa_id, hukuman_id):
13         try:
14             cursor.execute("SELECT id, nama, poin, status FROM siswa WHERE id = %s", (siswa_id,))
15             siswa = cursor.fetchone()
16
17             if not siswa:
18                 print(f"❌ Siswa dengan ID {siswa_id} tidak ditemukan.")
19                 return
20
21             if siswa[3] == "tidak aktif":
22                 print(f"⚠️ Siswa '{siswa[1]}' sudah tidak aktif.")
23                 return
24
25             cursor.execute("""
26                 SELECT h.id, h.nama, k.skor
27                 FROM hukuman h
28                 JOIN kategori_hukuman k ON h.id_kategori_hukuman = k.id
29                 WHERE h.id = %s
30             """, (hukuman_id,))
31             hukuman = cursor.fetchone()
32
33             if not hukuman:
34                 print(f"❌ Hukuman dengan ID {hukuman_id} tidak ditemukan.")
35                 return
36
37             siswa_id, nama_siswa, poin_siswa, _ = siswa
38             _, nama_hukuman, skor_hukuman = hukuman
39
40             poin_baru = poin_siswa + skor_hukuman
41             cursor.execute("UPDATE siswa SET poin = %s WHERE id = %s", (poin_baru, siswa_id))
42             conn.commit()
43
44             cursor.execute("INSERT INTO riwayat_hukuman (id_siswa, id_hukuman) VALUES (%s, %s)", (siswa_id, hukuman_id))
45             conn.commit()
46
47             print(
48                 f"✅ Siswa '{nama_siswa}' dikenai hukuman '{nama_hukuman}' dengan skor +{skor_hukuman}. Poin sekarang: {poin_baru}."
49             )
50
51             if poin_baru >= 100:
52                 print(f"❌ Siswa '{nama_siswa}' telah dikeluarkan karena poin mencapai 100.")
53                 cursor.execute("UPDATE siswa SET status = 'tidak aktif' WHERE id = %s", (siswa_id,))
54                 conn.commit()
55
56         except mysql.connector.Error as err:
57             print(f"⚠️ Kesalahan saat memproses skorsing: {err}")
```

- Tampilan Menu “Skorsing Siswa”, jika tidak ada hukuman yang terdaftar

Daftar Siswa:

ID	Nama	NIS	Poin	Status
1	Taufik	5230411040	0	aktif
2	Calvin	5230411006	0	aktif
3	Dhafa	5230411007	0	aktif
4	Putra	5230411026	0	aktif
5	Rum	5230411044	0	aktif
6	Mahes	5230411017	0	aktif

❌ Tidak ada hukuman yang terdaftar.

- Tampilan Menu “Skorsing Siswa”, jika terdapat hukuman yang tersimpan pada sistem dan ID siswa yang dipilih valid

Daftar Siswa:

ID	Nama	NIS	Poin	Status
1	Taufik	5230411040	0	aktif
2	Calvin	5230411006	0	aktif
3	Dhafa	5230411007	0	aktif
4	Putra	5230411026	15	aktif
5	Rum	5230411044	0	aktif
6	Mahes	5230411017	0	aktif

Daftar Hukuman:

ID	Nama Hukuman	Skor	Kategori
1	Terlambat masuk sekolah > 15 menit	5	Sangat Ringan
2	Tidak memakai seragam sekolah	5	Sangat Ringan
3	Membawa handphone	5	Sangat Ringan
4	Membolos	5	Sangat Ringan
5	Meninggalkan kelas tanpa izin	10	Ringan
6	Mengecat rambut	10	Ringan
7	Menyontek saat ulangan	10	Ringan
8	Memakai pakaian tidak sopan di lingkungan sekolah	10	Ringan
9	Merokok di lingkungan sekolah	25	Sedang
10	Melakukan vandalisme di lingkungan sekolah	25	Sedang
11	Berkelahi di lingkungan sekolah	25	Sedang
12	Merusak fasilitas sekolah	25	Sedang
13	Kedapatan mencuri	55	Berat
14	Berjudi dan sejenisnya	55	Berat
15	Melakukan bullying fisik maupun non fisik	55	Berat
16	Membuat keributan di lingkungan sekolah	55	Berat
17	Membawa senjata tajam ke lingkungan sekolah tanpa izin	55	Berat
18	Melakukan tindak asusila	100	Sangat Berat
19	Membawa atau terlibat penyalahgunaan narkoba	100	Sangat Berat

✓ Siswa 'Putra' dikenai hukuman 'Meninggalkan kelas tanpa izin' dengan skor +10. Poin sekarang: 10.

- **Tampilan Menu “Skorsing Siswa”,** jika terdapat hukuman yang tersimpan pada sistem dan ID siswa yang dipilih valid, serta setelah ditambahkan skorsing baru, skor siswa mencapai  $\Rightarrow 100$  poin, maka siswa akan di keluarkan dari sekolah

Daftar Hukuman:

ID	Nama Hukuman	Skor	Kategori
1	Terlambat masuk sekolah > 15 menit	5	Sangat Ringan
2	Tidak memakai seragam sekolah	5	Sangat Ringan
3	Membawa handphone	5	Sangat Ringan
4	Membolos	5	Sangat Ringan
5	Meninggalkan kelas tanpa izin	10	Ringan
6	Mengecat rambut	10	Ringan
7	Menyontek saat ulangan	10	Ringan
8	Memakai pakaian tidak sopan di lingkungan sekolah	10	Ringan
9	Merokok di lingkungan sekolah	25	Sedang
10	Melakukan vandalisme di lingkungan sekolah	25	Sedang
11	Berkelahi di lingkungan sekolah	25	Sedang
12	Merusak fasilitas sekolah	25	Sedang
13	Kedapatan mencuri	55	Berat
14	Berjudi dan sejenisnya	55	Berat
15	Melakukan bullying fisik maupun non fisik	55	Berat
16	Membuat keributan di lingkungan sekolah	55	Berat
17	Membawa senjata tajam ke lingkungan sekolah tanpa izin	55	Berat
18	Melakukan tindak asusila	100	Sangat Berat
19	Membawa atau terlibat penyalahgunaan narkoba	100	Sangat Berat

- ✓ Siswa 'Calvin' dikenai hukuman 'Melakukan tindak asusila' dengan skor +100. Poin sekarang: 105.
- ✗ Siswa 'Calvin' telah dikeluarkan karena poin mencapai 100.

## 2.1.7 Menu Tambah Siswa

- **Source Code**

```

1 class Siswa:
2     def __init__(self, NIS, nama, poin=0, status='aktif'):
3         self.NIS = NIS
4         self.nama = nama
5         self.poin = poin
6         self.status = status
7
8 class AdminSiswa(Siswa):
9     def __init__(self, NIS=None, nama=None, poin=0, status='aktif'):
10         super().__init__(NIS, nama, poin, status)
11
12     def tambah_siswa(self, nama, NIS, poin=0):
13         try:
14             cursor.execute("INSERT INTO siswa (nama, NIS, poin, status) VALUES (%s, %s, %s, 'aktif')", (nama, NIS, poin))
15             conn.commit()
16             print(f"✓ Siswa '{nama}' dengan NIS '{NIS}' berhasil ditambahkan dengan skor awal {poin}.")
17         except mysql.connector.IntegrityError:
18             print(f"⚠ Siswa '{nama}' dengan NIS '{NIS}' sudah ada.")
19         except mysql.connector.Error as err:
20             print(f"⚠ Kesalahan saat menambahkan siswa: {err}")

```

- Tampilan Menu “Tambah Siswa”, jika data siswa yang ditambahkan sudah tersimpan di sistem

```
=====
                        Sistem Skoring Siswa
=====
1. Tampilkan Siswa
2. Tampilkan Hukuman
3. Tampilkan Kategori Hukuman
4. Tampilkan Histori Skorsing
5. Skorsing Siswa (Admin)
6. Tambah Siswa (Admin)
7. Hapus Data Siswa (Admin)
8. Tambah Hukuman (Admin)
9. Tambah Kategori Hukuman (Admin)
10. Ubah Skor Siswa (Admin)
0. Keluar
=====
⚠ Siswa 'Taufik' dengan NIS '5230411040' sudah ada.
```

- Tampilan Menu “Tambah Siswa”, jika data siswa yang ditambahkan belum tersimpan pada sistem dan proses penambahan siswa berhasil dilakukan

```
=====
                        Sistem Skoring Siswa
=====
1. Tampilkan Siswa
2. Tampilkan Hukuman
3. Tampilkan Kategori Hukuman
4. Tampilkan Histori Skorsing
5. Skorsing Siswa (Admin)
6. Tambah Siswa (Admin)
7. Hapus Data Siswa (Admin)
8. Tambah Hukuman (Admin)
9. Tambah Kategori Hukuman (Admin)
10. Ubah Skor Siswa (Admin)
0. Keluar
=====
✅ Siswa 'Ridho' dengan NIS '5230411021' berhasil ditambahkan dengan skor awal 0.
```

## 2.1.8 Menu Hapus Data Siswa

- Source Code

```
1 class Siswa:
2     def __init__(self, NIS, nama, poin=0, status='aktif'):
3         self.NIS = NIS
4         self.nama = nama
5         self.poin = poin
6         self.status = status
7
8 class AdminSiswa(Siswa):
9     def __init__(self, NIS=None, nama=None, poin=0, status='aktif'):
10         super().__init__(NIS, nama, poin, status)
11
12     def hapus_siswa(self, siswa_id):
13         try:
14             cursor.execute("SELECT nama FROM siswa WHERE id = %s", (siswa_id,))
15             siswa = cursor.fetchone()
16
17             if not siswa:
18                 print(f"❌ Siswa dengan ID {siswa_id} tidak ditemukan.")
19                 return
20
21             cursor.execute("DELETE FROM riwayat_hukuman WHERE id_siswa = %s", (siswa_id,))
22             conn.commit()
23
24             cursor.execute("DELETE FROM siswa WHERE id = %s", (siswa_id,))
25             conn.commit()
26
27             print(f"✅ Siswa '{siswa[0]}' dan histori terkait berhasil dihapus.")
28         except mysql.connector.Error as err:
29             print(f"⚠ Kesalahan saat menghapus siswa: {err}")
```

- Tampilan Menu “Hapus Siswa”, jika ID siswa yang dipilih untuk dihapus tidak tersedia di dalam sistem

Daftar Siswa:

ID	Nama	NIS	Poin	Status
1	Taufik	5230411040	0	aktif
2	Calvin	5230411006	105	tidak aktif
3	Dhafa	5230411007	0	aktif
4	Putra	5230411026	20	aktif
5	Rum	5230411044	0	aktif
6	Mahes	5230411017	0	aktif

✗ Siswa dengan ID 10 tidak ditemukan.

- Tampilan Menu “Hapus Siswa”, jika ID siswa yang dipilih untuk dihapus tersedia di dalam sistem dan proses penghapusan siswa berhasil dilakukan

Daftar Siswa:

ID	Nama	NIS	Poin	Status
1	Taufik	5230411040	0	aktif
2	Calvin	5230411006	105	tidak aktif
3	Dhafa	5230411007	0	aktif
4	Putra	5230411026	20	aktif
5	Rum	5230411044	0	aktif
6	Mahes	5230411017	0	aktif
8	Ridho	5230411021	0	aktif

✓ Siswa 'Ridho' dan histori terkait berhasil dihapus.

## 2.1.9 Menu Tambah Hukuman

- Source Code

```

1 class Siswa:
2     def __init__(self, NIS, nama, poin=0, status='aktif'):
3         self.NIS = NIS
4         self.nama = nama
5         self.poin = poin
6         self.status = status
7
8 class AdminSiswa(Siswa):
9     def __init__(self, NIS=None, nama=None, poin=0, status='aktif'):
10         super().__init__(NIS, nama, poin, status)
11
12     def tambah_hukuman(self, nama, id_kategori):
13         try:
14             cursor.execute("INSERT INTO hukuman (nama, id_kategori_hukuman) VALUES (%s, %s)", (nama, id_kategori))
15             conn.commit()
16             print(f"✓ Hukuman '{nama}' berhasil ditambahkan.")
17         except mysql.connector.IntegrityError:
18             print(f"⚠ Hukuman '{nama}' sudah ada.")
19         except mysql.connector.Error as err:
20             print(f"⚠ Kesalahan saat menambahkan hukuman: {err}")

```

- Tampilan Menu “Tambah Hukuman”, jika pada saat akan dilakukan penambahan hukuman, tetapi belum ada kategori yang tersimpan pada sistem

```

=====
                        Sistem Skoring Siswa
=====
1. Tampilkan Siswa
2. Tampilkan Hukuman
3. Tampilkan Kategori Hukuman
4. Tampilkan Histori Skorsing
5. Skorsing Siswa (Admin)
6. Tambah Siswa (Admin)
7. Hapus Data Siswa (Admin)
8. Tambah Hukuman (Admin)
9. Tambah Kategori Hukuman (Admin)
10. Ubah Skor Siswa (Admin)
0. Keluar
=====
❌ Tidak ada kategori hukuman yang terdaftar.

```

- Tampilan Menu “Tambah Hukuman”, jika pada saat akan dilakukan penambahan hukuman, sudah terdapat kategori yang tersimpan dan nama hukuman yang ditambahkan belum tersedia di dalam sistem, serta proses penambahan berhasil dilakukan

Daftar Kategori Hukuman:

ID	Nama Kategori	Skor
1	Sangat Ringan	5
2	Ringan	10
3	Sedang	25
4	Berat	55
5	Sangat Berat	100

✅ Hukuman 'Membawa atau terlibat penyalahgunaan miras' berhasil ditambahkan.

## 2.1.10 Menu Tambah Kategori Hukuman

- Source Code

```

1 class Siswa:
2     def __init__(self, NIS, nama, poin=0, status='aktif'):
3         self.NIS = NIS
4         self.nama = nama
5         self.poin = poin
6         self.status = status
7
8 class AdminSiswa(Siswa):
9     def __init__(self, NIS=None, nama=None, poin=0, status='aktif'):
10         super().__init__(NIS, nama, poin, status)
11
12     def tambah_kategori_hukuman(self, nama_kategori, skor):
13         try:
14             cursor.execute("INSERT INTO kategori_hukuman (nama_kategori, skor) VALUES (%s, %s)", (nama_kategori, skor))
15             conn.commit()
16             print(f"✅ Kategori hukuman '{nama_kategori}' berhasil ditambahkan.")
17         except mysql.connector.IntegrityError:
18             print(f"⚠️ Kategori '{nama_kategori}' sudah ada.")
19         except mysql.connector.Error as err:
20             print(f"⚠️ Kesalahan saat menambahkan kategori hukuman: {err}")

```

- Tampilan Menu “Tambah Kategori Hukuman”, jika proses penambahan kategori berhasil dilakukan dengan kategori yang belum tersimpan pada sistem

```
=====
                        Sistem Skoring Siswa
=====
1. Tampilkan Siswa
2. Tampilkan Hukuman
3. Tampilkan Kategori Hukuman
4. Tampilkan Histori Skorsing
5. Skorsing Siswa (Admin)
6. Tambah Siswa (Admin)
7. Hapus Data Siswa (Admin)
8. Tambah Hukuman (Admin)
9. Tambah Kategori Hukuman (Admin)
10. Ubah Skor Siswa (Admin)
0. Keluar
=====
✅ Kategori hukuman 'Cukup Berat' berhasil ditambahkan.
```

### 2.1.11 Menu Ubah Skor Siswa

- Source Code

```
1 class Siswa:
2     def __init__(self, NIS, nama, poin=0, status='aktif'):
3         self.NIS = NIS
4         self.nama = nama
5         self.poin = poin
6         self.status = status
7
8 class AdminSiswa(Siswa):
9     def __init__(self, NIS=None, nama=None, poin=0, status='aktif'):
10         super().__init__(NIS, nama, poin, status)
11
12     def ubah_skor_siswa(self, siswa_id, skor_baru):
13         try:
14             cursor.execute("SELECT nama, poin FROM siswa WHERE id = %s", (siswa_id,))
15             siswa = cursor.fetchone()
16
17             if not siswa:
18                 print(f"❌ Siswa dengan ID {siswa_id} tidak ditemukan.")
19                 return
20
21             cursor.execute("UPDATE siswa SET poin = %s WHERE id = %s", (skor_baru, siswa_id))
22             conn.commit()
23
24             print(f"✅ Skor siswa '{siswa[0]}' berhasil diubah menjadi {skor_baru}.")
25         except mysql.connector.Error as err:
26             print(f"⚠️ Kesalahan saat mengubah skor siswa: {err}")
```

- Tampilan Menu “Ubah Skor Siswa”, jika ID siswa yang akan diubah skor nya tidak tersedia di dalam sistem

Daftar Siswa:

ID	Nama	NIS	Poin	Status
1	Taufik	5230411040	0	aktif
2	Calvin	5230411006	105	tidak aktif
3	Dhafa	5230411007	0	aktif
4	Putra	5230411026	20	aktif
5	Rum	5230411044	0	aktif
6	Mahes	5230411017	0	aktif

❌ Siswa dengan ID 10 tidak ditemukan.

- Tampilan Menu “Ubah Skor Siswa”, jika ID siswa yang akan diubah skor nya, tersedia di dalam sistem, dan proses pengubahan skor siswa berhasil dilakukan

Daftar Siswa:

ID	Nama	NIS	Poin	Status
1	Taufik	5230411040	0	aktif
2	Calvin	5230411006	105	tidak aktif
3	Dhafa	5230411007	0	aktif
4	Putra	5230411026	40	aktif
5	Rum	5230411044	0	aktif
6	Mahes	5230411017	0	aktif

✓ Skor siswa 'Putra' berhasil diubah menjadi 50.

## 2.1.12 Menu Keluar

- Source Code

```

1 def menu():
2     admin = AdminSiswa()
3     sistem = SkoringSiswa(None, None)
4     while True:
5         print("\n" + "=" * 40)
6         print("Sistem Skoring Siswa".center(40))
7         print("=" * 40)
8         print("1. Tampilkan Siswa")
9         print("2. Tampilkan Hukuman")
10        print("3. Tampilkan Kategori Hukuman")
11        print("4. Tampilkan Histori Skorsing")
12        print("5. Skorsing Siswa (Admin)")
13        print("6. Tambah Siswa (Admin)")
14        print("7. Hapus Data Siswa (Admin)")
15        print("8. Tambah Hukuman (Admin)")
16        print("9. Tambah Kategori Hukuman (Admin)")
17        print("10. Ubah Skor Siswa (Admin)")
18        print("0. Keluar")
19        print("=" * 40)
20        pilihan = input("Pilih menu: ")
21        elif pilihan == "0":
22            print("Terima kasih telah menggunakan Sistem Skoring Siswa.")
23            break
24 if __name__ == "__main__":
25     menu()

```

- Tampilan keluar dari sistem skorsing

```

=====
                        Sistem Skoring Siswa
=====
1. Tampilkan Siswa
2. Tampilkan Hukuman
3. Tampilkan Kategori Hukuman
4. Tampilkan Histori Skorsing
5. Skorsing Siswa (Admin)
6. Tambah Siswa (Admin)
7. Hapus Data Siswa (Admin)
8. Tambah Hukuman (Admin)
9. Tambah Kategori Hukuman (Admin)
10. Ubah Skor Siswa (Admin)
0. Keluar
=====
Terima kasih telah menggunakan Sistem Skoring Siswa.

```



## 2.2 Source Code dan Database Sistem

### 2.2.1 Souce Code Sistem

```
1 import mysql.connector
2 from tabulate import tabulate
3 DB_CONFIG = {"host": "localhost", "user": "root", "password": "", "database": "pbo_skorsing"}
4 try:
5     conn = mysql.connector.connect(**DB_CONFIG)
6     cursor = conn.cursor()
7 except mysql.connector.Error as err:
8     print(f"Error saat koneksi ke database: {err}")
9     exit()
10
11 class Siswa:
12     def __init__(self, NIS, nama, poin=0, status='aktif'):
13         self.NIS = NIS
14         self.nama = nama
15         self.poin = poin
16         self.status = status
17
18 class SkoringSiswa(Siswa):
19     def __init__(self, NIS, nama, poin=0, status='aktif'):
20         super().__init__(NIS, nama, poin, status)
21     def tampilkan_siswa(self):
22         try:
23             cursor.execute("SELECT id, nama, NIS, poin, status FROM siswa")
24             siswa = cursor.fetchall()
25             if not siswa:
26                 print("❌ Tidak ada siswa yang terdaftar.")
27             else:
28                 headers = ["ID", "Nama", "NIS", "Poin", "Status"]
29                 table = []
30                 for id_siswa, nama, NIS, poin, status in siswa:
31                     status_color = status if status == "aktif" else status
32                     table.append([id_siswa, nama, NIS, poin, status_color])
33                 print("\n 📋 Daftar Siswa:")
34                 print(tabulate(table, headers=headers, tablefmt="fancy_grid"))
35             except mysql.connector.Error as err:
36                 print(f"⚠️ Kesalahan saat mengambil data siswa: {err}")
37     def tampilkan_hukuman(self):
38         try:
39             cursor.execute(
40                 """
41                 SELECT h.id, h.nama, k.skor, kh.nama_kategori
42                 FROM hukuman h
43                 JOIN kategori_hukuman k ON h.id_kategori_hukuman = k.id
44                 JOIN kategori_hukuman kh ON h.id_kategori_hukuman = kh.id
45                 """
46             )
47             hukuman = cursor.fetchall()
48             if not hukuman:
49                 print("❌ Tidak ada hukuman yang terdaftar.")
50             else:
51                 headers = ["ID", "Nama Hukuman", "Skor", "Kategori"]
52                 table = [[h[0], h[1], h[2], h[3]] for h in hukuman]
53                 print("\n 📋 Daftar Hukuman:")
54                 print(tabulate(table, headers=headers, tablefmt="fancy_grid"))
55             except mysql.connector.Error as err:
56                 print(f"⚠️ Kesalahan saat mengambil data hukuman: {err}")
57     def tampilkan_kategori_hukuman(self):
58         try:
59             cursor.execute("SELECT id, nama_kategori, skor FROM kategori_hukuman")
60             kategori = cursor.fetchall()
61             if not kategori:
62                 print("❌ Tidak ada kategori hukuman yang terdaftar.")
63             else:
64                 headers = ["ID", "Nama Kategori", "Skor"]
65                 table = [[k[0], k[1], k[2]] for k in kategori]
66                 print("\n 📋 Daftar Kategori Hukuman:")
67                 print(tabulate(table, headers=headers, tablefmt="fancy_grid"))
68             except mysql.connector.Error as err:
69                 print(f"⚠️ Kesalahan saat mengambil data kategori hukuman: {err}")
70     def histori_skorsing(self, siswa_id):
71         try:
72             cursor.execute("SELECT nama FROM siswa WHERE id = %s", (siswa_id,))
73             siswa = cursor.fetchone()
74             if not siswa:
75                 print(f"❌ Siswa dengan ID {siswa_id} tidak ditemukan.")
76                 return
77             print(f"\n 📋 Histori Skorsing untuk '{siswa[0]}':")
78             cursor.execute(
79                 """
80                 SELECT h.nama, k.skor
81                 FROM riwayat_hukuman r
82                 JOIN hukuman h ON r.id_hukuman = h.id
83                 JOIN kategori_hukuman k ON h.id_kategori_hukuman = k.id
84                 WHERE r.id_siswa = %s
85                 ORDER BY r.id ASC
86                 """
87             )
88             riwayat = cursor.fetchall()
89             if not riwayat:
90                 print("⚠️ Tidak ada histori skorsing.")
91             else:
92                 headers = ["Keterangan", "Poin"]
93                 table = [[r[0], r[1]] for r in riwayat]
94                 print(tabulate(table, headers=headers, tablefmt="fancy_grid"))
95             except mysql.connector.Error as err:
96                 print(f"⚠️ Kesalahan saat menampilkan histori skorsing: {err}")
```

```

94
95 class AdminSiswa(Siswa):
96     def __init__(self, NIS=None, nama=None, poin=0, status='aktif'):
97         super().__init__(NIS, nama, poin, status)
98     def skorsing(self, siswa_id, hukuman_id):
99         try:
100             cursor.execute("SELECT id, nama, poin, status FROM siswa WHERE id = %s", (siswa_id,))
101             siswa = cursor.fetchone()
102             if not siswa:
103                 print(f"❌ Siswa dengan ID {siswa_id} tidak ditemukan.")
104                 return
105             if siswa[3] == "tidak aktif":
106                 print(f"⚠️ Siswa '{siswa[1]}' sudah tidak aktif.")
107                 return
108             cursor.execute("""
109                 SELECT h.id, h.nama, k.skor
110                 FROM hukuman h
111                 JOIN kategori_hukuman k ON h.id_kategori_hukuman = k.id
112                 WHERE h.id = %s
113             """, (hukuman_id,))
114             hukuman = cursor.fetchone()
115             if not hukuman:
116                 print(f"❌ Hukuman dengan ID {hukuman_id} tidak ditemukan.")
117                 return
118             siswa_id, nama_siswa, poin_siswa, _ = siswa
119             _, nama_hukuman, skor_hukuman = hukuman
120             poin_baru = poin_siswa + skor_hukuman
121             cursor.execute("UPDATE siswa SET poin = %s WHERE id = %s", (poin_baru, siswa_id))
122             conn.commit()
123             cursor.execute("INSERT INTO riwayat_hukuman (id_siswa, id_hukuman) VALUES (%s, %s)", (siswa_id, hukuman_id))
124             conn.commit()
125             print(f"✅ Siswa '{nama_siswa}' dikenai hukuman '{nama_hukuman}' dengan skor +{skor_hukuman}. Poin sekarang: {poin_baru}.")
126             if poin_baru >= 100:
127                 print(f"❌ Siswa '{nama_siswa}' telah dikeluarkan karena poin mencapai 100.")
128                 cursor.execute("UPDATE siswa SET status = 'tidak aktif' WHERE id = %s", (siswa_id,))
129                 conn.commit()
130             except mysql.connector.Error as err:
131                 print(f"⚠️ Kesalahan saat memproses skorsing: {err}")
132     def hapus_siswa(self, siswa_id):
133         try:
134             cursor.execute("SELECT nama FROM siswa WHERE id = %s", (siswa_id,))
135             siswa = cursor.fetchone()
136             if not siswa:
137                 print(f"❌ Siswa dengan ID {siswa_id} tidak ditemukan.")
138                 return
139             cursor.execute("DELETE FROM riwayat_hukuman WHERE id_siswa = %s", (siswa_id,))
140             conn.commit()
141             cursor.execute("DELETE FROM siswa WHERE id = %s", (siswa_id,))
142             conn.commit()
143             print(f"✅ Siswa '{siswa[0]}' dan histori terkait berhasil dihapus.")
144             except mysql.connector.Error as err:
145                 print(f"⚠️ Kesalahan saat menghapus siswa: {err}")
146     def tambah_siswa(self, nama, NIS, poin=0):
147         try:
148             cursor.execute("INSERT INTO siswa (nama, NIS, poin, status) VALUES (%s, %s, %s, 'aktif')", (nama, NIS, poin))
149             conn.commit()
150             print(f"✅ Siswa '{nama}' dengan NIS '{NIS}' berhasil ditambahkan dengan skor awal {poin}.")
151             except mysql.connector.IntegrityError:
152                 print(f"⚠️ Siswa '{nama}' dengan NIS '{NIS}' sudah ada.")
153             except mysql.connector.Error as err:
154                 print(f"⚠️ Kesalahan saat menambahkan siswa: {err}")
155     def tambah_hukuman(self, nama, id_kategori):
156         try:
157             cursor.execute("INSERT INTO hukuman (nama, id_kategori_hukuman) VALUES (%s, %s)", (nama, id_kategori))
158             conn.commit()
159             print(f"✅ Hukuman '{nama}' berhasil ditambahkan.")
160             except mysql.connector.IntegrityError:
161                 print(f"⚠️ Hukuman '{nama}' sudah ada.")
162             except mysql.connector.Error as err:
163                 print(f"⚠️ Kesalahan saat menambahkan hukuman: {err}")
164     def ubah_skor_siswa(self, siswa_id, skor_baru):
165         try:
166             cursor.execute("SELECT nama, poin FROM siswa WHERE id = %s", (siswa_id,))
167             siswa = cursor.fetchone()
168             if not siswa:
169                 print(f"❌ Siswa dengan ID {siswa_id} tidak ditemukan.")
170                 return
171             cursor.execute("UPDATE siswa SET poin = %s WHERE id = %s", (skor_baru, siswa_id))
172             conn.commit()
173             print(f"✅ Skor siswa '{siswa[0]}' berhasil diubah menjadi {skor_baru}.")
174             except mysql.connector.Error as err:
175                 print(f"⚠️ Kesalahan saat mengubah skor siswa: {err}")

```

```

176 def tambah_kategori_hukuman(self, nama_kategori, skor):
177     try:
178         cursor.execute("INSERT INTO kategori_hukuman (nama_kategori, skor) VALUES (%s, %s)", (nama_kategori, skor))
179         conn.commit()
180         print(f"✅ Kategori hukuman '{nama_kategori}' berhasil ditambahkan.")
181     except mysql.connector.IntegrityError:
182         print(f"⚠️ Kategori '{nama_kategori}' sudah ada.")
183     except mysql.connector.Error as err:
184         print(f"⚠️ Kesalahan saat menambahkan kategori hukuman: {err}")
185
186 def menu():
187     admin = AdminSiswa()
188     sistem = SkoringSiswa(None, None)
189     while True:
190         print("\n" + "=" * 40)
191         print("Sistem Skoring Siswa".center(40))
192         print("=" * 40)
193         print("1. Tampilkan Siswa")
194         print("2. Tampilkan Hukuman")
195         print("3. Tampilkan Kategori Hukuman")
196         print("4. Tampilkan Histori Skorsing")
197         print("5. Skorsing Siswa (Admin)")
198         print("6. Tambah Siswa (Admin)")
199         print("7. Hapus Data Siswa (Admin)")
200         print("8. Tambah Hukuman (Admin)")
201         print("9. Tambah Kategori Hukuman (Admin)")
202         print("10. Ubah Skor Siswa (Admin)")
203         print("0. Keluar")
204         print("=" * 40)
205         pilihan = input("Pilih menu: ")
206         if pilihan == "1":
207             sistem.tampilkan_siswa()
208         elif pilihan == "2":
209             sistem.tampilkan_hukuman()
210         elif pilihan == "3":
211             sistem.tampilkan_kategori_hukuman()
212         elif pilihan == "4":
213             sistem.tampilkan_siswa()
214             try:
215                 siswa_id = int(input("Masukkan ID siswa untuk melihat histori: "))
216                 sistem.histori_skorsing(siswa_id)
217             except ValueError:
218                 print("ID harus berupa angka.")
219         elif pilihan == "5":
220             sistem.tampilkan_siswa()
221             try:
222                 siswa_id = int(input("Masukkan ID siswa: "))
223                 sistem.tampilkan_hukuman()
224                 hukuman_id = int(input("Masukkan ID hukuman: "))
225                 admin.skorsing(siswa_id, hukuman_id)
226             except ValueError:
227                 print("ID harus berupa angka.")
228         elif pilihan == "6":
229             NIS = input("Masukkan NIS siswa: ")
230             nama = input("Masukkan nama siswa: ").strip()
231             if nama and NIS:
232                 admin.tambah_siswa(nama, NIS)
233             else:
234                 print("Nama siswa tidak boleh kosong.")
235         elif pilihan == "7":
236             sistem.tampilkan_siswa()
237             try:
238                 siswa_id = int(input("Masukkan ID siswa yang ingin dihapus: "))
239                 admin.hapus_siswa(siswa_id)
240             except ValueError:
241                 print("ID harus berupa angka.")
242         elif pilihan == "8":
243             nama_hukuman = input("Masukkan nama hukuman: ").strip()
244             if not nama_hukuman:
245                 print("Nama hukuman tidak boleh kosong.").strip()
246             if not nama_hukuman:
247                 print("Nama hukuman tidak boleh kosong.")
248                 continue
249             sistem.tampilkan_kategori_hukuman()
250             try:
251                 id_kategori = int(input("Masukkan ID kategori hukuman: "))
252                 admin.tambah_hukuman(nama_hukuman, id_kategori)
253             except ValueError:
254                 print("ID kategori harus berupa angka.")
255         elif pilihan == "9":
256             nama_kategori = input("Masukkan nama kategori hukuman: ").strip()
257             if not nama_kategori:
258                 print("Nama kategori hukuman tidak boleh kosong.")
259                 continue

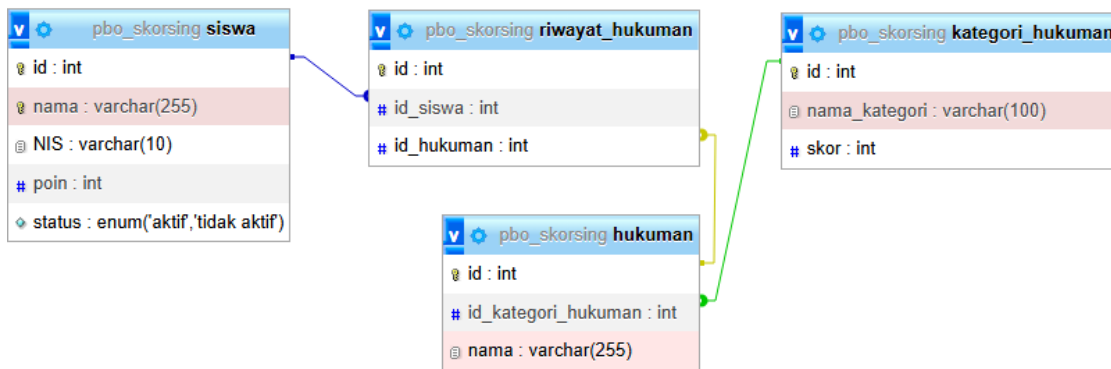
```

```

260         try:
261             skor = int(input("Masukkan skor untuk kategori hukuman ini: "))
262             admin.tambah_kategori_hukuman(nama_kategori, skor)
263         except ValueError:
264             print("Skor harus berupa angka.")
265     elif pilihan == "10":
266         sistem.tampilkan_siswa()
267         try:
268             siswa_id = int(input("Masukkan ID siswa yang ingin diubah skor: "))
269             skor_baru = int(input("Masukkan skor baru: "))
270             admin.ubah_skor_siswa(siswa_id, skor_baru)
271         except ValueError:
272             print("ID siswa dan skor harus berupa angka.")
273     elif pilihan == "0":
274         print("Terima kasih telah menggunakan Sistem Skoring Siswa.")
275         break
276     else:
277         print("Pilihan tidak valid. Silakan coba lagi.")
278 if __name__ == "__main__":
279     menu()

```

## 2.2.2 Database Sistem



```

1  CREATE DATABASE pbo_skorsing;
2  -- Tabel siswa
3  CREATE TABLE siswa (
4      id INT AUTO_INCREMENT PRIMARY KEY,
5      nama VARCHAR(255) NOT NULL,
6      NIS VARCHAR(10) NOT NULL UNIQUE,
7      poin INT NOT NULL,
8      status ENUM('aktif', 'tidak aktif') NOT NULL
9  );
10 -- Tabel kategori_hukuman
11 CREATE TABLE kategori_hukuman (
12     id INT AUTO_INCREMENT PRIMARY KEY,
13     nama_kategori VARCHAR(100) NOT NULL,
14     skor INT NOT NULL
15 );
16 -- Tabel hukuman
17 CREATE TABLE hukuman (
18     id INT AUTO_INCREMENT PRIMARY KEY,
19     id_kategori_hukuman INT NOT NULL,
20     nama VARCHAR(255) NOT NULL,
21     FOREIGN KEY (id_kategori_hukuman) REFERENCES kategori_hukuman(id)
22 );
23 -- Tabel riwayat_hukuman
24 CREATE TABLE riwayat_hukuman (
25     id INT AUTO_INCREMENT PRIMARY KEY,
26     id_siswa INT NOT NULL,
27     id_hukuman INT NOT NULL,
28     FOREIGN KEY (id_siswa) REFERENCES siswa(id),
29     FOREIGN KEY (id_hukuman) REFERENCES hukuman(id)
30 );

```

## **BAB III**

### **PENUTUP**

#### **3.1 Kesimpulan**

Sistem Skorsing Siswa yang dikembangkan oleh kelompok kami adalah aplikasi berbasis Python yang memanfaatkan database MySQL untuk mengelola data pelanggaran dan hukuman siswa secara efisien di lingkungan sekolah. Sistem ini memiliki berbagai fitur unggulan yang dirancang untuk mendukung pengelolaan data pelanggaran, seperti pencatatan poin, histori pelanggaran, dan pemberian sanksi sesuai kriteria yang telah ditentukan.

Fitur utama sistem ini meliputi:

1. **Pengelolaan Data Siswa:** Menambahkan, menampilkan, memperbarui, dan menghapus data siswa beserta histori pelanggaran mereka.
2. **Pengelolaan Hukuman dan Kategori Hukuman:** Menambahkan, menampilkan, dan mengatur hukuman berdasarkan kategori dan skor yang relevan.
3. **Pemberian Skorsing:** Memungkinkan administrator untuk memberikan sanksi kepada siswa yang melanggar, dengan otomatisasi pembaruan poin dan perubahan status siswa sesuai dengan aturan.
4. **Histori Skorsing:** Menyediakan pelacakan lengkap histori pelanggaran setiap siswa untuk memantau perilaku mereka.

Sistem ini dirancang untuk memastikan keadilan dan transparansi dalam pengelolaan pelanggaran di sekolah, serta memberikan kontrol penuh kepada administrator dalam memantau dan memberikan tindakan disiplin. Dengan fitur yang lengkap dan implementasi yang terstruktur, aplikasi ini diharapkan dapat membantu sekolah dalam menciptakan lingkungan yang lebih tertib dan terorganisir.

## **PEMBAGIAN TUGAS**

- 1. Calvin Darmawan Harun (5230411006)**
  - Membuat algoritma dan skenario sistem
  - Membuat BAB III Penutup
- 2. Dhafa Widy Pratama (5230411007)**
  - Membuat class diagram
  - Membuat BAB III Penutup
- 3. Taufik Nur Ardiansyah (5230411040)**
  - Menyusun laporan
  - Membuat deskripsi laporan
  - Membuat algoritma dan skenario sistem
  - Membuat database sistem
  - Membuat sistem skorsing
  - Membuat BAB II Pembahasan
  - Membuat BAB III Penutup

## DAFTAR PUSTAKA

- Rosita, D. (2016). IMPLEMENTASI KEBIJAKAN PENINGKATAN MUTU DI SMP N 1 NGAGLIK SLEMAN. *JURNAL SPEKTRUM ANALISIS KEBIJAKAN PENDIDIKAN*, V(3), 334-340.
- Sugiyarta, A., Jubaedi, A. D., & Pambudi, S. A. (2018, September). Sistem Informasi Skorsing Siswa Berbasis SMS Gateway Di SMP Informatika Kota Serang. *Jurnal Sistem Informasi*, 5(2), 34-39. DOI <https://doi.org/10.30656/jsii.v5i2.773>

## LAMPIRAN

