

BAB IV

HASIL DAN PEMBAHASAN

4.1 Analisis Data

Hasil pengumpulan data ini membahas temuan-temuan utama yang berkaitan dengan tujuan dan hipotesis penelitian. Hasil pengumpulan data dalam penelitian ini merupakan bukti konkret yang didapatkan dari pelaksanaan penelitian yang menggunakan beberapa teknik dan menghasilkan data berupa :

1. Hasil Wawancara

Wawancara dilakukan dengan narasumber seorang wakil kepala sekolah bidang kesiswaan sekaligus guru di SMA N 1 Kerinci yaitu bapak Sunardi Latif, S.Pd, ME. Berikut merupakan hasil dari wawancara yang dilakukan :

- a. SMA N 1 Kerinci masih menggunakan sistem pendidikan konvensional yaitu sebagai contohnya ialah proses input informasi, nilai dan data masih dilakukan secara manual, dan guru masih menuliskan nilai siswa di dalam kertas yang berisikan nama siswa. Kemudian, kertas tersebut diserahkan kepada bagian TU (Tata Usaha) agar diproses dan digabungkan dengan nilai lainnya yang diberikan oleh guru yang mengajar berbagai mata pelajaran.
- b. Siswa cenderung mengandalkan grup WhatsApp sebagai sumber utama untuk mengumpulkan Informasi mengenai sekolah, karena tidak adanya website khusus sekolah sebagai sumber Informasi.

2. Hasil Observasi

Pada observasi ditemukan proses input nilai yang masih konvensional yaitu ditulis pada selembar kertas dengan contoh pada gambar berikut :

NILAI EKONOMI XI SEMESTER 2		KELAS XI IPA 3	
No	Nama	Pengetahuan	Keterampilan
1	ADIT SAPUTRA	70	72
2	AGUNG SANJAYA	80	83
3	ALSYA TITANIA DESPITA	90	90
4	ANTONI		
5	ARMY KANAHMANI	91	91
6	DEORA FIRDA DARU	84	82
7	DILLA KARLINA	85	86
8	DINDA REFINA INSANI	90	90
9	FARA ADILAH	86	86
10	FIDRA FIBRIA HARVIRA	88	89
11	ILHAM SAPUTRA		
12	LAVIONA	90	90
13	M. AL-AZIZI ERPAN	90	90
14	MABU AFRANITA	89	87
15	M. HARAL		
16	M. HICANY		
17	M. NISUL HIDAYAT		
18	NADIA RAFIANTI	90	90
19	NABIDATUL SYLEHA	82	84
20	RIPA MARLINA ULANDARI	88	90
21	REVAL ANTONI	84	84
22	WANDA DESMIRA	75	75
23	WAHYU ERISON	75	78

DINDA REFINA INS

Gambar 4.1 Nilai yang masih konvensional

Dengan begitu didapatkan hasil data berupa data nama nama siswa, mata pelajaran, dan data guru.

4.2 Analisis Kebutuhan Pengguna

Analisis kebutuhan pengguna ini menjadi landasan penting dalam merancang sistem yang dapat memenuhi tuntutan serta memberikan pengalaman pengguna yang efisien dan efektif. Dengan memahami apa yang diinginkan dan dibutuhkan oleh para pengguna, hal ini dapat menghasilkan solusi yang sesuai dan dapat mendukung kelancaran proses akademik di SMA N 1 Kerinci. Berikut beberapa analisis kebutuhan pengguna :

1. Pengguna dapat mengakses jadwal pelajaran dan kegiatan ekstrakurikuler secara online.
2. Pengguna dapat melihat dan memperbarui informasi pribadi serta riwayat nilai UTS dan UAS.
3. Pengguna dapat menerima pemberitahuan tentang pengumuman penting atau berita.
4. Pengguna dapat cetak laporan nilai untuk kebutuhan arsip sekolah.
5. Pengguna dapat melakukan komunikasi online dengan sekolah melalui telp ataupun email.

4.3 Analisis Kebutuhan Sistem

Pada Aplikasi Sistem Informasi Akademik SMA N 1 KERINCI yang dibangun. Adapun analisis kebutuhan sistem, spesifikasi kebutuhan fungsional dan non fungsionalnya sebagai berikut :

a. Analisis Kebutuhan Sistem

Melalui analisis yang cermat terhadap kebutuhan sistem, dirancang solusi yang sesuai serta memastikan kelancaran dan keberhasilan implementasi sistem informasi akademik di SMA N 1 Kerinci sebagai berikut :

- 1) Sistem dapat menyediakan platform yang mudah diakses dan responsif untuk akses informasi akademik dari mana saja.
- 2) Sistem dapat mengelola dan menyimpan data siswa, termasuk informasi pribadi dan riwayat nilai secara aman dan terstruktur.
- 3) Sistem dapat menyediakan laporan yang berguna bagi pengguna, seperti laporan nilai akademik siswa.
- 4) Sistem dapat mengintegrasikan jadwal pelajaran dan berita ke dalam satu platform untuk memudahkan pengelolaan dan akses informasi.

b. Analisis Kebutuhan Fungsional

- 1) Kebutuhan *Login* pengguna yaitu berupa *username* dan *password*.
- 2) Tersedia 3 akun diantaranya yaitu Admin, Guru, dan Siswa
- 3) Akun Admin dapat mengelola semua data di aplikasi SIA
- 4) Akun Guru dapat mengelola data data siswa

- 5) Akun Siswa dapat memeriksa informasi yang telah diinput oleh Guru atau Admin serta dapat mengubah profil dari akun masing masing.

c. Analisis kebutuhan Non Fungsional

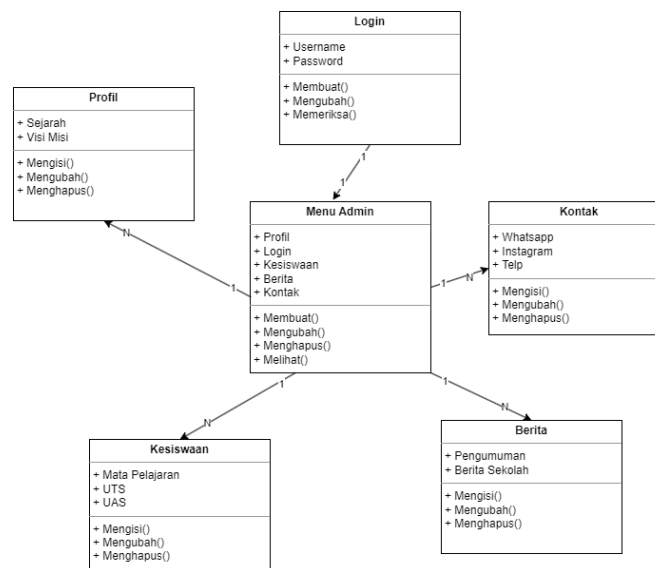
- 1) Sistem berbentuk website portal
- 2) Aplikasi memiliki kinerja yang cepat dan responsif, terutama saat mengakses data nilai dan informasi penting lainnya.
- 3) *User Interface* aplikasi intuitif, mudah digunakan, dan responsif, sehingga memudahkan siswa, dan guru untuk berinteraksi dengan sistem.
- 4) Aplikasi hanya berupa CRUD (*Create, Read, Update, dan Delete*)

4.4 User Requirement Analysis dan User Requirement Specification

Requirement Analysis dan Requirement Specification adalah salah satu langkah awal dalam siklus pengembangan SIA SMA N 1 KERINCI. Pada tahap ini, menjelaskan identifikasi kebutuhan atau persyaratan yang harus dipenuhi oleh SIA yang akan dibuat. Tahap ini bertujuan untuk mendefinisikan secara jelas dan lengkap apa yang diharapkan dari SIA tersebut, sehingga dapat menjadi dasar bagi seluruh proses pengembangan.

4.4.1 Class Diagram Menu Admin

Dalam **Gambar 4.2 Diagram menu admin** class diagram ini, ditampilkan struktur dan hubungan entitas yang terkait dengan peran admin dalam sistem. Diagram ini memberikan pandangan yang jelas tentang kelas-kelas, atribut, dan metode yang dimiliki oleh admin, serta interaksi yang dimungkinkan antara admin dan komponen lain dalam sistem.

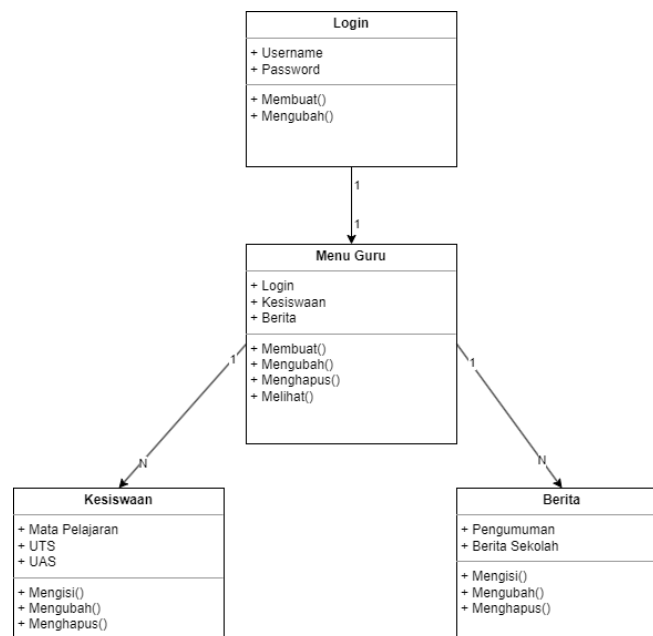


Gambar 4.2 Diagram menu admin

Pada **Gambar 4.2 Diagram menu admin** terdapat 5 class diagram yaitu Login, Profil, Kontak, Kesiswaan, dan Berita. Pada class diagram profil, kontak, berita, dan kesiswaan, admin dapat mengisi, mengubah, serta menghapus data data yang ada di website SIA. Sedangkan pada class diagram login, admin dapat membuat, mengubah, dan memeriksa data.

4.4.2 Class Diagram Menu Guru

Pada **Gambar 4.3 Diagram menu guru** ini, Berisi struktur kelas, atribut, dan metode yang terkait dengan peran guru dalam sistem. Diagram ini memperlihatkan bagaimana guru berinteraksi dengan entitas lainnya dalam sistem, serta memfasilitasi pemahaman tentang bagaimana guru menggunakan fungsionalitas yang tersedia untuk keperluan pengajaran dan manajemen data

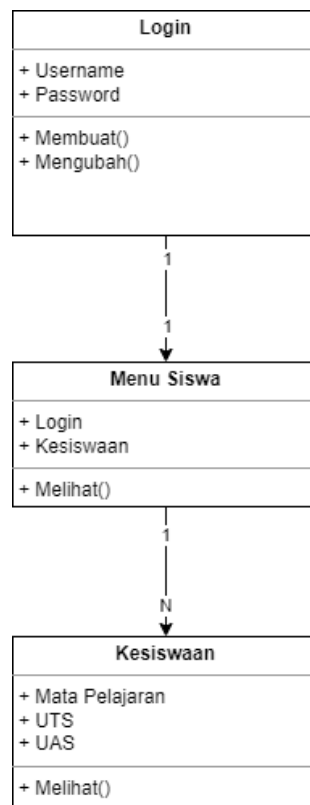


Gambar 4.3 Diagram menu guru

Pada **Gambar 4.3 Diagram menu guru** terdapat 3 class diagram yaitu Login, Kesiswaan, dan Berita. Pada class diagram kesiswaan dan berita, guru dapat mengisi, mengubah, dan menghapus data data di website SIA. Sedangkan pada class diagram login, guru dapat membuat dan mengubah data.

4.4.3 Class Diagram Menu Siswa

Diagram pada **Gambar 4.4 Diagram menu siswa** memberikan gambaran tentang bagaimana siswa berinteraksi dengan sistem, termasuk cara mereka mengakses informasi, dan memanfaatkan fungsionalitas yang tersedia untuk mendukung proses pembelajaran.



Gambar 4.4 Diagram menu siswa

Pada **Gambar 4.4 Diagram menu siswa** terdapat 2 class diagram yaitu Login, dan Kesiswaan. Pada class diagram kesiswaan, siswa dapat melihat data data di website SIA. Sedangkan pada class diagram login, siswa dapat membuat dan mengubah data.

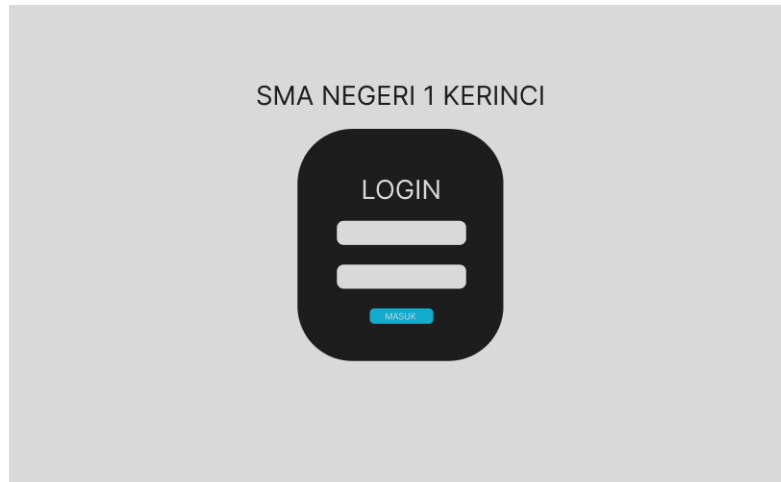
4.5 Software Requirement Specifiation

Pada tahap ini Sistem Informasi Akademik akan mencakup berbagai aspek, termasuk manajemen data siswa, guru, jadwal pelajaran, dan nilai. Sistem ini dirancang untuk mendukung proses administrasi dan manajemen akademik di SMA N 1 Kerinci. Fokus utama mencakup pencatatan data siswa, penjadwalan pelajaran, dan pelaporan nilai.

4.6 System/broad design (Logical design) dan Program/detailed design (Physical design)

Berikut adalah design dari aplikasi Sistem Informasi Akademik SMA N 1 KERINCI yang dapat dilihat dibawah ini :

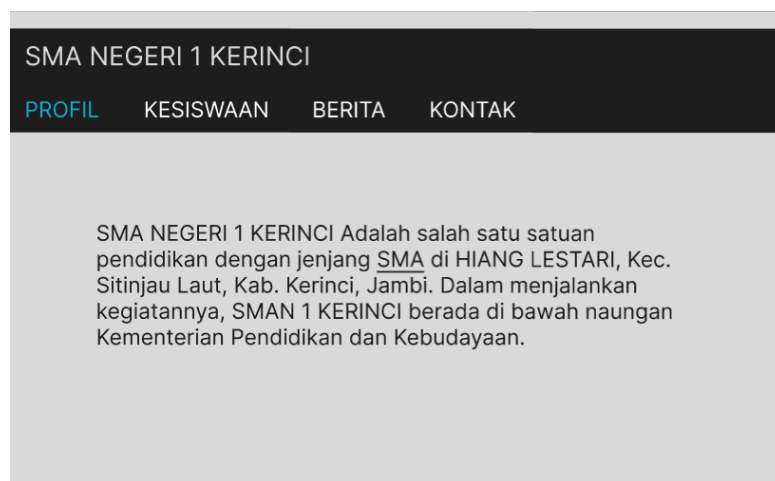
4.5.1 Halaman Login



Gambar 4. 5 Halaman Login

Pada **Gambar 4.5 Halaman Login**, menu login terdapat username dan password yang wajib diisi oleh admin, guru, maupun siswa jika ingin mengakses fitur fitur yang ada pada website SIA SMA N 1 KERINCI.

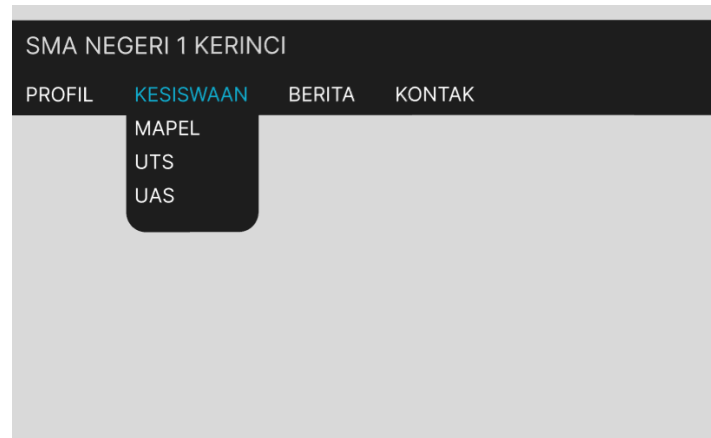
4.5.2 Halaman Profil



Gambar 4. 6 Halaman Profil

Pada **Gambar 4.6 Halaman Profil** diisi dengan Visi, Misi, dan Tujuan dari SMA

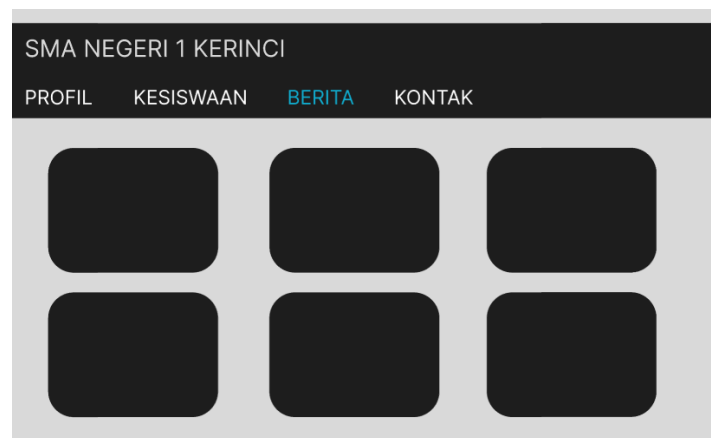
N 1 KERINCI



Gambar 4. 7 Halaman Kesiswaan

Pada **Gambar 4.7 Halaman Kesiswaan** berisi Mata Pelajaran, UTS, dan UAS dari siswa.

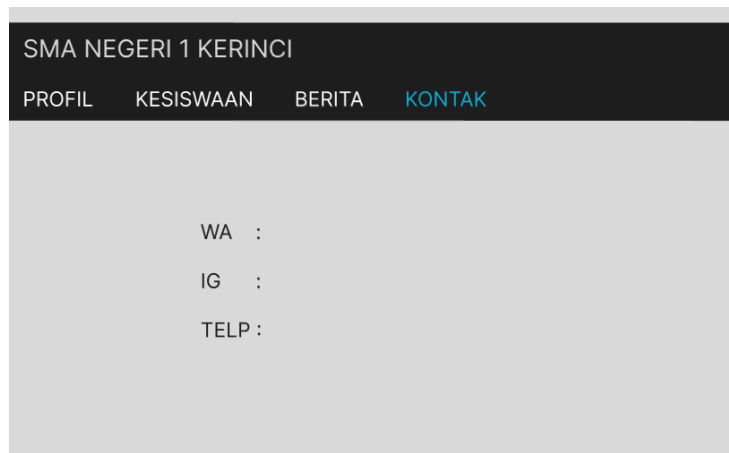
4.5.3 Halaman Berita



Gambar 4. 8 Halaman Berita

Pada **Gambar 4.8 Halaman Berita** berisi berita-berita yang telah diinput, kotak-kotak hitam di **Gambar 4.8 Halaman Berita** yaitu berita-berita yang telah diinput sebelumnya dan dapat diklik apabila ingin membaca berita secara sepenuhnya.

4.5.4 Halaman Kontak



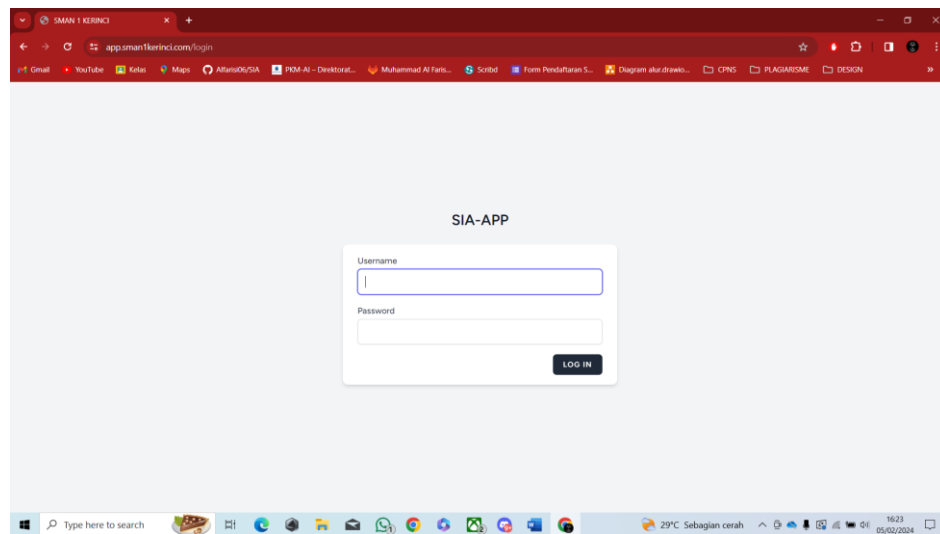
Gambar 4. 9 Halaman Kontak

Pada **Gambar 4.9 Halaman Kontak** berisi no telp. Dan sosial media dari SMA N 1 KERINCI, yang bertujuan untuk mempermudah siswa dalam menghubungi guru di SMA N 1 KERINCI.

4.7 Implementation

Proses implementasi website sistem informasi akademik (SIA) SMA N 1 Kerinci dilakukan untuk memastikan bahwa situs web dirancang dan dikembangkan dengan baik untuk memenuhi kebutuhan pengguna.

4.6.1 Bagian Login



Gambar 4.10 Login

Pada **Gambar 4.10 Login** terdapat menu login yang berisi username dan password yang wajib diisi untuk mengakses fitur fitur website.

```
public function store(LoginRequest $request): RedirectResponse
{
    $request->authenticate();

    $request->session()->regenerate();

    return redirect()->intended(RouteServiceProvider::HOME);
}
```

Kode Program 4. 1 Halaman login SIA SMA N 1 KERINCI

Pada **Kode Program 4.1 Halaman Login SIA SMAN 1 KERINCI** Username & Password diambil dari form website, kemudian di proses menggunakan authenticate.

Berikut versi lengkap dari function authenticate.

```
public function authenticate(): void
{
    $this->ensureIsNotRateLimited();

    if (! Auth::attempt($this->only('username', 'password'))) {
        RateLimiter::hit($this->throttleKey());

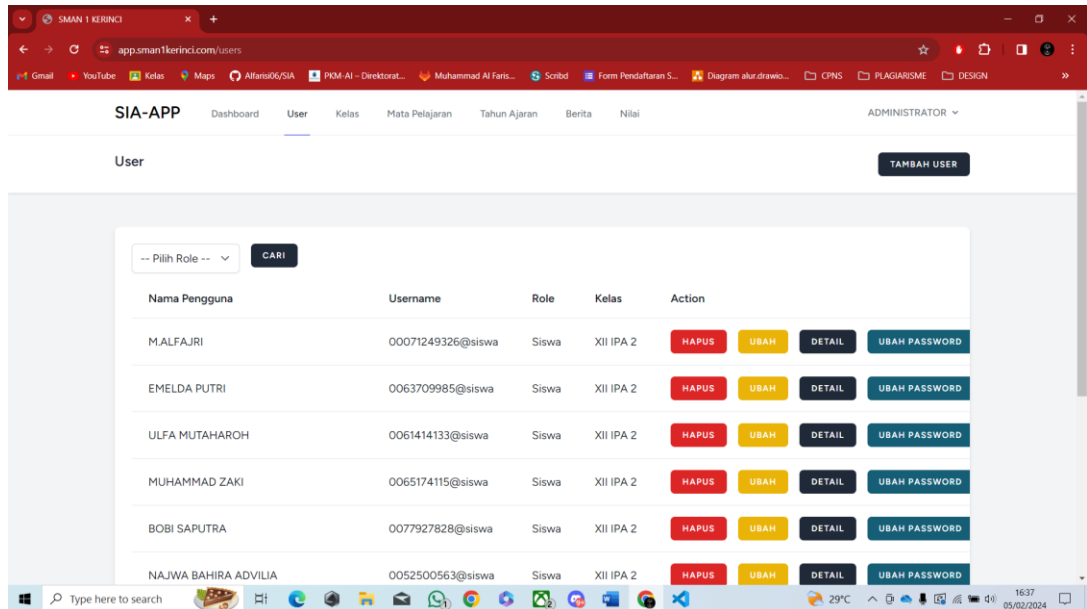
        throw ValidationException::withMessages([
            'username' => trans('auth.failed'),
        ]);
    }

    RateLimiter::clear($this->throttleKey());
}
```

Kode Program 4. 2 Halaman login function authenticate

Pada **Kode Program 4.2 Halaman login function authenticate** Jika username ditemukan di database maka akan dialihkan ke halaman dashboard.

4.6.2 Bagian User



Gambar 4. 11 Halaman User

Pada **Gambar 4.11 Halaman User** terdapat akun akun yang telah terdaftar diantaranya yaitu akun siswa dan akun guru.

```
public function getUsers(){
    $userQuery = User::latest();

    if(request()->filled('role')){
        $userQuery->where('role', request()->role);
    }

    return $userQuery->paginate(10);
}
```

Kode Program 4. 3 Halaman User function getUsers

Pada **Kode Program 4.3 Halaman User function getUsers** dijelaskan bahwa function getUsers bertujuan untuk mengambil semua data user dari database table user. Di bagian user ini admin dapat Hapus, ubah data, melihat detail setiap akun serta mengubah passwordnya.

```
public function getUsersById($id){
    return User::find($id);
}
```

Kode Program 4. 4 Halaman User function getUserById

Pada **Kode Program 4.4 Halaman User function getUsersById** berfungsi untuk mencari user sesuai id nya kemudian ditampilkan di halaman Edit.

```
public function store(StoreUsersRequest $request){
    User::create([
        'nama' => $request->nama,
        'email' => $request->email,
        'role' => $request->role,
        'username' => $request->username,
        'password' => $request->password,
        'nis' => $request->nis,
        'kelas_id' => $request->kelas_id,
        'gender' => $request->gender
    ]);

    return redirect()->route('users.index')->withSuccess('Data berhasil disimpan');
}
```

Kode Program 4. 5 Halaman User function store

Pada **Kode Program 4.5 Halaman User function store** dijelaskan untuk menyimpan request ke database user setelah berhasil akan dikembalikan ke halaman sebelumnya dengan notifikasi data berhasil disimpan.

```
public function update(UpdateUsersRequest $request, $id){
    $user = User::find($id);

    $user->nama = $request->nama;
    $user->email = $request->email;
    $user->role = $request->role;
    $user->username = $request->username;
    $user->nis = $request->nis ?? null;
    $user->kelas_id = $request->kelas_id;
    $user->gender = $request->gender;

    $user->save();
}
```



```

        return redirect()->route('users.index')->withSuccess('Data berhasil
        diubah');
    }

    public function destroy($id){
        User::destroy($id);

        return redirect()->route('users.index')->withSuccess('Data berhasil hapus');
    }

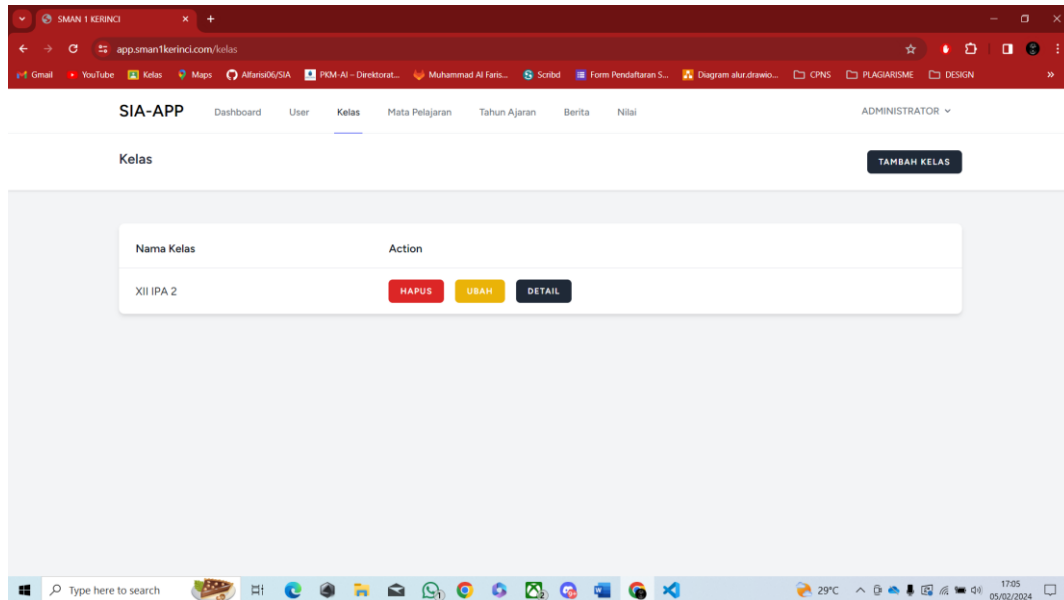
    public function changePasswordUser($payload, $id){
        return User::where('id', $id)->update([
            'password' => Hash::make($payload['password'])
        ]);
    }
}

```

Kode Program 4. 6 Halaman User function update dan destroy

Pada **Kode Program 4.6 Halaman User function update dan destroy** mengirim request dari form edit kemudian disimpan ke model user dengan method save. Jika berhasil maka akan diarahkan ke halaman sebelumnya dengan notifikasi data berhasil diubah/dihapus.

4.6.3 Bagian Kelas



Gambar 4. 12 Halaman Kelas

Pada **Gambar 4.12 Halaman Kelas** berisi kelas yang terdaftar di sistem Informasi akademik yang dapat kita hapus, ubah, serta melihat detail dari kelas tersebut.

```
public function getKelas(){
    return Kelas::latest()->paginate(10);
}

public function getAll(){
    return Kelas::all();
}
```

Kode Program 4. 7 Halaman Kelas function getKelas

Pada Kode **Program 4.7 Halaman Kelas function getKelas** digunakan untuk mengambil semua data kelas dari database table kelas.

```
public function getKelasById($id){
```

```

    return Kelas::find($id);
}

```

Kode Program 4. 8 Halaman Kelas function getKelasById

Pada **Kode Program 4.8 Halaman Kelas function getKelasById** berfungsi untuk mencari kelas sesuai id nya kemudian ditampilkan di halaman Kelas.

```

public function update(StoreKelasRequest $request, $id){
    $kelas = Kelas::find($id);

    $kelas->nama = $request->nama;

    $kelas->save();

    return redirect()->route('kelas.index')->withSuccess('Data berhasil diubah');
}

public function destroy($id){
    Kelas::destroy($id);

    return redirect()->route('kelas.index')->withSuccess('Data berhasil dihapus');
}

```

Kode Program 4. 9 Halaman Kelas function update

Pada **Kode Program 4.9 Halaman Kelas function update** public function update dan public function destroy mengirim request dari form edit kemudian disimpan ke model user dengan method save. Jika berhasil maka akan diarahkan ke halaman sebelumnya dengan notifikasi data berhasil diubah/dihapus.

4.7.4 Bagian Mata Pelajaran

Nama Mata Pelajaran	Kelas	Hari	Guru	Waktu Mulai	Waktu Selesai	Action
BIOLOGI	XII IPA 2	Sabtu	YULIANA, S.Pd.	11:50	13:10	HAPUS UBAH
MATEMATIKA PEMINATAN	XII IPA 2	Sabtu	DEVI AFIARMI, S.Pd.	10:25	11:50	HAPUS UBAH
EKONOMI	XII IPA 2	Sabtu	SUNARDI LATIF, S.Pd.,ME.	08:50	10:10	HAPUS UBAH
BAHASA INGGRIS	XII IPA 2	Sabtu	NING AYUNI, S.Pd.	07:30	08:50	HAPUS UBAH
BAHASA INDONESIA	XII IPA 2	Jumat	Dra.YULIA ERLISDA	09:45	11:05	HAPUS UBAH
SEJARAH INDONESIA	XII IPA 2	Jumat	SELMAYENI, S.Pd.	08:50	09:30	HAPUS UBAH

Gambar 4. 13 Halaman Mata Pelajaran

Pada **Gambar 4.13 Halaman Mata Pelajaran** berisi mata pelajaran yang telah diinput dan juga dapat dihapus, dan diubah.

```
public function getMapel(){
    return Mapel::latest()->get();
}

public function getMapelByUserId($id){
    return Mapel::where('user_id', $id)->latest()->get();
}

public function getMapelByKelasId($id){
    return Mapel::where('kelas_id', $id)->get();
}

public function getMapelByGuruid($id){
```

```

    return Mapel::where('guru_id', $id)->get();
}

public function getMapelById($id){
    return Mapel::find($id);
}

```

Kode Program 4. 10 Halaman Mata Pelajaran function getMapel

Pada **Kode Program 4.10 Halaman Mata Pelajaran function getMapel** Public Function getMapel digunakan untuk mengambil semua data Mapel dari database tabel mapel.

Public Function getMapelByUserId berfungsi untuk mencari mapel sesuai user id nya kemudian ditampilkan di halaman mapel.

Public Function getMapelByGuruId berfungsi untuk mencari mapel sesuai id gurunya kemudian ditampilkan di halaman mapel.

Public Function getMapelByUserId berfungsi untuk mencari mapel sesuai user id nya kemudian ditampilkan di halaman mapel.

Public Function getMapelById berfungsi untuk mencari mapel sesuai id nya kemudian ditampilkan di halaman mapel.

```

public function store(StoreMapelRequest $request){
    Mapel::create([
        'nama' => $request->nama,
        'kelas_id' => $request->kelas_id,
        'user_id' => $request->user_id,
        'hari' => $request->hari,
        'schedule_start_at' => $request->waktu_mulai,
    ])
}

```

```

'schedule_end_at' => $request->waktu_selesai,
]);

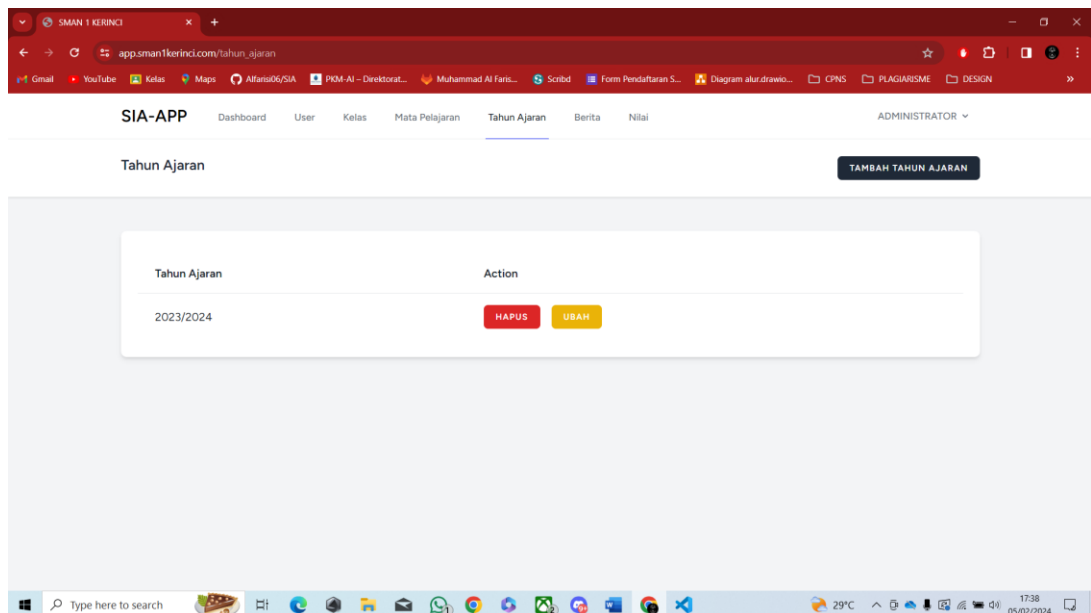
return redirect()->route('mapel.index')->withSuccess('Data berhasil disimpan');
}

```

Kode Program 4. 11 Halaman Mata Pelajaran function store

Pada **Kode Program 4.11 Halaman Mata Pelajaran function store** Public Function store mengirim request dari form tambah mapel kemudian disimpan ke model mapel dengan method save. Jika berhasil maka akan diarahkan ke halaman sebelumnya dengan notifikasi data berhasil disimpan.

4.7.5 Bagian Tahun Ajaran



Gambar 4. 14 Halaman Tahun Ajaran

Pada **Gambar 4.14 Halaman Tahun Ajaran** berisi tahun ajaran dari SMA N 1 KERINCI, yang dimana dapat dihapus dan diubah

```
public function getTahunAjaran(){
    return TahunAjaran::latest()->get();
}

public function getTahunAjaranById($id){
    return TahunAjaran::find($id);
}
```

Kode Program 4. 12 Halaman Tahun Ajaran function getTahunAjaran

Pada **Kode Program 4.12 Halaman Tahun Ajaran function getTahunAjaran** Public Function getTahunAjaran untuk mengambil semua data Tahun Ajaran dari database tabel Tahun Ajaran. Public Function getTahunAjaranById berfungsi untuk mencari Tahun ajaran sesuai user id nya kemudian ditampilkan di halaman Tahun Ajaran.

```
public function store(StoreTahunAjaranRequest $request){
    TahunAjaran::create([
        'nama' => $request->nama
    ]);

    return redirect()->route('tahun_ajaran.index')->withSuccess('Data berhasil
    disimpan');
}
```

Kode Program 4. 13 Halaman Tahun Ajaran function store

Pada **Kode Program 4.13 Halaman Tahun Ajaran function store** mengirim request dari form tambah tahun ajaran kemudian disimpan ke model tahun ajaran dengan method save. Jika berhasil maka akan diarahkan ke halaman sebelumnya dengan notifikasi data berhasil disimpan.

```
public function update(UpdateTahunAjaranRequest $request, $id){
    $tahunAjaran = TahunAjaran::find($id);
```

```

        $tahunAjaran->nama = $request->nama;

        $tahunAjaran->save();

        return redirect()->route('tahun_ajaran.index')->withSuccess('Data berhasil
        diubah');
    }

    public function destroy($id){
        TahunAjaran::destroy($id);

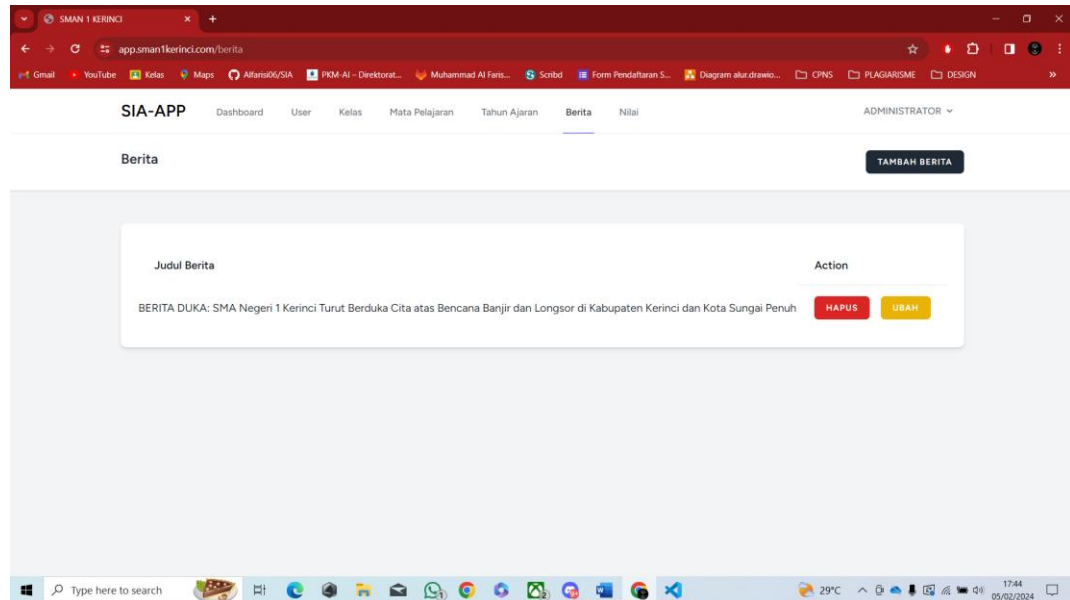
        return redirect()->route('tahun_ajaran.index')->withSuccess('Data berhasil
        dihapus');
    }

```

Kode Program 4. 14 Halaman Tahun Ajaran function update dan destroy

Pada **Kode Program 4.14 Halaman Tahun Ajaran function update dan destroy** mengirim request dari form hapus dan ubah kemudian disimpan ke model tahun ajaran dengan method save. Jika berhasil maka akan diarahkan ke halaman sebelumnya dengan notifikasi data berhasil diubah/dihapus.

4.6.6 Bagian Berita



Gambar 4. 15 Halaman Berita

Pada **Gambar 4.15 Halaman Berita** berisi berita-berita yang telah diinput dan dapat dihapus ataupun diubah.

```
public function getAll(){
    return Berita::all();
}

public function getThreeLatest(){
    return Berita::orderBy('created_at', 'desc')->take(3)->get();
}
```

Kode Program 4. 15 Halaman Berita function getAll

Pada **Kode Program 4.15 Halaman Berita function getAll** mengambil semua data Berita dari table berita. Public Function getThreeLatest mengambil 3 data berita terbaru melalui table berita.

```

public function storeData($payload, $filePath){
    return Berita::create([
        'judul' => $payload['judul'],
        'isi_berita' => $payload['isi_berita'],
        'cover' => $filePath
    ]);
}

```

Kode Program 4. 16 Halaman Berita function storeData

Pada **Kode Program 4.16 Halaman Berita function storeData** mengirim request dari form tambah berita kemudian disimpan ke model berita dengan method save.

```

public function findById($id){
    return Berita::find($id);
}

public function destroyById(Berita $news){
    return $news->delete();
}

public function updateDataById(Berita $news, $payload, $filePath){
    return $news->update([
        'judul' => $payload['judul'],
        'isi_berita' => $payload['isi_berita'],
        'cover' => $filePath ?? $news->cover
    ]);
}

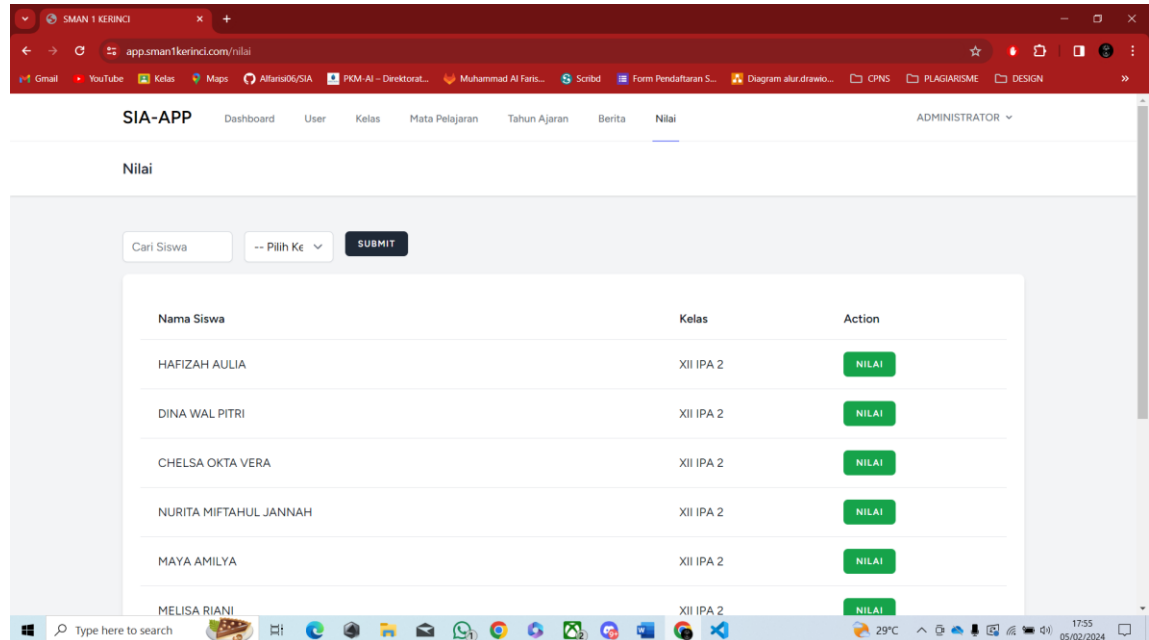
```

Kode Program 4. 17 Halaman Berita function

Pada **Kode Program 4.17 Halaman Berita function** Public Function FindById berfungsi untuk mencari berita sesuai id nya kemudian ditampilkan di halaman berita. Public Function updateDataById dan Public Function destroyById mengirim request dari form

hapus dan ubah kemudian disimpan ke model berita dengan method save. Jika berhasil maka data berhasil diubah/dihapus.

4.6.7 Bagian Nilai



Gambar 4. 16 Halaman Nilai

Pada **Gambar 4.16 Halaman Nilai** berisi nilai nilai siswa berdasarkan kelasnya.

```
public function getNilaiById($id){
    return Nilai::where('user_id', $id)->get();
}
```

Kode Program 4. 18 Halaman Nilai function getNilaiById

Pada **Kode Program 4.18 Halaman Nilai function getNilaiById** berfungsi untuk mencari nilai sesuai user_id nya kemudian ditampilkan di halaman nilai.

```

public function storeData($payload){

    return Nilai::create([

        'kelas_id' => $payload['kelas_id'],

        'user_id' => $payload['user_id'],

        'mapel_id' => $payload['mapel_id'],

        'tahun_ajaran_id' => $payload['tahun_ajaran_id'],

        'nilai_uts' => $payload['nilai_uts'],

        'nilai_uas' => $payload['nilai_uas']

    ]);

}

```

Kode Program 4. 19 Halaman Nilai function storeData

Pada **Kode Program 4.19 Halaman Nilai function storeData** mengirim request dari form tambah nilai kemudian disimpan ke model nilai dengan method save.

```

public function findById($id){

    return Nilai::find($id);

}

public function deleteById($id){

    return Nilai::destroy($id);

}

public function updateData($payload, $id){

    return Nilai::where('id', $id)->update([

        'user_id' => $payload['user_id'],

        'mapel_id' => $payload['mapel_id'],

        'tahun_ajaran_id' => $payload['tahun_ajaran_id'],

```

```

        'nilai_uts' => $payload['nilai_uts'],
        'nilai_uas' => $payload['nilai_uas']
    });
}

```

Kode Program 4. 20 Halaman Nilai function

Pada **Kode Program 4.20 Halaman Nilai function** Public Function FindById berfungsi untuk mencari nilai sesuai id nya kemudian ditampilkan di halaman nilai. Public Function updateData dan Public Function deleteById mengirim request dari form hapus dan ubah kemudian disimpan ke model nilai dengan method save. Jika berhasil maka data berhasil diubah/dihapus.

4.7.8 Program Testing Units dan Program Testing System

Pada tahap ini digunakan 2 metode testing dan pengujian yaitu diantaranya metode System Usability Scale (SUS) dan metode Blackbox.

a. System Usability Scale (SUS)

System Usability Scale (SUS) adalah metode penilaian kegunaan (usability) yang digunakan untuk menentukan seberapa mudah sistem bagi pengguna untuk digunakan, dengan scale sampai 1 – 5. Pada penelitian ini 38 responden yaitu siswa dan guru diminta untuk mengisi kuesioner sesuai ketentuan metode SUS yang memiliki beberapa pertanyaan seperti pada **Tabel 2.2 Pertanyaan SUS**.

Tabel 4. 1 Skor asli hasil tanggapan dari responden Full

RESPONDEN SUS	Nomor Pertanyaan										
	1	2	3	4	5	6	7	8	9	10	
1	5	2	4	2	5	2	4	2	5	2	NILAI
2	5	1	5	1	5	1	5	1	5	1	
3	5	1	5	1	5	1	5	1	5	1	
4	4	4	4	3	4	3	4	2	4	3	
5	5	3	5	2	4	4	4	2	4	2	
6	5	4	4	4	5	4	5	2	5	1	
7	4	1	5	1	3	2	4	2	5	2	
8	5	1	5	1	5	1	5	1	5	1	
9	5	1	5	1	5	1	5	2	5	1	
10	5	1	5	1	5	1	5	2	5	1	
11	4	1	5	2	5	1	5	1	5	2	
12	3	1	5	1	5	1	5	1	5	1	
13	4	2	5	2	4	2	4	2	4	2	
14	5	1	5	1	5	1	5	1	5	1	
15	3	2	4	3	4	3	3	2	3	2	
16	5	1	5	1	5	3	5	4	5	3	
17	4	1	4	1	5	1	5	1	5	2	
18	4	2	4	2	5	2	5	1	4	3	
19	4	2	5	2	5	2	4	1	4	3	
20	4	2	5	2	3	4	5	1	5	1	
21	1	2	1	1	1	2	5	2	5	2	
22	4	1	5	3	5	4	5	1	5	1	
23	4	2	4	1	4	2	4	2	5	1	
24	5	2	4	1	4	1	5	1	5	1	
25	4	2	5	3	5	2	4	2	3	1	
26	5	1	5	5	5	1	5	1	5	1	
27	4	2	4	3	4	2	5	1	4	3	
28	5	1	5	3	5	1	5	1	5	1	
29	4	1	5	2	5	1	5	1	5	3	
30	5	1	5	1	5	1	5	1	5	1	
31	5	1	5	1	5	3	5	3	4	2	
32	4	2	4	2	4	3	5	1	4	1	
33	3	2	4	3	4	2	4	1	3	2	
34	3	2	4	2	4	2	5	2	5	2	
35	3	1	5	1	5	3	5	1	5	3	
36	3	3	4	3	4	3	5	2	5	2	
37	2	2	5	2	4	2	5	2	4	2	
38	3	2	4	2	4	3	4	2	5	2	

Tabel 4. 2 Skor setelah diolah dengan metode SUS (full)

RESPONDEN SUS	Nomor Pertanyaan										Total	Total * 2,5
	1	2	3	4	5	6	7	8	9	10		
1	4	3	3	3	4	3	3	3	4	3	33	82,5
2	4	4	4	4	4	4	4	4	4	4	40	100
3	4	4	4	4	4	4	4	4	4	4	40	100
4	3	1	3	2	3	2	3	3	3	2	25	62,5
5	4	2	4	3	3	1	3	3	3	3	29	72,5
6	4	1	3	1	4	1	4	3	4	4	29	72,5
7	3	4	4	4	2	3	3	3	4	3	33	82,5
8	4	4	4	4	4	4	4	4	4	4	40	100
9	4	4	4	4	4	4	4	3	4	4	39	97,5
10	4	4	4	4	4	4	4	3	4	4	39	97,5
11	3	4	4	3	4	4	4	4	4	3	37	92,5
12	2	4	4	4	4	4	4	4	4	4	38	95
13	3	3	4	3	3	3	3	3	3	3	31	77,5
14	4	4	4	4	4	4	4	4	4	4	40	100
15	2	3	3	2	3	2	2	3	2	3	25	62,5
16	4	4	4	4	4	2	4	1	4	2	33	82,5
17	3	4	3	4	4	4	4	4	4	3	37	92,5
18	3	3	3	3	4	3	4	4	3	2	32	80
19	3	3	4	3	4	3	3	4	3	2	32	80
20	3	3	4	3	2	1	4	4	4	4	32	80
21	0	3	0	4	0	3	4	3	4	3	24	60
22	3	4	4	2	4	1	4	4	4	4	34	85
23	3	3	3	4	3	3	3	3	4	4	33	82,5
24	4	3	3	4	3	4	4	4	4	4	37	92,5
25	3	3	4	2	4	3	3	3	2	4	31	77,5
26	4	4	4	0	4	4	4	4	4	4	36	90
27	3	3	3	2	3	3	4	4	3	2	30	75
28	4	4	4	2	4	4	4	4	4	4	38	95
29	3	4	4	3	4	4	4	4	4	2	36	90
30	4	4	4	4	4	4	4	4	4	4	40	100
31	4	4	4	4	4	2	4	2	3	3	34	85
32	3	3	3	3	3	2	4	4	3	4	32	80
33	2	3	3	2	3	3	3	4	2	3	28	70
34	2	3	3	3	3	3	4	3	4	3	31	77,5
35	2	4	4	4	4	2	4	4	4	2	34	85
36	2	2	3	2	3	2	4	3	4	3	28	70
37	1	3	4	3	3	3	4	3	3	3	30	75
38	2	3	3	3	3	2	3	3	4	3	29	72,5

	Jumlah	3172,5
	Hasil Akhir	83

Berdasarkan rumus perhitungan SUS pada **Gambar 2.1 Rumus Skor SUS** mendapatkan hasil akhir skor yaitu 83 setelah itu, dapat dibuat kesimpulan bahwa Sistem Informasi Akademik SMA N 1 KERINCI yang berbentuk Website tersebut dapat dikatakan lulus pengujian kelayakan, mudah dalam penggunaan dan mudah dipahami.

b. Blackbox

Blackbox testing adalah pengujian yang memeriksa eksekusi berdasarkan data uji dan memastikan bahwa software berfungsi dengan baik. Teknik pengujian yang digunakan adalah Equivalence Partitions, Boundary Value Analysis, All-pair, Decision Table, State Transition, Error Guessing, dan Use Case. Bentuk pengujian sistemnya yaitu dengan bentuk pertanyaan dalam Gform yang diisi oleh salah satu guru SMA N 1 KERINCI yaitu bapak Sunardi Latif, S.Pd, ME. dengan temuan sebagai berikut :

Tabel 4. 3 Hasil Pengujian Sistem

No	Pengujian	Pertanyaan	Lolos / Tidak Lolos
1.	Equivalence Partitions	<p>Bagaimana sistem memproses nilai valid / tidak valid pada proses input nilai</p> <p>lolos : jika hasil pemrosesan menghasilkan output yang berbeda</p> <p>contoh : input nilai = '12dqwe', output = error</p>	LOLOS

2.	Boundary Value Analysis	<p>Bagaimana sistem memproses nilai yang memiliki rentang batas</p> <p>lolos : jika output = error, saat input melebihi batas yang telah ditetapkan</p> <p>contoh : input nilai = 300, output = error</p>	LOLOS
3.	All-pair	<p>Bagaimana sistem memproses kecocokan seluruh data yang disajikan</p> <p>lolos : jika sistem berhasil memasangkan seluruh data yang tersedia</p> <p>contoh : input mata pelajaran = "PJOK" "AKMALUDDIN S.Pd.", output = success</p>	LOLOS
4.	Decision Table	<p>Bagaimana sistem memproses lebih dari 1 input pada form login</p> <p>lolos : jika lebih dari 1 input memiliki output yang sesuai</p> <p>contoh : input username & password = 'test' 'test' (username / password salah), output = False</p>	LOLOS
5.	State Transition	<p>Bagaimana sistem memproses suatu kondisi tertentu</p>	LOLOS

		lolos : jika output berubah mengikuti suatu kondisi seperti pada form login contoh : input username dan password = *3 kali salah*, output = block	
6.	Error Guessing	Menebak input yang dapat menyebabkan error lolos : jika output sesuai dengan tebakan contoh : input username dan password di form login = 'ascasdas','1111' output = error	LOLOS
7.	Use Case	Bagaimana sistem merespon terhadap input tertentu dan mengevaluasi output lolos : jika output sesuai dengan harapan contoh : input username dan password di form login = 'ascasdas','1111' output = error	LOLOS

Hasil tes blackbox pada **Tabel 4.3 Hasil Pengujian Sistem** menunjukkan bahwa sistem beroperasi dengan lancar. Maka dapat disimpulkan bahwa Sistem Informasi Akademik SMA N 1 KERINCI yang berbentuk Website tersebut dapat dan layak digunakan.

4.8 Program Use

Pada tahap ini Guru, dan Siswa diedukasi tata cara penggunaan website melalui media PDF dengan judul “PANDUAN PENGGUNAAN WEBSITE” yang telah dibuat, di PANDUAN PENGGUNAAN WEBSITE dijelaskan bagaimana setiap akun dapat mengakses sesuai dengan level nya masing masing. Sesuai dengan penjelasan diagram di point **4.3. *User Requirement Analysis dan User Requirement Specification.***

4.9 Software Maintenance

Dengan metode fountain, proses maintenance software di website Sistem Informasi Akademik SMA N 1 Kerinci dapat dilakukan dengan lebih sistematis dan terstruktur. Ini membantu memastikan bahwa perbaikan yang diperlukan dapat dilakukan dengan tepat dan efisien, sehingga website dapat terus berfungsi dengan baik dan memenuhi kebutuhan pengguna.