**Sistem Informasi Pengolahan Nilai Harian SDN 1 Warga Mulya Berbasis Web**

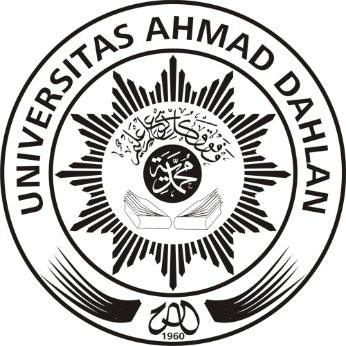
Rui Faqri

1800016004

Skripsi diajukan kepada   
Fakultas Sains dan Teknologi Terapan   
sebagai bagian persyaratan untuk meraih derajat

Sarjana ****Komputer****

pada Program Studi Sistem Informasi



Universitas Ahmad Dahlan   
Yogyakarta

Oktober, 2022

# **Daftar Isi**

# Daftar Gambar

# Daftar Tabel

# Bab 1

# Pendahuluan

## **1.1 Latar Belakang**

Keberadaan teknologi kini telah menjadi kebutuhan yang menunjang bagi kehidupan manusia di era modern ini. Hal tersebut dapat dilihat dengan dibutuhkannya sebuah sistem untuk mendukung aktifitas manusia sehari-hari. Pertukaran data yang cepat, efisien dan akurat merupakan keunggulan teknologi modern. Metode ini dapat diaplikasikan untuk mendapatkan informasi atau sumber daya dari berbagai sumber data. Salah satu bentuk penerapan teknologi berada pada sistem informasi, diantara teknologi sistem informasi yaitu website, aplikasi mobile dan lain-lain (Kurniawan, 2019). Dalam setiap sistem informasi tersebut memiliki struktur yang berbeda-beda diantaranya bahasa pemrograman atau framework (Ramirez, 2019) yang digunakan, database (Khan, 2019), UI/UX (Kyi, 2016), dan lain-lain (Utomo, 2020). Sistem informasi mampu membuat berbagai pekerjaan menjadi lebih efektif dan efisien (Komalasari, 2018).

Pada SDN 1 Warga Mulya memang belum memiliki aplikasi yang praktis untuk pengolahan data nilai harian siswa. Pengolahan dilakukan dengan sistem manual atau konvensional yaitu guru kelas jika melaporkan nilai harian hasil belajar siswa selama satu tema per bulan (Kurikulum 2013) harus menuliskan dalam lembar penilaian hasil belajar siswa yang sudah disediakan oleh sekolah berdasarkan mata pelajaran yang diampu. Sedangkan jika guru wali kelas membutuhkan nilai-nilai siswanya untuk menghitung nilai rata-rata akhir tema, maka harus datang mencari satu persatu tumpukan lembaran tersebut. Apabila data nilai belum ditemukan guru wali kelas harus mencari personel guru mata pelajaran yang mengerjakan nilai bersangkutan. Dalam hal penghitungan nilai harian per tema beberapa wali kelas memang sudah memanfaatkan aplikasi *Microsoft Excel*. Namun, untuk beberapa guru yang berusia tidak muda lagi harus menghitung dengan kalkulator manual. Sistem seperti ini sangatlah lamban, belum efektif, serta mempersulit kerja guru mata pelajaran maupun guru wali kelas dalam mengontrol nilai siswanya, bahkan kehilangan data rekapan nilai rentan sekali terjadi.

Manajemen sistem yang berjalan saat ini di sekolah harus diperbaiki, sehingga diperlukan pengembangan sistem untuk pengolahan data nilai harian hasil belajar siswa sebagai pengganti sistem yang lama. Guru mata pelajaran cukup memasukkan nilai-nilai siswanya kedalam sistem informasi pengolah data nilai harian hasil belajar siswa tersebut dengan komputer atau laptop mereka masing-masing di sekolah dan secara otomatis data yang telah dimasukkan akan tersimpan dalam database. Data yang telah tersimpan, secara otomatis akan membentuk nilai rata-rata per tema yang kemudian akan dilihat, disimpan maupun dicetak oleh guru wali kelas sebagai acuan untuk penulisan pada laporan hasil belajar siswa atau raport. Sistem informasi ini juga digunakan oleh guru sebagai media informasi hasil belajar anak ke pada orang tua.

Berdasarkan penjelasan dan permasalahan yang ditemukan, maka penelitian ini mengajukan judul “**Sistem Informasi Pengolahan Nilai Harian SDN 1 Warga Mulya Berbasis Web”.**

## **1.2 Identifikasi Masalah**

Berdasarkan latar belakang yang telah dijabarkan di atas, maka permasalahan yang dapat diangkat dalam penelitian ini adalah sebagai berikut:

1. Terdapat sistem pengolahan data nilai harian hasil belajar siswa yang berjalan di sekolah belum efektif, karena proses pengolahan data nilai harian hasil belajar siswa diolah oleh masing-masing guru dengan cara yang berbeda.
2. Proses memasukkan nilai yang dilakukan secara manual dapat menyita waktu dan tenaga yang banyak, sehingga kesalahan dalam penghitungan nilai dan keterlambatan dalam pengumpulan nilai rentan terjadi.
3. Akibat pengelolaan nilai yang berbeda-beda, guru mengalami kesulitan mengelola nilai akhir.

## **1.3 Ruang Lingkup**

Berdasarkan penjelasan pada latar belakang sebelumnya, dan memfokuskan pembahasan pada masalah yang dibahas, maka dibuat ruang lingkup yaitu:

1. Pembahasan hanya pada perancangan dan pembuatan sistem informasi yang khusus menangani pengolahan nilai harian siswa hasil belajar berbasis Kurikulum 2013.
2. Sistem ini berisi data masukan yang meliputi data siswa, data kelas, data mata pelajaran, data guru, nilai harian siswa, dan nilai tugas.
3. System tersebut dapat digunakan oleh guru mata pelajaran dan guru wali kelas. Dimana guru mata pelajaran dan guru wali dapat menambahkan data nilai harian sesuai dengan mata pelajaran yang di ampu.
4. Metode yang digunakan pada penelitian ini adalah RAD (Rapid Application Development).
5. System akan dikembangkan menggunakan framework Angular dan menggunakan database MySQL.

## **1.4 Rumusan Masalah**

Berdasarkan latar belakang yang dijelaskan diatas, maka rumusan masalah yang dapat diajukan dalam penelitian ini adalah bagaimana merancang dan membangun sistem informasi untuk memenuhi kebutuhan pengelolaan nilai harian siswa SDN 1 Warga Mulya berbasis web?

## **1.5 Tujuan Penelitian**

Tujuan yang hendak dicapai pada penelitian ini adalah untuk merancang dan membangun sistem informasi untuk memenuhi kebutuhan pengelolaan nilai harian SDN 1 Warga Mulya berbasis web.

## **1.6 Manfaat Penelitian**

Manfaat dari penelitian ini adalah sebagai berikut:

1. Memberikan kontribusi positif bagi lembaga pendidikan, khususnya bagi sekolah dalam membantu mengolah nilai harian hasil belajar siswa secara efektif dan efisien.
2. Pengembangan Sistem Informasi pengelolaan nilai harian ini diharapkan dapat mempermudah dalam melakukan penilaian oleh guru.
3. Hasil penilitian ini diharapkan dapat menjadi referensi yang bermanfaat bagi dunia akademik khususnya dalam penelitian yang dilakukan oleh peneliti tentang pengembangan sistem informasi pengelolaan nilai.

# Bab 2

# Kajian Teori

## **2.1 Kajian Pustaka Terdahulu**

Pertama, penelitian sebelumnya oleh (Supriyanta, 2018), di SMP Muhammadiyah Imogiri Yogyakarta memerlukan sebuah sistem untuk menunjang sekaligus mempermudah pengolahan nilai siswa. Tujuannya agar tercipta sistem informasi berbasis web yang diimplementasi untuk pengolahan nilai dan mengatasi sulitnya penyampaian informasi ulangan harian, tugas, ulangan tengah semester, ulangan akhir semester. Berdasarkan penelitian secara observasi, studi pustaka, dan wawancara yang telah dilakukan, maka dirancanglah sebuah sistem informasi pengolahan nilai siswa menggunakan metode waterfall yang menjadi sebuah solusi guna menuntaskan masalah yang ada. Sistem informasi pengolahan nilai siswa memudahkan guru, wali kelas, maupun bagian pengajaran dalam pengolahan nilai siswa serta menunjang proses kegiatan belajar mengajar.

Kedua, penelitian sebelumnya oleh (Haerani, 2019), Permasalahan yang terjadi dalam pengolahan nilai raport di MA Al-Kanawiyah Cikulur Lebak Banten saat ini masih menggunakan Aplikasi Raport Digital (ARD) dan penginputan nilai siswa masih area lokal. Penelitian ini bertujuan untuk membangun suatu system informasi nilai yang mempermudah pengecekan, pencatatan dan laporan data nilai siswa yang terkomputerisasi. Sistem ini bekerja memasukan dan menyimpan data, proses penilaian siswa serta laporan nilai akhir siswa sehingga lebih mudah mengetahui informasi nilai siswa yang akan disampaikan oleh pihak sekolah. Penelitian telah mengahasilkan sebuah system pengolahan nilai yang membantu kerja dari administrasi dan wali kelas dan dapat mempermudah pengguna untuk melakukan proses pengolahan nilaiagar pengelolaan nilai dapat diolah lebih efisien dan efektif , sehingga bisa langsung diakses serta informasi nilai dapat tersampaikan dengan baik. Selain itu antarmuka system diimplementasikan sesuai dengan tampilan raport yang sudah ada sehingga system dapat digunakan dengan mudah dan menghasilkan perhitungan nilai yang akurat.

Ketiga, penelitian sebelumnya oleh (Mentari, 2017), di SMP Negeri 1 Karangrayung dalam mengolah data nilai hasil belajar siswa masih menggunakan sistem manual yaitu mengerjakan dengan Microsoft excel kemudian dikumpulkan ke bagian kurikulum baru setelahnya diberikan kepada wali kelas dalam bentuk lembaran kertas. Tujuan dari penelitian ini adalah untuk mengembangkan sistem informasi pengolahan data nilai hasil belajar siswa berbasis web, pengujian sistem berdasarkan uji blackbox, uji portability dan uji efficiency, dan hasil implementasi sistem informasi pengolahan data nilai hasil belajar siswa berbasis web dengan angket uji pengguna. Penelitian ini menghasilkan sebuah sistem informasi yang dibuat menggunakan bahasa pemrograman PHP dan MySQL sebagai databasenya. Hasil yang didapatkan dari pengujian black box atau fungsionalitas pada sistem menunjukkan masing-masing test case adalah valid, uji portability yang memiliki persentase kelayakan sebesar 95%, dan uji efficiency dengan skor performance 72%. Hasil implementasi dari sistem melalui uji respon pengguna menunjukkan bahwa SIOLDAN sangat baik sebagai sistem informasi pengolahan data nilai hasil belajar siswa.

Keempat, penelitian sebelumnya oleh (Hidayat, 2021), Sistem pengolahan rapor pada SD Islam Imam Syafi’i Jember masih belum terkomputerisasi, yaitu mengola nilai rapor masih menggunakan Microsot Excel dan menghitungnya masih dengan cara manual, sehingga bisa terjadi kesalahan dalam perhitungan dan proses pengolahan rapor membutuhkan waktu yang lama. Dengan adanya sistem informasi rapor online ini, diharapkan dapat mempermudah pihak sekolah dalam mengola rapor dengan cara yang lebih hemat waktu dan tenaga, dan menghindari kesalahan perhitungan nilai rapor. Pada penulisan ini penulis menggunakan metode Rapid Aplication Development (RAD). Aplikasi yang dibuat menggunakan Codeigniter 3 untuk framework, PHP untuk bahasa pemrograman, MySQL untuk database dan Visual Studio Code sebagai tools membantu pembuatan sistem informasi tersebut. Sistem ini membantu dalam proses pengolahan rapor sampai mencetak hasil rapor yang tersampaikan dengan baik dan cepat.

Kelima, penelitian sebelumnya oleh (Abdussomad, 2019), DTA Nurul Islam Karawang merupakan sekolah Yayasan Islam Alhasan yaitu lembaga setingkat sekolah dasar yang beralamat di Bubulak Sauyunan, Rt. 04/08 Kel. Tanjungpura, Kec. Karawang Barat, Kab. Karawang dan memiliki 4 tingkatan kelas yaitu dari kelas 1 hingga kelas 4, dimana dalam pengolahan nilai akademiknya masih menggunakan sistem manual. Akibat dari hal tersebut banyak kendala yang dihadapi oleh pihak sekolah dalam urusan pengolahan nilai akademik seperti waktu yang dibutuhkan dalam mencari atau memasukan data nilai siswa memerlukan waktu yang lama, akurasi data yang dihasilkan dari nilai siswa terkadang mengalami kesalahan dikarenakan guru harus menghitung nilai secara manual. Untuk memudahkan admin, guru serta siswa DTA Nurul Islam dalam pencarian dan pengolahan data nilai, maka penulis membangun Sistem Pengolahan Nilai dengan menggunakan pemrograman berbasis web yaitu CodeIgniter dengan database MySQL. Sistem informasi akademik berbasis web ini dapat membantu pengolahan dan pengarsipan data akademik yaitu: data siswa, data guru, data kelas, nilai siswa.

**Tabel 2.1** Kajian Peneliti Terdahulu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Kriteria** | **Kajian 1** | **Kajian 2** | **Kajian 3** | **Kajian 4** | **Kajian 5** |
| **Peneliti** | Dyah Wahyu Mentari | Supriyanta, Elsa Murbiastuti | Reni Haerani, Robiyanto | Abdussomad, Robi Sopandi, Eriska Sinurat, Miranti | Nurman Hidayat, Kusuma Hati |
| **Tahun** | 2017 | 2018 | 2019 | 2019 | 2021 |
| **Judul** | Sistem Informasi Pengolahan Data Nilai  Hasil Belajar Siswa Pada Smp Negeri 1  Karangrayung Kabupaten Grobogan  Berbasis Web | Sistem Informasi Pengolahan Nilai Siswa Berbasis Web  Pada SMP Muhammadiyah Imogiri Yogyakarta | Sistem Informasi Pengolahan  Data Nilai Siswa Berbasis Web | Sistem Pengolahan Nilai Siswa Pada Data Nurul Islam  Karawang | Penerapan Metode Rapid Application Development (RAD)  dalam Rancang Bangun Sistem Informasi  Rapor Online (SIRALINE) |
| **Objek** | Smp Negeri 1  Karangrayung Kabupaten Grobogan | SMP Muhammadiyah Imogiri Yogyakarta | MA Al-Kanawiyah Cikulur Lebak Banten | DTA Nurul Islam Karawang | SD Islam Imam Safi’s Jember |
| **Metode/ Teori yang digunakan** | Waterfall | Waterfall | Waterfall | Waterfall | RAD |
| **Hasil** | Sistem pengelolaan nilai setelah dilakukan testing dan uji pengguna kepada 30 responden, dihasilkan bahwa sistem layak untuk digunakan. Dan uji coba menggunakan testing tools Yslow mendapatkan nilai C (cukup). | Telah berhasil dibuat sebuah website  sistem informasi pengolahan nilai siswa  yang diperuntukan sebagai sarana dalam  kegiatan pengolahan nilai siswa sampai  menghasilkan nilai raport di SMP  Muhammadiyah Imogiri Yogyakarta. Struktur navigasi yang digunakan yaitu  struktur navigasi komposit karena  pengguna dapat melakukan penjelasan  lebih bebas. | Sistem informasi yang baru ini akan memudahkan  bagian pengolahan nilai dalam membuat laporan  sesuai dengan kebutuhan, penyajian informasi yang lebih akurat dan relevan, menghemat waktu pencatatan nilai, mudah untuk merekap nilai bulanan sesuai kebutuhan manajemen yang  diterapkan oleh MA Al-Kanawiyah Cikulur  Lebak Banten. | Telah dihasilkan sebuah sistem informasi pengolahan nilai rapor. Sistem informasi akademik berbasis web ini dapat membantu pengolahan dan pengarsipan  data akademik yaitu: data siswa, data guru, data kelas, nilai siswa. Sistem informasi tersebut  dapat menyajikan laporan  pengelolaan data siswa, serta laporan pengelolaan nilai siswa dengan cepat. | Sistem dapat memberikan informasi yang dibutuhkan secara  cepat dan tepat. Sistem informasi membantu  pengolahan dan pengarsipkan data rapor. Sistem mempermudah proses pengolahan rapor yang sebelumnya  prosesnya masih manual menggunakan file excel. |

Dari **Tabel 2.1** diatas dapat disimpulkan bahwa pengembangan perangkat lunak tidak dapat dilakukan sembaran atau asal jadi saja. Kita perlu memperhatikan beberapa aspek agar pengembangan perangkat lunak dapat berjalan dengan baik dan sesuai dengan kebutuhan kita. Dalam pengembangan perangkat lunak terdapat metodologi yang dapat digunakan seperti Agile, Water Fall, Lean Development, TDD (Test- Driven Development), BDD (Behavior-Driven Development), T-BDD (Test Behavior- Driven Development) dan banyak lagi yang telah diterapkan sebagai pedoman untuk proses pengembangan perangkat lunak. Untuk penelitian ini kita akan membahas metode RAD (Rapid Aplication Development), dalam beberapa penelitian dan studi kasus yang berbeda. Pada penelitian pertama objek yang digunakan adalah SMP Negeri 1 Karangrayung Kabupaten Grobogan, penelitian bertujuan untuk membuat system untuk mengolah data nilai hasil belajar siswa dan akan di uji menggunakan uji black box, uji portability, uji efficiency dan uji respon pengguna. Pada penelitian kedua objek yang digunakan adalah SMP Muhammadiyah Imogiri Yogyakarta, Sistem informasi pengolahan nilai siswa ini memudahkan guru, wali kelas, siswa serta bagian pengajaran untuk mengelola nilai harian, nilai tugas, nilai UTS serta nilai UAS. Struktur navigasi yang digunakan adalah struktur navigasi komposit karena pengguna dapat melakukan penjelasan lebih bebas dan metode yang digunakan adalah waterfall. Pada penelitian ketiga objek yang digunakan adalah MA Al-Kanawiyah Cikulur Lebak Banten, Sistem informasi pengolahan nilai mempermudah dalam membuat laporan sesuai dengan kebutuhan, penyajian informasi yang lebih akurat dan relevan, menghemat waktu pencatatan nilai, mudah untuk merekap nilai bulanan sesuai kebutuhan manajemen yang diterapkan oleh MA Al-Kanawiyah Cikulur Lebak Banten. Pada penelitian keempat objek yang digunakan adalah DTA Nurul Islam Karawang, Sistem informasi berbasis web di DTA Nurul Islam Karawang Barat dapat menyajikan laporan-laporan seperti laporan pengelolaan data siswa, serta laporan pengelolaan nilai siswa dengan cepat. Pada penelitian kelima objek yang digunakan adalah SD Islam Imam Safi’s Jember, sistem dapat memberikan informasi yang dibutuhkan secara cepat dan tepat. Sistem informasi membantu pengolahan dan pengarsipkan data rapor. System mampu mengolah dan mengarsipkan data rapor yaitu data sekolah, data guru, data wali kelas, data siswa, data akademik dan nilai siswa.

Berdasarkan penelitian terdahulu sudah banyak tim pengembang yang mengolah sebuah sistem nilai atau rapor sebagai topik penelitian dan menggunakan berbagai macam metode seperti RAD, Waterfall, REST. Agile, dll. Metode pengembangan perangkat lunak RAD (Rapid Aplication Development) memberikan dampak yang bagus kepada para pengembang. Tujuan yang hendak dicapai dari penelitian ini adalah merancang dan membangun sistem informasi untuk memenuhi kebutuhan pengelolaan nilai harian SDN 1 Warga Mulya berbasis web.

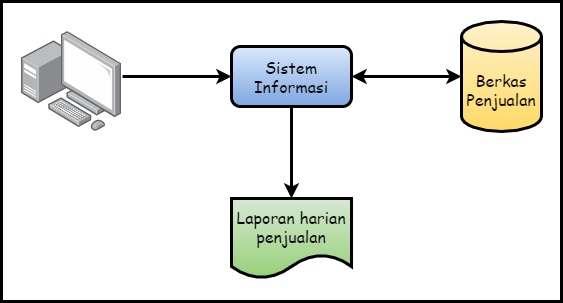
## **2.2 Dasar Teori**

### **2.2.1 Sistem Informasi**

Pada saat ini setiap kegiatan manusia selalu berkaitan dengan sistem informasi. Sistem informasi diterapkan pada berbagai bidang seperti kantor, pasar swalayan, bandara, ataupun di rumah saat kita menggunakan sebuah aplikasi pada sebuah *smartphone*.

Traveloka menyediakan reservasi pesawat terbang yang digunakan untuk melayani pembelian tiket yang bersifat *online*. Dengan menggunakan aplikasi web, orang yang sedang melakukan perjalanan dengan menggunakan pesawat terbang dapat melakukan pembelian tiket secara langsung.

Sebuah sistem informasi tidak harus selalu kompleks. Sebuah sistem tidak harus selalu rumit ada juga sistem yang sederhana. Sebagai contoh dimana sistem digunakan untuk mencatat transaksi penjualan dan hanya melibatkan satu orang saja Seperti pada **Gambar 2.1.**

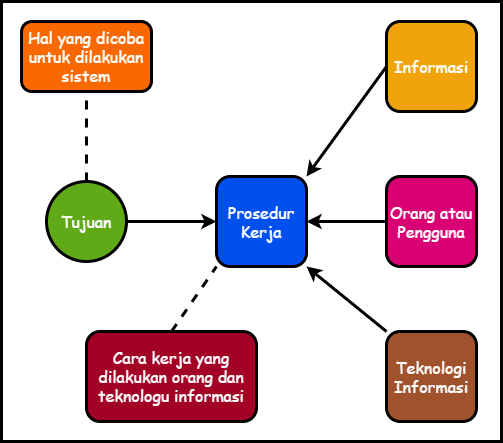


**Gambar 2.1** Sistem informasi sederhana

Sebuah sistem informasi yang kompleks, terdapat banyak pengguna dan jaringan yang tersebar luas, hal ini bertujuan agar pengguna dapat menggunakan sebuah sistem informasi dari berbagai tempat. Kemampuan yang dapat dilakukan oleh sistem informasi diantaranya:

1. Melaksanakan komputasi numerik, bervolume besar, dan berkecepatan tinggi.
2. Mampu menjadi media komunikasi.
3. Dapat menyimpan informasi yang besar dan mudah diakses.
4. Menjadi media informasi yang dapat diakses dari seluruh dunia dengan cepat dan mudah.
5. Meningkatkan efektivitas dan efisiensi dalam pekerjaan.
6. Menyajikan informasi dengan jelas.
7. Membuat sebuah pekerjaan yang dilakukan secara manual menjadi otomatis.
8. Mempercepat dalam hal pengetikan dan penyuntingan.

Menurut Alter (1992) “Sistem informasi adalah kombinasi antara prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi”. Dari penjelasan Alter tersebut diilustrasikan seperti **Gambar 2.2.**



**Gambar 2.2** Sistem informasi

Sistem informasi tidak selalu berhubungan dengan komputer. Pekerjaan yang menggunakan sistem informasi komputer disebut Sistem Informasi Berbasis Komputer (Computer Based Information Systems atau CBIS). Sistem informasi mencangkup beberapa komponen utama yaitu manusia, komputer, prosedur kerja, teknologi informasi, proses, dan bertujuan untuk mencapai tujuan atau *goals*.

### **2.2.2 Basis Data**

Sistem informasi memiliki sebuah media untuk menyimpan dan mengelola sebuah data yang disebut basis data. Basis data dapat mempercepat upaya layanan pelanggan menyediakan informasi yang cepat dan akurat untuk memantu memuat keputusan pemecahan masalah segera erdasarkan informasi yang tersedia. Banyak aplikasi berbasis database termasuk: semua perbankan, pemesanan dan perencanaan penerbangan penerimaan universitas dan proses entri data, pemrosesan pesanan penjualan, pembelian dan pendaftaran data barang di perusahaan komersial, pendaftaran data karyawan dan aktivitas mereka. Termasuk kegiatan penggajian di perusahaan dan lain-lain. Basis data adalah kumpulan data yang saling berelasi. Data itu sendiri adalah fakta tentang sesuatu ataupun manusia. Data digambarkan sebagai nilai (angka, string, karakter, atau simbol). Basis data dimaksudkan untuk mengatur data sedemikian rupa sehingga mudah, akurat, dan cepat untuk diakses. Manfaat yang kita dapatkan dari basis data sendiri beragam, diantaranya adalah:

1. **Kecepatan dan kemudahan**

Dengan menggunakan basis data pengambilan informasi dapat dilakukan

dengan cepat dan mudah.

1. **Kebersamaan pemakai**

Sebuah database dapat diakses oleh berbagai pengguna yang memiliki hak akses dan dari banyak aplikasi. Untuk database yang digunakan oleh banyak pengguna dan organisasi, tidak perlu menyimpan di setiap bagian, tetapi cukup dengan satu database untuk digunakan bersama.

1. **Pemusatan kontrol data**

Karena satu database cukup untuk banyak tujuan pemantauan data juga cukup untuk dilakukan di satu tempat.

1. **Efisiensi ruang penyimpanan**

Dengan penggunaan bersama kita tidak perlu menyediakan ruang penyimpanan di banyak tempat tetapi hanya perlu menyediakan satu untuk menghemat ruang penyimpanan data organisasi.

1. **Keakuratan**

Menjalankan aturan secara ketat terkait tipe data domain, data keunikan, data hubungan, data lainnya., dapat meningkatkan keakuratan pemasukan atau penyimpanan data.

1. **Ketersediaan**

Dengan database kita bisa membackup data, mengkategorikan data apa saja yang masih dibutuhkan dan data apa saja yang perlu dibackup di tempat lain.

1. **Keamanan**

Seagian besar DBMS dilengkapi dengan fungsi manajemen pengguna dan diberi hak akses yang berbeda tergantung pada pengguna dan lokasinya.

1. **Kemudahan dalam pembuatan program aplikasi baru**

Dengan database pemuatan aplikasi dapat memanfaatkan kemampuan DBMS, sehingga pengembangan aplikasi tidak perlu berurusan dengan penyimpanan data, tetapi hanya perlu mengelola antarmuka pengguna.

1. **Pemakain secara langsung**

Basis data mampu menampilkan data langsung menggunakan alat yang disediakan oleh DBMS.

1. **Kebebasan data**

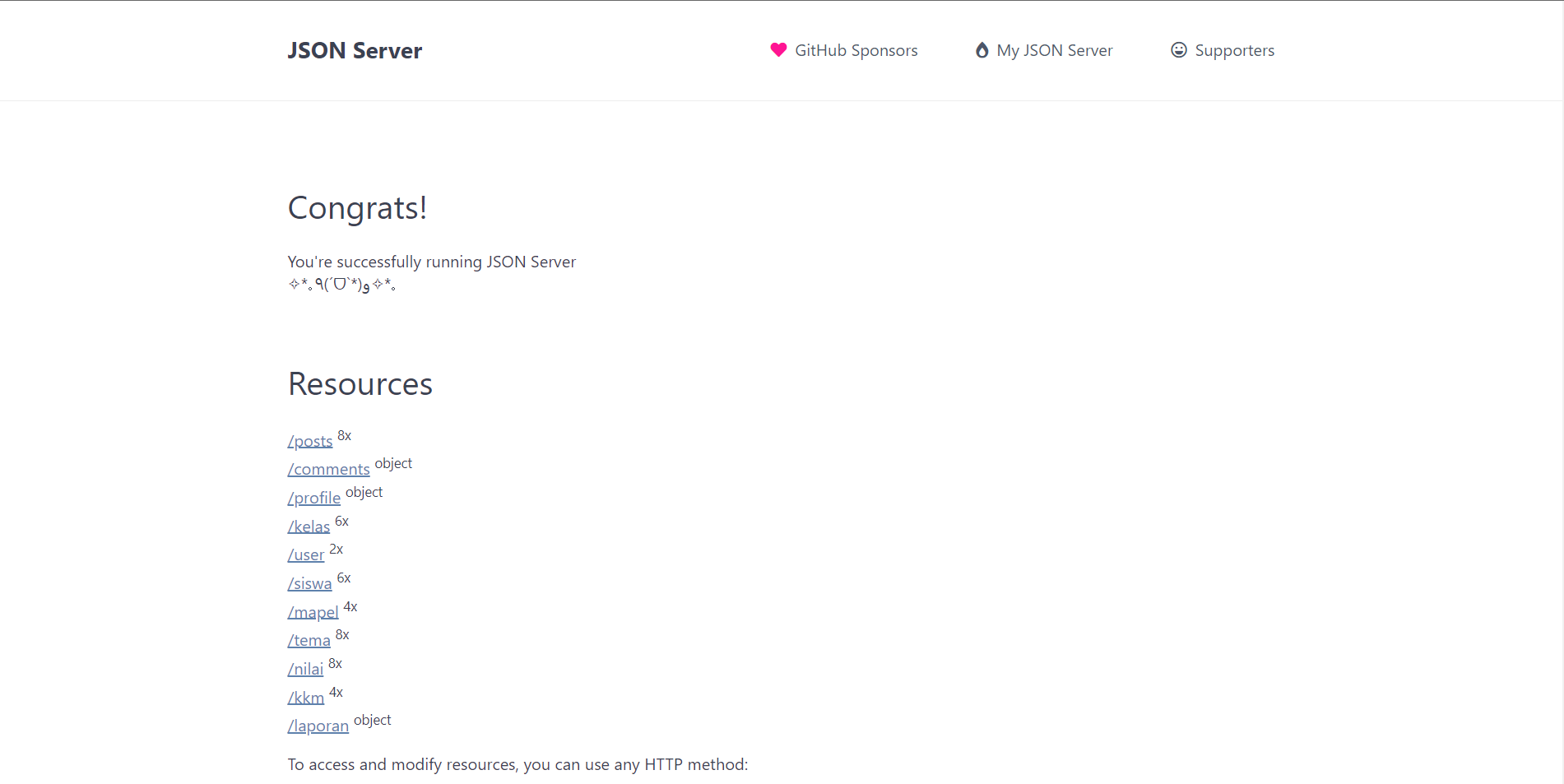
Jika suatu program dihentikan dan ternyata terjadi perubahan isi atau struktur data. Oleh karena itu dengan database modifikasi ini hanya oleh dilakukan pada tingkat DBMS tanpa membongkar program aplikasi.

1. **User view**

Basis data memberikan tampilan yang berbeda untuk setiap pengguna.

### **2.2.3 JSON Server**

JSON Server adalah sebuah tool atau perangkat lunak open-source yang digunakan untuk membuat REST API palsu dengan cepat dan mudah, menggunakan data dalam format JSON. JSON Server dapat digunakan sebagai pengganti server REST API yang sebenarnya, terutama dalam pengembangan aplikasi atau prototyping.



Gambar 1. JSON Server Berjalan di Web Browser

Dalam JSON Server, Anda dapat membuat file JSON sebagai sumber data dan menentukan rute atau endpoint untuk mengakses data tersebut melalui HTTP request, seperti GET, POST, PUT, dan DELETE. JSON Server kemudian akan menangani permintaan HTTP tersebut dan memberikan respons yang sesuai dengan data yang telah diatur dalam file JSON.



Gambar 1. Contoh Data Tersimpan Dalam JSON Server

Keuntungan menggunakan JSON Server adalah kemudahan dan kecepatan dalam membuat REST API palsu, karena tidak memerlukan konfigurasi server yang rumit atau bahkan database terpisah. JSON Server juga menyediakan beberapa fitur seperti pembatasan data, pengurutan, dan pencarian data. JSON Server cocok untuk digunakan dalam pengembangan aplikasi atau prototyping di lingkungan pengembangan web, terutama untuk membuat model data yang realistis.

### **2.2.4 Website**

Webiste atau situs juga dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman(hyperlink). Secara garis besar website bisa digolongkan menjadi tiga bagian, yaitu :

1. Website Statis adalah web yang mempunyai halaman tidak berbah.
2. Website Dinamis merupakan website yang memerlukan update sesering mungkin. Contoh website dinamis adalah web berita atau web portal yang didalamnya terdapat fasilitas berita, polling dan sebagainya.
3. Website Interaktif adalah web yang saat ini memang sedang booming user bisa berinteraksi dan beradu argument. Salah satu contoh website interaktif adalah blog dan forum.

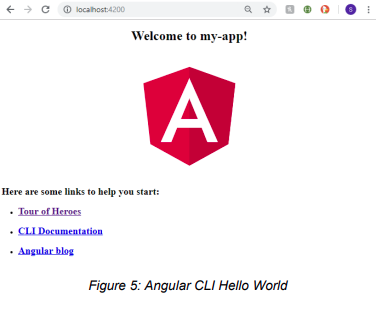
Saat ini terdapat sebuah aplikasi berbasis web, yaitu sebuah aplikasi yang dapat diakses melalui internet dan pada sekarang ini ternyata lebih banyak dan lebih luas pemakaiannya. Banyak dari perusahaan-perusahaan berkembang yang menggunakan Aplikasi Berbasis Web dalam merencanakan sumber daya mereka dan untuk mengelola perusahaan mereka.

Aplikasi Berbasis Web dapat digunakan untuk berbagai macam tujuan yang berbeda. Sebagai contoh, Aplikasi Berbasis Web dapat digunakan untuk membuat invoice dan memberikan cara yang mudah dalam penyimpanan data di database. Aplikasi ini juga dapat dipergunakan untuk mengatur persediaan karena fitur tersebut sangat berguna. Bukan hanya itu Aplikasi Berbasis Web juga dapat bekerja memonitoring dalam sistem hal tampilan. Bahkan jumlah dari Aplikasi Berbasis Web sekarang sudah tak terhitung lagi dan dapat dipesan dan disesuaikan dengan kebutuhan konsumen.

Selain fungsi-fungsi tersebut salah satu keunggulan kompetitif dari Aplikasi Berbasis Web adalah bahwa aplikasi tersebut ringan dan dapat diakses dengan cepat melalui browser dan koneksi internet atau intranet ke server. Ini berarti bahwa pengguna dapat mengakses data atau informasi apapun melalui laptop, smartphone bahkan komputer PC dirumah mereka dengan mudah, tidak seperti aplikasi-aplikasi dekstop dimana pengguna harus menginstal perangkat lunak atau aplikasi yang diperlukan hanya untuk mengakses data/informasi.

### **2.2.5 Framework Angular**

AngularJS adalah framework JavaScript yang dikelola dan dikembangkan oleh Google Inc. dan beberapa kontributor *open source*. Angular awalnya dirilis pada tahun 2009 terutama untuk tujuan memuat sintaks HTML lebih cocok untuk pengembangan aplikasi. Ini termasuk konsep seperti pengikatan data dan template HTML. Dalam bentuknya yang paling sederhana aplikasi AngularJS akan terdiri dari halaman HTML yang disematkan (dengan "variabel") dan ojek JavaScript (disebut pengontrol) dengan properti dan metode. Pengembang akan memanipulasi properti utama dan Angular akan secara otomatis memperbarui DOM (halaman HTML) untuk menampilkan perubahan yang dibuat. Angular 2 dirilis pada tahun 2014 dan telah sepenuhnya ditulis ulang untuk memanfaatkan fitur-fitur baru yang memungkinkan pengembangan komponen. Berikut adalah pendekatan lain untuk pengembangan UI; alih-alih membangun halaman dan berharap perpustakaan (*library*) pihak ketiga tidak bertentangan dengan kode yang dibuat pengembang, pengembang memuat komponen yang akan berfungsi seperti yang diharapkan dan kemudian menggunakan Angular untuk merender komponen bagi pengguna. *Framework Angular* pada saat pertama kali dijalankan akan terdapat tampilan awal aplikasi angular dan terdapat pesan “*Welcome to my-app*”, seperti pada **Gambar 2.3.**



**Gambar 2.3** Angular First Run

Angular 2 adalah penulisan ulang lengkap dari perpustakaan Angular dan tidak kompatibel dengan aplikasi Angular 1. Hal ini menimbulkan kekhawatiran di antara pengembang tetapi tim Angular ingin memanfaatkan banyak fitur baru yang tidak tersedia untuk versi sebelumnya. Angular 2 adalah tentang memuat sebagian besar pengembangan browser baru, dapat bergerak maju dan membangun aplikasi yang lebih baik. Terdapat beberapa fitur web baru yang ada di dalam Angular seperti ECMAScript dan TypeScript. ECMAScript adalah struktur bahasa *scripting* berstandar internasional yang ditetapkan oleh ECMA. JavaScript adalah salah satu implementasi ECMAScript paling populer untuk aplikasi web sisi klien. Ini pertama kali diterbitkan pada tahun 1997 dan telah berkemang selama bertahun-tahun. Versi terbaru (ECMAScript 2018) menambahkan peningkatan sintaks yang signifikan untuk menulis aplikasi skrip yang kompleks. Meskipun tidak semua browser mendukung semua fitur baru, ECMAScript merupakan terobosan terbaru untuk JavaScript. TypeScript adalah superset dari JavaScript yang dikelola dan dikembangkan oleh Microsoft. Itu menambahkan fungsionalitas ke JavaScript terutama tipe data untuk variabel. Pengembang juga menambahkan banyak fungsi dalam ECMAScript (bahasa *scripting* yang didukung oleh sebagian besar *browser* saat ini). Angular sendiri ditulis dalam TypeScript.

Angular adalah kerangka kerja yang terus berkemang dan meningkat. Versi kerangka kerja saat ini adalah 15.0. Secara umum rilis besar baru direncanakan setiap enam bulan dan rilis baru kompatibel dengan rilis besar terakhir. Jika memungkinkan pengembang harus memperbarui ke rilis utama baru segera setelah stabil. Komponen web pertama kali diperkenalkan pada tahun 2011 meskipun komponen merupakan bagian dari proses pengembangan perangkat lunak selama bertahun-tahun sebelumnya. Standar yang diperlukan untuk mengimplementasikan komponen web sedang dikembangkan oleh W3C dan mewakili masa depan pengemangan aplikasi web.

### **2.2.7 Unified Modeling Language (UML)**

Unified Modeling Language (UML) adalah salah satu metode pemodelan visual yang digunakan dalam perancangan dan pembuatan software berbasis objek. UML merupakan standar industri yang digunakan untuk membuat blueprint software yang jelas dan mudah dipahami. Penggunaan UML membantu dalam menggambarkan bisnis proses, kelas-kelas dalam bahasa yang spesifik, serta hubungan dan interaksi antara objek-objek yang ada dalam sebuah sistem. UML terdiri dari berbagai macam diagram, seperti use case diagram, class diagram, sequence diagram, state diagram, dan lain-lain, yang sering digunakan dalam pengembangan sistem. Berikut penjelasan masing-masing diagram:

1. Use case diagram, digunakan untuk menggambarkan interaksi antara aktor (user, sistem lain, atau sistem lainnya) dan sistem dalam sebuah sistem atau aplikasi. Use case diagram memperlihatkan bagaimana aktor terlibat dalam sebuah use case (fitur atau fungsi sistem), serta menunjukkan bagaimana use case tersebut berhubungan dengan use case lainnya.
2. Class diagram, digunakan untuk merepresentasikan struktur dari sistem atau aplikasi dalam bentuk objek atau kelas. Class diagram biasanya digunakan untuk menggambarkan objek-objek atau kelas-kelas dalam sistem, termasuk atribut dan operasi dari setiap objek atau kelas, serta hubungan antar objek atau kelas.
3. Sequence diagram, digunakan untuk merepresentasikan interaksi antar objek dalam sistem dalam bentuk urutan waktu. Sequence diagram biasanya digunakan untuk menggambarkan alur pesan atau panggilan metode antar objek dalam sistem, serta menggambarkan waktu atau urutan dari setiap pesan atau panggilan metode tersebut.
4. State diagram atau diagram keadaan, digunakan untuk menggambarkan keadaan atau status sebuah objek dalam sistem dan transisi antara keadaan tersebut. State diagram biasanya digunakan untuk menggambarkan perilaku dari sebuah objek atau proses dalam sistem yang mengalami perubahan status atau keadaan selama proses berlangsung.

Dengan penggunaan UML, pengembang software dapat membuat desain yang lebih baik dan lebih mudah dimengerti oleh tim pengembang lainnya. Dalam hal ini, UML dapat membantu dalam meningkatkan produktivitas dan efisiensi pengembangan software.

### **2.2.6 Class Diagram**

Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. Class diagram membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, class diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.Class diagram memiliki 3 area pokok (utama) yaitu: nama, atribut dan operasi. Nama berfungsi untuk member identitas pada sebuah class, atribut fungsinya adalah untuk member karakteristik pada data yang dimiliki suatu objek didalam kelas, sedangkan operasi fungsinya adalah memberikan sebuah fungsi ke sebuah objek. Dalam mendefinisikan metode yang ada didalam class diagramharus diperhatikan yang namanya Cohesion dan Coupling, Cohesion adalah ukuran keterkaitan sebuah instruksi disebuah metode, Coupling adalah ukuran keterkaitan antar metode. Didalam class diagram terdapat hubungan antar class secara konseptual, yang disebut relasi antar class.. Simbol dari class diagram dapat dilihat pada **Tabel 2.2**

**Tabel 2.2** Simbol Class Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| 1 |  | Asosiasi/Association | Asosiasi yaitu hubungan statis antar kelas. |
| 2 |  | Agregasi / Agregation | Agregasi  merupakan hubungan antara satu object dengan object lainnya dimana object satu  dengan object lainnya sebenarnya terpisah namun disatukan, sehingga tidak  terjadi kebergantungan (Object lain bisa terbentuk walaupun object  penampungnya belum terbentuk). |
| 3 |  | Antarmuka/interface | Sama dengan konsep interface dalam pemporgraman berorientasi objek |
| 4 |  | Kelas/class | Kelas pada struktur sistem |
| 5 |  | Asosiasi berarah | Kelas antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi berarah biasanya juga disertai dengan multiplicity. |
| 6 |  | Generalisasi | hubungan pewarisan (inheritance) antar unsur dalam  class diagram. Pewarisan memungkinkan suatu kelas mewarisi semua atribut,  operasi, relasi dari kelas yang berada dalam hirarki pewarisannya |
| 7 |  | Kebergantungan/dependency | hubungan antar class di mana sebuah class memiliki ketergantungan pada class lainnya tetapi tidak sebaliknya |

### **2.2.7 Entity Relationship Diagram (ERD)**

Entity Relationship Diagram adalah kumpulan konsep yang menguraikan struktur basis data dan suatu hubungan timbal-balik dan proses pembaruan pada basis data. Tujuan utamanya untuk mengembangkan suatu teknik hubungan tingkat tinggi dengan perancangan basis data. ERD digunakan untuk memodelkan 29 struktur data dan hubungan antar data. Dengan ERD, model dapat diuji dengan mengabaikan proses yang dilakukan. Notasi yang digunakan pada Entity Relationship Diagram dapat dilihat pada **Tabel 2.4.** (Supardi, 2010).

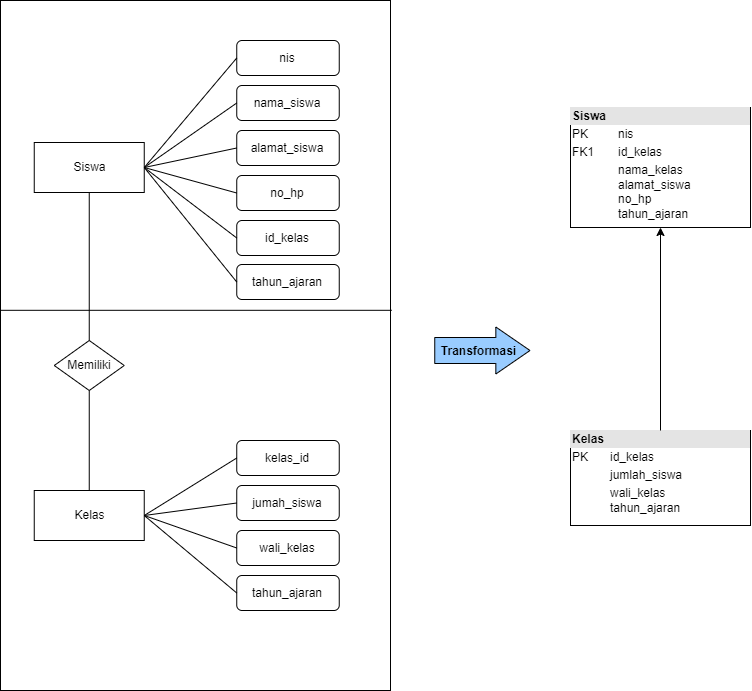
**Tabel 2.4** Notasi Entity Relationship Diagram

|  |  |
| --- | --- |
| **Notasi** | **Keterangan** |
|  | Entitas merupakan suatu objek yang dapat diidentifikasi dalam lingkungan pemakai |
|  | Relasi berfungsi sebagai penunjuk adanya hubungan di antara sejumlah entitas yang berbeda |
|  | Atribut merupakan deskripsi karakter entitas |
|  | Garis berfungsi sebagai penghubung antara relasi dengan entitas, relasi dan antitas dengan atribut |

## **2.2.8 Logical Record Structure (LRS)**

Logical Record Structure (LRS) adalah representasi dari struktur record-record pada tebel-tabel yang terbentuk dari hasil antar himpunan entitas yang digunakan untuk Menentukan kardinalitas, jumlah table dan Foreign Key (FK). Setelah menggambarkan ER-Diagram, selanjutnya kita akan mentransfomasi ER-Diagram tersebut ke Logical Record Structure (LRS). Aturan-aturan yang berlaku untuk mentransformasikan ER-Digram ke LRS, sebagai berikut:

* Setiap Entity akan diubah kebentuk sebuah kotak dengan nama Entity berada diluar kotak dan atribut-atributnya berada didalam kotak.
* Sebuah Relasi terkadang disatukan dalam sebuah kotak bersama Entity, terkadang bisa juga secara terpisah dalam sebuah kotak tersendiri.



**Gambar 2.** Contoh *Logical Record Structure* (LRS)

Aturan-aturan diatas akan sangat dipengaruhi oleh elemen-elemen yang menjadi titik perhatian utama pada langkah transformasi yaitu *Cardinality* atau Kardinalitas.

1. Kardinalitas One to One (1:1)

Pada kardinalitas One to One, salah satu jenis hubungan antar tabel dalam sebuah diagram ERD yang menunjukkan bahwa satu data pada entitas A hanya memiliki satu data yang sesuai pada entitas B, dan begitu juga sebaliknya. Artinya, hubungan antara entitas A dan B bersifat satu-satu atau satu ke satu.

1. Kardinalitas One to Many (1:M)

Pada kardinalitas One to Many sebuah relasi mengindikasikan bahwa satu entitas pada entitas induk (One) dapat berelasi dengan banyak entitas pada entitas anak (Many). Namun, saat menggabungkan relasi dengan entitas, penting untuk mempertimbangkan atribut pada kedua entitas.

1. Kardinalitas Many to Many (M:M)

Pada kardinalitas Many to Many sebuah relasi menunjukkan bahwa banyak entitas pada entitas induk dapat berelasi dengan banyak entitas pada entitas anak. Untuk menangani kardinalitas Many to Many, diperlukan sebuah tabel tambahan (tabel asosiasi) untuk memetakan hubungan antara entitas di kedua sisi relasi.

Dalam desain relasi database, pemahaman tentang kardinalitas sangat penting untuk membangun struktur database yang efisien dan sesuai dengan kebutuhan bisnis. Terdapat tiga kardinalitas utama yang digunakan dalam relasi database, yaitu kardinalitas One to One, One to Many, dan Many to Many. Dengan memahami kardinalitas dalam relasi database, kita dapat merancang struktur database yang tepat dan efisien, serta memudahkan dalam melakukan manipulasi dan analisis data pada database.

### **2.2.11 Metode Rapid Aplication Development (RAD)**

RAD adalah sebuah strategi pengembangan sistem yang menekankan kecepatan pengembangan melalui keterlibatan pengguna yang ekstensif dalam konstruksi, cepat, berulang dan bertambah serangkain prototype / prototipe bekerja sebuah sistem yang pada akhirnya berkembang kedalam sistem final (atau sebuah versi) (Whitten, 2004).

Tujuan utama dari semua metode sistem development adalah memberikan suatu sistem yang dapat memenuhi harapan dari para pemakai, tapi terkadang para pemakai tidak dilibatkan langsung dalam melakukan pengembangan sistem sehingga hal ini menyebabkan sistem informasi yang dibuat jauh dari harapan. Metode RAD mempunyai 3 tahapan utama, yaitu :



**Gambar 2.5** Metode RAD (Kendall, 2003)

Model pengembangan RAD memiliki empat fase, yaitu fase perencanaan syarat-syarat, fase perencangan, fase konstruksi, dan fase pelaksanaan. Berikut adalah penjelasan fase-fase dalam penelitian ini (Kendall, 2003):

1. **Fase perencanaan persyaratan**

Pada tahap ini dilakukan pengidentifikasian tujuan aplikasi atau sistem serta untuk mengidentifikasi syarat-syarat informasi yang ditimbulkan dari tujuan-tujuan tersebut. Pada tahap ini penelitian melakukan analisis kebutuhan.

1. **Fase perancangan**

Pada tahap ini adalah melakukan proses desain dan melakukan perbaikan-[erbaikan apabila masih terdapat ketidak sesuaian desain antara user dan analyst. Untuk tahap ini keaktifan user terlibat sangat menentukan untuk mencapai tujuan, karena user bisa langsung memberikan masukkan apabila terdapat ketidaksesuaian pada desain.

1. **Fase konstruksi**

Setelah desain dari sistem yang akan dibuat sudah disetujui baik itu oleh user dan analyst, maka pada tahap ini programmer mengembangkan desain menjadi satu program. Hal terpenting adalah keterlibatan user sangat diperlukan supaya sistem yang dikembangkan dapat sesuai dengan permintaan user.

1. **Fase pelaksanaan**

Pada tahap ini dilakukan pengujian aplikasi, dengan menggunakan black box testing. Black box testing dilakukan tanpa memperhatikan struktur internal dari aplikasi atau kode yang digunakan dalam pembuatan aplikasi tersebut.

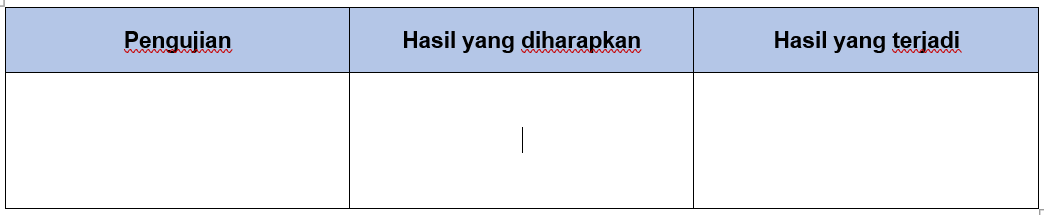
### **2.2.12. Blackbox Testing**

Black box testing adalah teknik pengujian perangkat lunak yang berfokus pada evaluasi fungsionalitas aplikasi perangkat lunak tanpa memiliki pengetahuan tentang struktur internalnya, implementasi kode, atau logika. Teknik ini memperlakukan perangkat lunak sebagai "*blackbox*/kotak hitam," di mana pengujian hanya berinteraksi dengan input dan mengamati output tanpa mengetahui detail internalnya. Dalam black box testing, pengujian dilakukan dengan memeriksa perilaku eksternal perangkat lunak dengan memberikan berbagai input dan memverifikasi output yang dihasilkan sesuai dengan hasil yang diharapkan. Tujuannya adalah memastikan bahwa perangkat lunak berfungsi dengan benar sesuai dengan spesifikasi, kebutuhan, atau harapan pengguna.

Tujuan Black Box Testing adalah untuk menguji fungsi perangkat lunak dan memastikan bahwa pemasukan data dan keluaran telah berjalan sesuai dengan yang diharapkan. Metode ini juga memastikan bahwa informasi yang disimpan dan interaksi dengan komponen eksternal tetap terjaga keutuhannya. Black Box Testing bukanlah solusi alternatif dari White Box Testing, melainkan merupakan pendekatan yang melengkapi pengujian yang dilakukan oleh White Box Testing. Black Box Testing berfokus pada pengujian hal-hal yang tidak terlihat dari segi struktur internal kode atau implementasi, melainkan lebih pada pengujian fungsionalitas perangkat lunak. Dalam Black Box Testing, jenis kesalahan yang diidentifikasi termasuk fungsi yang tidak benar atau tidak ada, kesalahan pada antarmuka, kesalahan pada struktur data dan akses basis data, kesalahan performansi, serta kesalahan pada inisialisasi dan terminasi sistem.

Teknik pengujian black box meliputi pengujian *Functional Testing*, *Equivalence Class Testing*, *Boundary Testing*, *Stress Testing*, *Configuration Testing,* dan lainnya. Teknik-teknik ini membantu memastikan bahwa perangkat lunak diuji secara menyeluruh dan memenuhi standar kualitas yang diinginkan sebelum digunakan oleh pengguna.

Salah satu teknik yang paling banyak digunakan pada blackbox testing yaitu teknik *Equivalence Partitioning*. Teknik ini melibatkan pembagian kelompok input ke dalam kelas ekuivalen yang saling setara, di mana setiap kelas ekuivalen diharapkan menghasilkan hasil yang serupa. Dalam setiap kelas ekuivalen, inputnya dianggap memiliki perilaku yang serupa terhadap perangkat lunak yang diuji. Contohnya, jika sebuah aplikasi membutuhkan input angka, maka teknik *Equivalence Partitioning* akan membagi input tersebut menjadi kelas ekuivalen berdasarkan kriteria tertentu. Jika aplikasi hanya menerima angka positif dan negatif, kelas ekuivalen dapat dibagi menjadi dua: satu untuk angka positif dan satu untuk angka negatif. Dalam pengujian, hanya perlu menguji satu angka positif dan satu angka negatif untuk mewakili masing-masing kelas ekuivalen.



**Gambar 2.** Tabel pengujian pada sebuah aplikasi

Keuntungan menggunakan *Equivalence Partitioning* adalah mengurangi jumlah kasus uji yang perlu diuji tanpa mengurangi cakupan pengujian. Dengan memilih satu contoh dari setiap kelas ekuivalen, kita dapat mencakup semua kemungkinan input dan memastikan bahwa perangkat lunak berperilaku secara konsisten pada setiap kelas.

Teknik *Equivalence Partitioning* membantu meningkatkan efisiensi pengujian dan memastikan bahwa kasus uji yang diuji mencakup berbagai kombinasi input yang mungkin terjadi pada perangkat lunak. Ini membantu mengidentifikasi cacat atau masalah dalam perangkat lunak sejak dini sehingga dapat diperbaiki sebelum dirilis ke pengguna.

**Bab 3**

**Metodologi**

**3.1 Subjek Penelitian**

Topik dari penelitian ini adalah melakukan pengembangan perangkat lunak pada sebuah sistem berbasis web. Pada penelitian ini sistem yang akan dikembangkan adalah Sistem Informasi untuk memenuhi kebutuhan pengelolaan nilai harian SDN 1 Warga Mulya berbasis web. Sistem Informasi pengelolaan nilai harian digunakan sebagai media pengelolaan data nilai harian siswa pada SDN 1 Warga Mulya, dimana guru kelas dan guru mata pelajaran tidak perlu lagi menggunakan mengelola nilai dengan Microsoft excel, Microsoft word, maupun tertulis, tetapi cukup dengan memanfaatkan sistem yang sudah tersedia. Penelitian ini akan menggunakan metode RAD (*Rapid Application Development*) untuk melakukan pengembangan mengikuti tahapan yang ada. Diharapkan hasil penelitian ini akan membuat Sistem Informasi untuk pengelolaan nilai harian SDN 1 Warga Mulya berbasis web yang mampu memenuhi kebutuhan sekolah dalam hal administrasi data nilai siswa.

**3.2 Alat Penelitian**

Penelitian ini menggunakan *hardware* dan *software*, diantaranya:

**3.2.1** **Perangkat Keras (Hardware)**

Penelitian ini menggunakan sebuah perangkat keras (*Hardware*) berupa laptop dengan spesifikasi seperti pada **Tabel 3.1.**

**Tabel 3.1** Spesifikasi Laptop

|  |  |  |
| --- | --- | --- |
| **No.** | **Nama** | **Keterangan** |
| 1. | Processor | Intel® Core™ i7-8750H (2.20GHz – 4.10GHz) |
| 2. | Graphics | Intel® UHD Graphics 630, Nvidia GeForce GTX 1050/1050 Ti 4GB |
| 3. | Memory | 16GB DDR4 |
| 4. | SSD | Samsung SSD 970 EVO Plus 500GB |
| 5. | Monitor | 15 inci 1920 x 1080 *pixel*, 300 nits, 144Hz *refresh-rate* (opsional) |

**3.2.2 Perangkat Lunak (Software)**

Penelitian ini menggunakan sebuah perangkat lunak (*Software*) yang digunakan untuk pengembangan perangkat lunak, seperti pada **Tabel 3.2.**

**Tabel 3.2** Spesifikasi Software Penelitian

|  |  |  |
| --- | --- | --- |
| **No.** | **Nama** | **Keterangan** |
| 1. | Sistem Operasi | Windows 10 Home |
| 2. | Sistem/Aplikasi Objek | Sistem Informasi Inventaris Barang |
| 3. | XAMPP versi 3.3.0 Tahun 2021 | Aplikasi yang berfungsi sebagai server yang indipenden (localhost) yang didalamnya ada beberapa sistem seperti: Apache, MySQL, FileZilla, Mercury dan Tomcat. |
| 4. | Visual Studio Code versi 1.61.2 tahun 2021 | Aplikasi yang digunakan sebagai pengolahan kode (kode editor) |
| 5. | MySQL versi 11.11 Tahun 2013 | Aplikasi yang digunakan sebagai pengelolaan database |
| 6. | PhpMyAdmin versi 5.1.1 tahun 2021 | Software yang terdapat didalam XAMPP, berfungsi sebagai pengelolaan database. |
| 7. | Draw.io (desktop) versi 13.9.9 Tahun 2021 | Aplikasi yang digunakan untuk membuat rancangan UI atau prototype |
| 8. | Git versi 2.34.1 Tahun 2021 | Aplikasi yang digunakan untuk melakukan instalisasi framework atau hal lainnya yang berasal dari GitHub atau GitLab. |

**3.3 Metode Pengembangan Sistem**

Adapun dalam tahap pengembangan sistem RAD (*Rapid Application Development*) ini terdiri dari beberapa aktifitas yang tentunya sesuai dengan tahapan yang telah dijabarkan pada alur proses pengembangan sistem. Tahapan tersebut yaitu:

**3.3.1 Perencanaan Syarat-Syarat**

Menggabungkan laporan hasil metode studi lapangan berupa kebijakan pemakai menjadi spesifikasi yang terstruktur dengan menggunakan pemodelan yang berfungsi untuk mengetahui kebutuhan pemakai, sistem terdahulu dan sistem yang masih manual sehingga data yang ada menjadi tidak teratur. Dari analisis sistem tersebut dapat ditetapkan tujuan perancangan. Tahap yang dilakukan antara lain:

1. Mengoleksi peryataan guru tentang pengelolaan nilai harian siswa pada saat ini.
2. Melakukan analisa terhadap aplikasi atau media lain yang digunakan sebagai pengolahan nilai harian siswa.
3. Mempelajari beberapa referensi baik dari aplikasi terhadulu, buku, jurnal, dokumentasi, dan media lain tentang pengembagan sistem pengolahan nilai.

**3.3.2** **Perancangan Sistem (Workshop Design)**

Pada tahap ini peneliti mendesain yang diusulkan agar dapat berjalan dengan lebih baik dan diharapkan dapat mengatasi masalah-masalah yang ada. Penerapan model yang diinginkan pemakai antara lain:

1. Pada tahap ini alat yang digunakan sama dengan tahap analisis sistem yaitu UML (Unified Modelling Language), alasannya adalah untuk lebih memahami langkah awal membangun sistem secara fisik. UML ini terdiri dari *Use Case Diagram, Activity Diagram,* dan *Class Diagram*
2. Perancangan basis data dilakukan dengan *Class Diagram* yang menggambarkan hubungan antar entity yang ada pada *Use Case Diagram* dan spesifikasi tabel.
3. Perancangan Input-Output, dengan membuat *User Interface* (UI) atau layar tampilan. Setelah rancangan layar tampilan terbentuk maka dilakukan tahap pembangunan sistem.

**3.3.3** **Implementasi Sistem (Coding & Testing)**

Setelah melakukan analisis sistem dan perancangan sistem secara rinci, maka tiba saatnya sistem untuk diimplementasikan. Pada tahap ini terdapat banyak aktivitas yang dilakukan. Aktivitas-aktivitas yang dimaksud berupa:

1. **Pengembangan Sistem (Coding)**

Pada tahap ini hasil desain dimasukkan ke dalam bentuk bahasa pemprograman yang digunakan agar dapat dijalankan dalam sistem atau aplikasi.

1. **Pengujian Sistem (Testing)**

Pada tahap ini dilakukan uji coba terhadap sistem yang baru agar dapat digunakan tanpa menemukan kendala-kendala apapun. Ujicoba dilakukan dengan menggunakan metode black box testing. Dalam pengujian black box testing, pengujian dilakukan dengan memberikan masukan atau input tertentu pada aplikasi dan memeriksa keluaran atau output yang dihasilkan oleh aplikasi. Tujuan dari pengujian ini adalah untuk menguji aplikasi secara keseluruhan, termasuk fitur-fitur yang ada dalam aplikasi dan kehandalan aplikasi dalam mengolah data atau informasi. Selain itu, pengujian black box testing juga dapat membantu untuk menemukan bug atau kesalahan dalam aplikasi dan memastikan bahwa aplikasi dapat berjalan dengan baik di berbagai platform dan lingkungan yang berbeda.

**3.4 Tahap Penelitian**

Tahapan penelitian dalam sebuah pengembangan perangkat lunak sangatlah penting, untuk menunjang keberhasilan pengembangan. Sebuah tahapan penelitian tidak selalu sama, tergantung dari apa yang hendak diteliti. Pada pengembangan perangkat lunak tahapan penelitian dapat dilihat pada **Gambar 3.1.**



**Gambar 3.1** Tahap Penelitian

Setiap tahapan pada penelitian memiliki perannya masing-masing diantaranya:

1. Studi literatur merupakan proses analisa perbandingan dari penelitian sebelumnya, dengan cara membaca dan mempelajari buku-buku yang berhubungan dengan analisa dan perancangan sistem dengan menggunakan metode RAD.
2. Perumusan Masalah merupakan masalah yang terjadi pada objek penelitian, dimana perumusan masalah yang didapat berasal dari hasil analisis penelitian pada waktu studi literatur dan data-data yang diambil dari hasil mengoleksi pernyataan mengenai pengolahan nilai harian siswa oleh guru.
3. Tujuan Peneltiian merupakan hasil yang berbentuk media tertentu untuk menangani sebuah permasalahan. Dalam penelitian ini hasil tersebut berupa aplikasi pengolahan nilai harian siswa berbasis web.
4. Tahapan RAD merupakan tahapan dimana dimulainya perancangan syarat-syarat, perancangan sistem seperti pembuatan UML, prototipe dan database, pembuatan sistem (*coding*), dan tahap pengujian (*testing*).
5. Penilisan Laporan merupakan tahapan dokumentasi penelitian dari awal hingga akhir penelitian.

# Bab 4

# Hasil dan Pembahasan

## **4.1 Perancangan Aplikasi**

### **4.1.1 Perancangan Kebutuhan Sistem**

Analisis kebutuhan alat yang diperlukan dalam melakukan pengembangan Sistem Informasi Pengolahan Nilai Harian SDN 1 Warga Mulya Berbasis Web, sebagai berikut :

1. **Hardware (Perangkat Keras)**

Adapun hardware (perangkat keras) yang digunakan pada perancangan aplikasi kali ini adalah sebuah computer atau laptop dengan spesifikasi sebagai berikut :

* Computer atau laptop
* RAM 4 GB
* SSD 120 GB
* HDD 500 GB
* Layar 14 Inci
* Intel Core i3-3217U
* AMD Radeon 8530M + Intel HD 4000
* Wireless 802.11 b/g/n, Gigabit LAN
* Sistem Operasi Windows 10

1. **Software (Perangkat Lunak)**

Untuk software (perangkat lunak) yang digunakan pada perancangan aplikasi ini adalah :

* Windows 10 sebagai sistem operasi
* Draw.io untuk membuat membuat diagram UML
* XAMPP sebagai web server dan database server
* Visual Studio Code (VSC) untuk menuliskan kode program yang digunakan pada aplikasi
* Browser Microsoft Edge untuk menjalankan aplikasi

1. **Brainware (Pengguna atau User)**

Penelitian ini membutuhkan pengguna untuk mengoperasikan aplikasi, adapaun untuk penggunanya adalah sebagai berikut :

* Admin
* Guru

### **4.1.2 Analisis Kebutuhan Fungsional**

Kebutuhan fungsional yang disediakan pada aplikasi ini yaitu :

**a. Admin**

* **Input Data Guru**

Pada poin ini, admin dapat memasukkan data guru ke dalam sistem. Data guru yang dapat dimasukkan meliputi NIP, nama, username, password, alamat, jabatan, no. hp, dan status yang relevan dengan profesi guru.

* **Input Data Siswa**

Pada poin ini, memungkinkan admin untuk memasukkan data siswa ke dalam sistem. Data yang dapat dimasukkan meliputi NIS, nama, alamat, no. hp, kelas, dan tahun ajaran yang relevan dengan profil siswa.

* **Input Data Kelas**

Pada poin ini, admin dapat memasukkan data kelas ke dalam sistem. Data yang dapat dimasukkan meliputi kelas, jumlah siswa, wali kelas, dan tahun ajaran yang relevan dengan profil kelas.

* **Input Data Mata Pelajaran**

Pada poin ini, memungkinkan admin untuk memasukkan data mata pelajaran yang diajarkan dalam suatu kelas. Data yang dapat dimasukkan meliputi nama mata pelajaran dan informasi lain yang relevan dengan profil mata pelajaran.

* **Input Data Tema**

Pada fitur ini, pengguna dapat memasukkan data tema pembelajaran ke dalam sistem. Data yang dapat dimasukkan meliputi nama mata pelajaran, tema, sub tema, dan informasi lain yang relevan dengan profil tema pembelajaran

**b. Guru**

* **Input data nilai**

Pada poin ini, guru dapat memasukkan nilai harian siswa ke dalam sistem. Guru harus mengisi beberapa kolom seperti NIS (Nomor Induk Siswa), nama siswa, kelas, mata pelajaran, tema, subtema, dan nilai yang diperoleh oleh siswa. Setelah selesai mengisi, guru dapat menekan tombol "simpan" untuk menyimpan data nilai harian siswa ke dalam database.

* **Input nilai KKM**

Pada poin ini, pengguna dapat memasukkan nilai KKM (Kriteria Ketuntasan Minimal) ke dalam sistem. Nilai KKM digunakan sebagai acuan dalam menentukan apakah siswa telah mencapai standar yang telah ditetapkan atau belum. Guru dapat memilih mata pelajaran tertentu dan mengisi nilai KKM pada kolom yang tersedia. Setelah selesai, guru dapat menekan tombol "simpan" untuk menyimpan data nilai KKM ke dalam database.

* **Export laporan data nilai**

Pada poin ini, guru dapat mengekspor data laporan nilai harian siswa ke dalam format tertentu seperti file Excel atau PDF. Guru dapat memilih format data tertentu untuk mengekspor data nilai harian siswa. Setelah selesai memilih, guru dapat menekan tombol "cetak" untuk menghasilkan file yang berisi data laporan nilai harian siswa.

* **Melihat data guru**

Pada poin ini, guru dapat melihat data dirinya secara lengkap, yang mana telah diinputkan oleh admin sebelumnya.

* **Melihat data siswa**

Pada poin ini, guru dapat melihat daftar siswa yang terdaftar dalam sistem. Guru dapat melihat informasi seperti NIS, nama lengkap, kelas, no. hp, alamat, dan tahun ajaran yang telah dimasukkan ke dalam sistem. Guru juga dapat mencari siswa berdasarkan NIS atau nama lengkap untuk mempermudah pencarian.

### **4.1.3 Analisis Kebutuhan Non Fungsional**

Kebutuhan non fungsional tidak tidak ada hubungan langsung untuk aplikasi yang ada. Kebutuhan ini lebih mengarah pada aksesibilitas aplikasi, kebutuhannya yaitu :

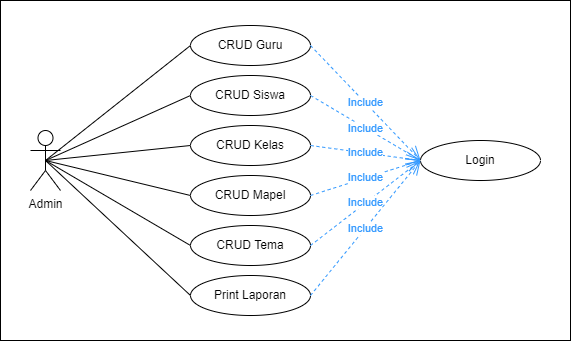
* 1. **Admin**
* Proses input data lebih efisien dan maksimal jika proses dilakukan di laptop atau computer yang terhubung ke internet.
* Aplikasi ini berjalan di web browser untuk halaman admin.
* Harus login terlebih dahulu jika ingin menggunakan aplikasi dan logout jika sudah selesai.
* Apabila username dan password yang dimasukkan salah maka tidak dapat masuk ke dalam aplikasi.
  1. **Guru**
* Proses input data lebih efisien dan maksimal jika proses dilakukan di laptop atau computer yang terhubung ke internet.
* Aplikasi ini berjalan di web browser, termasuk untuk tampilan guru
* Mendapatkan username dan password yang telah dibuatkan oleh admin
* Harus login terlebih dahulu jika ingin menggunakan aplikasi dan logout jika sudah selesai.
* Apabila username dan password yang dimasukkan salah maka tidak dapat masuk ke dalam aplikasi.

## **4.2 Desain Sistem**

Peneliti pada perancangan sistem kali ini ada beberapa rancangan desain, yaitu rancangan pengembangan sistem menggunakan model *Unified Modeling Lenguage* (UML) yang terdiri dari *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*. Sedangkan desain database menggunakan model *Entity Relationship Diagram (ERD)*, *Logical Relationship Diagram (LRS)* dan *Sequence Diagram* untuk pengolahan nilai.

## **4.3 Use Case Diagram**

*Use case diagram* merupakan sebuah diagram yang akan digunakan untuk menampilkan fitur dari aplikasi yang dibuat. *Use Case Diagram* mendeskripsikan sebuah interaksi antara satu atau lebih fungsi di dalam Sistem Informasi Pengolahan Nilai Harian SDN 1 Warga Mulya Berbasis Web.

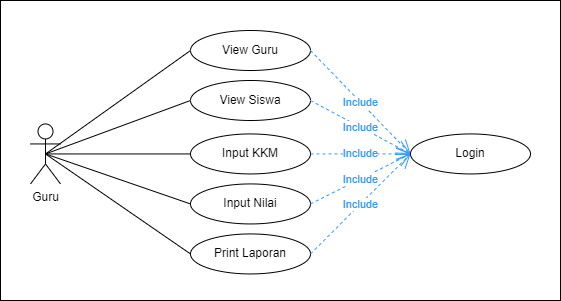


**Gambar 4.** *Use Case Diagram* Admin Pengolahan Nilai Harian Siswa

Gambar 4. menunjukkan beberapa hal yang dapat dilakukan oleh seorang admin, diantaranya dapat mengelola data guru, siswa, kelas, dan mapel (Mata Pelajaran). Kemudian admin juga dapat mencetak atau print laporan.

**Tabel 4.** Definisi *Use Case Diagram* Admin

|  |  |  |
| --- | --- | --- |
| **No** | **Use Case** | **Deskripsi** |
| 1 | Login | Proses masuk ke dalam aplikasi |
| 2 | Admin | Orang yang memiliki hak akses untuk melakukan pengelolaan data guru, siswa, kelas, maple, dan dapat mencetak laporan. |
| 3 | CRUD Guru | Proses pengelolaan data guru mulai dari membuat (create), melihat (read), mengedit (update), dan menghapus (delete). |
| 4 | CRUD Siswa | Proses pengelolaan data siswa mulai dari membuat (create), melihat (read), mengedit (update), dan menghapus (delete). |
| 5 | CRUD Kelas | Proses pengelolaan data kelas mulai dari membuat (create), melihat (read), mengedit (update), dan menghapus (delete). |
| 6 | CRUD Mapel | Proses pengelolaan data mapel mulai dari membuat (create), melihat (read), mengedit (update), dan menghapus (delete). |
| 7 | CRUD Tema | Proses pengelolaan data tema dari setiap mata pelajaran (mapel), dimana aktivitas yang dilakukan mulai dari membuat (create), melihat (read), mengedit (update), dan menghapus (delete). |
| 8 | Print laporan | Proses mencetak data yang dikelola oleh admin seperti cetak data guru, siswa, kelas atau mapel. |



**Gambar 4.** *Use Case Diagram* Guru Pengolahan Nilai Harian Siswa

Gambar 4. menunjukkan beberapa hal yang dapat dilakukan oleh guru dalam sistem, tidak hanya dapat mengelola data nilai saja, guru juga dapat melihat data guru, siswa, kelas, mata pelajaran, dan dapat mencetak atau print laporan.

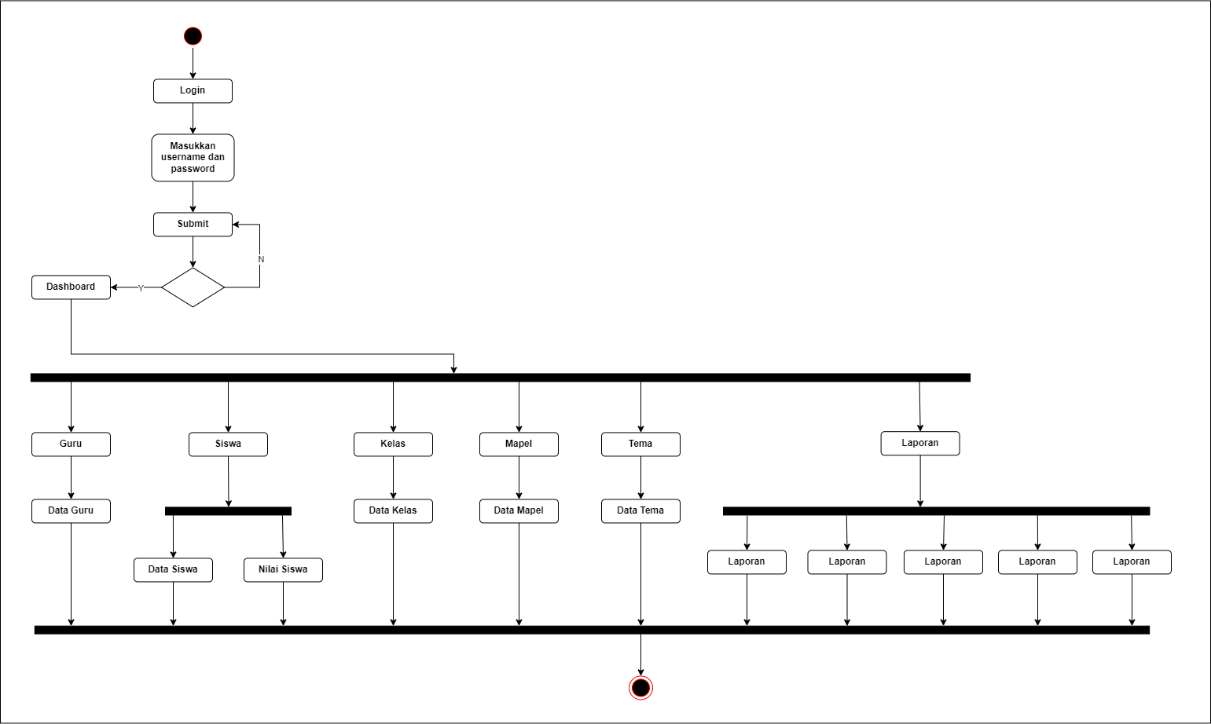
**Tabel 4.** Definisi *Use Case Diagram* Guru

|  |  |  |
| --- | --- | --- |
| **No** | **Use Case** | **Deskripsi** |
| 1 | Login | Proses masuk ke dalam aplikasi |
| 2 | Guru | Orang yang memiliki akses untuk melihat data guru dan data siswa. Dapat mengelola data kkm, nilai, dan dapat mencetak laporan. |
| 3 | View guru | Proses melihat data guru |
| 4 | View siswa | Proses melihat data siswa |
| 5 | Input KKM | Proses memasukkan nilai KKM mata pelajaran |
| 6 | Input nilai | Proses memasukkan nilai mata pelajaran |
| 7 | Print laporan | Proses mencetak data yang dikelola seperti cetak data nilai, data KKM, dan laporan |

## **4.4 Activity Diagram**

### **4.4.1 Activity Diagram Admin**

Sistem informasi pengolahan nilai harian adalah sebuah aplikasi atau sistem yang digunakan untuk mengumpulkan, menyimpan, mengolah, dan menganalisis data nilai harian siswa. Activity diagram yang pertama adalah Admin, dapat dilihat pada Gambar 4.

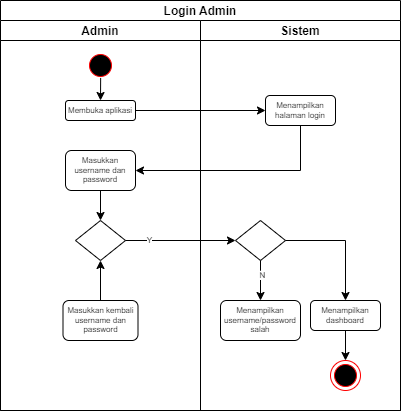


**Gambar 4.** *Diagram Activity* Admin Keseluruhan

Pada Gambar 4. merupakan gambaran aktifitas admin pada sistem informasi pengolahan nilai yang akan dibangun. Pada gambar 4. Terlihat admin mengelola beberapa bagian penting dalam system tersebut, seperti mengelola data guru, data siswa, data kelas, data mata pelajaran, data tema, dan data laporan. Berikut penjelasan aktifitas admin pada setiap bagiannya:

1. Aktifitas login atau masuk ke dalam aplikasi oleh admin

Pertama, admin harus login ke dalam sistem dengan menggunakan username dan password yang sudah terdaftar.

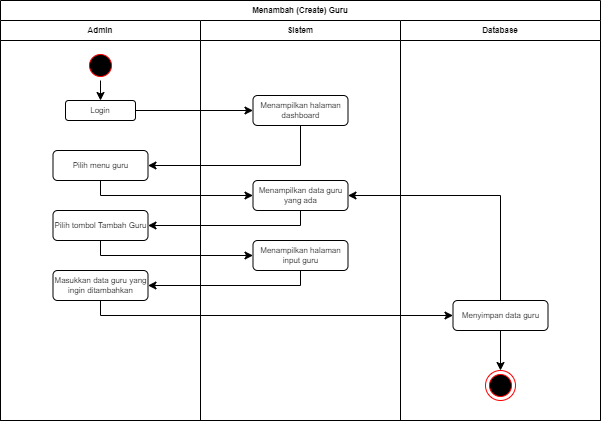


**Gambar 4.** *Activity Diagram* Login Admin

Jika dilihat dari diagram pada gambar 4., admin memasukkan username dan password yang sudah terdaftar, dimana jika data valid maka akan langsung diteruskan oleh system ke dalam halaman dashboard, tetapi jika tidak maka tetap berada di halaman login dan pemberitahuan jika username dan password yang dimasukkan salah.

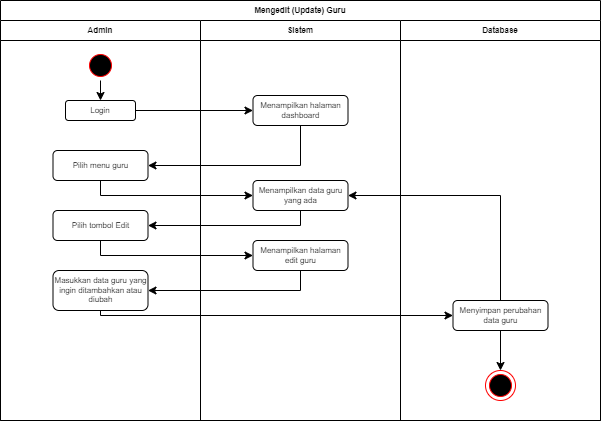
1. Kelola data guru oleh admin

Admin dapat mengelola data guru yang sudah bisa melakukan proses penginputan nilai harian siswa. Mulai dari melihat, menambah, mengedit, dan menghapus data guru.



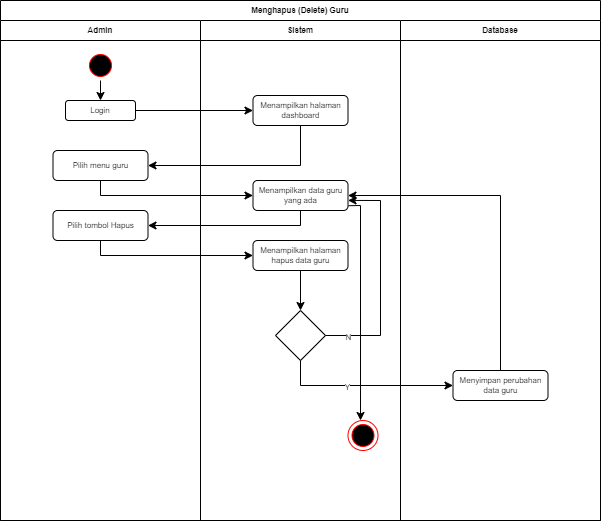
**Gambar 4.** *Activity Diagram* Menambah Data Guru

Pada gambar 4. Menjelaskan proses admin untuk menambah data guru. Hal pertama yang dilakukan oleh admin adalah login ke dalam sistem, kemudian masuk ke dalam menu guru dan tambah data guru. Masukkan data guru yang valid dan secara otomatis akan tersimpan ke dalam database.



**Gambar 4.** *Activity Diagram* Mengedit Data Guru

Pada gambar 4. Menjelaskan proses admin untuk mengedit data guru. Hal pertama yang dilakukan oleh admin adalah login ke dalam sistem, kemudian masuk ke dalam menu guru dan edit data guru. Masukkan data guru yang ingin dirubah dan secara otomatis akan tersimpan ke dalam database.

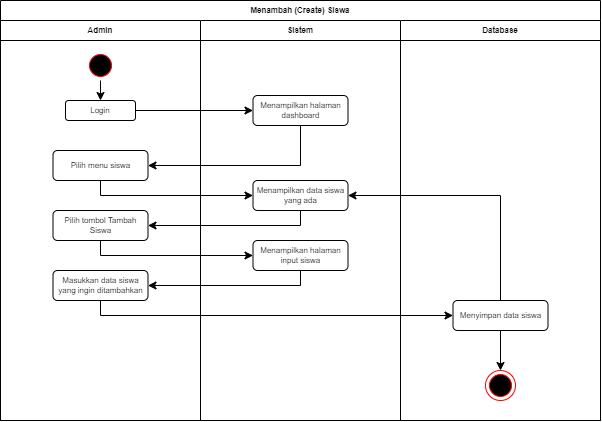


**Gambar 4.** *Activity Diagram* Menghapus Data Guru

Pada gambar 4. Menjelaskan proses admin untuk menghapus data guru. Hal pertama yang dilakukan oleh admin adalah login ke dalam sistem, kemudian masuk ke dalam menu guru dan hapus data guru. Pastikan data guru yang ingin dihapus sudah cek kembali, agar tidak terjadi kesalahan yang tidak diinginkan dan secara otomatis system akan menghapus data guru di dalam database.

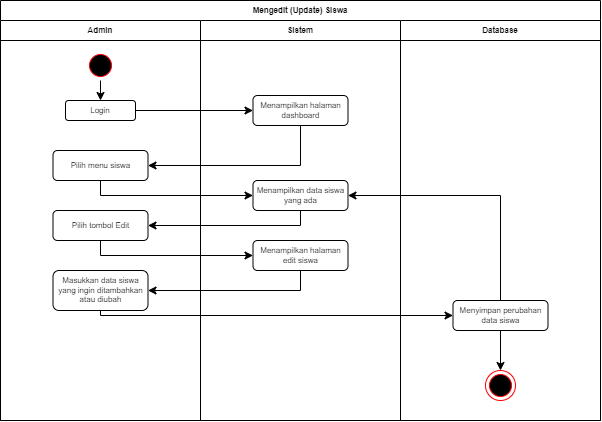
1. Kelola data siswa oleh admin

Admin dapat mengelola data siswa yang sudah terdata sebelumnya ke dalam system. Mulai dari melihat, menambah, mengedit, dan menghapus data siswa.



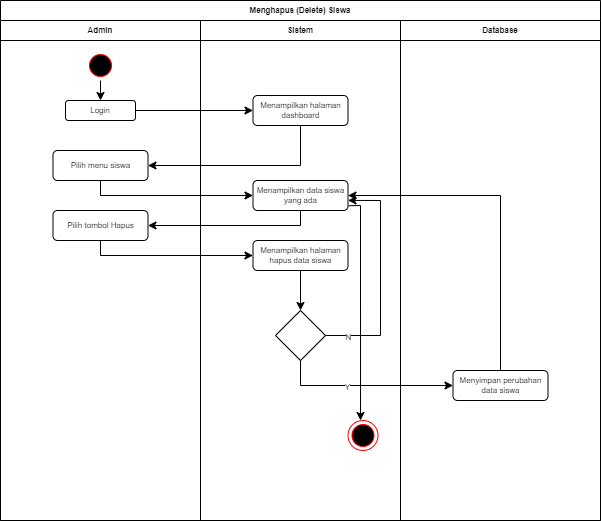
**Gambar 4.** *Activity Diagram* Menambah Data Siswa

Pada gambar 4. Menjelaskan proses admin untuk menambah data siswa. Hal pertama yang dilakukan oleh admin adalah login ke dalam sistem, kemudian masuk ke dalam menu siswa dan tambah data siswa. Masukkan data guru yang valid dan secara otomatis akan tersimpan ke dalam database.



**Gambar 4.** *Activity Diagram* Mengedit Data Siswa

Pada gambar 4. Menjelaskan proses admin untuk mengedit data siswa. Hal pertama yang dilakukan oleh admin adalah login ke dalam sistem, kemudian masuk ke dalam menu siswa dan edit data siswa. Masukkan data siswa yang ingin dirubah dan secara otomatis akan tersimpan ke dalam database.

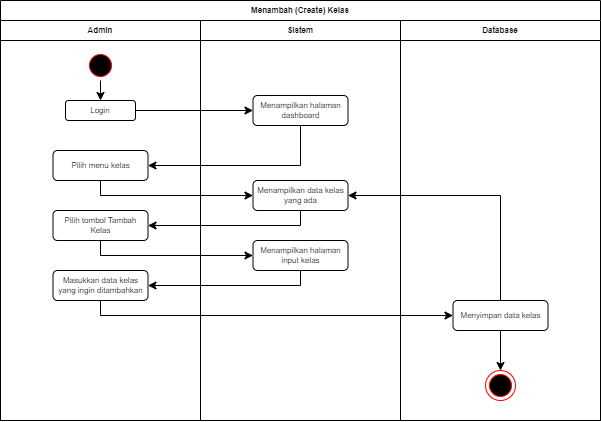


**Gambar 4.** *Activity Diagram* Menghapus Data Siswa

Pada gambar 4. Menjelaskan proses admin untuk menghapus data siswa. Hal pertama yang dilakukan oleh admin adalah login ke dalam sistem, kemudian masuk ke dalam menu siswa dan hapus data siswa. Pastikan data siswa yang ingin dihapus sudah cek kembali, agar tidak terjadi kesalahan yang tidak diinginkan dan secara otomatis system akan menghapus data siswa di dalam database.

1. Kelola data kelas oleh admin

Admin dapat mengelola data kelas yang sudah terdata sebelumnya ke dalam system. Mulai dari melihat, menambah, mengedit, dan menghapus data kelas.



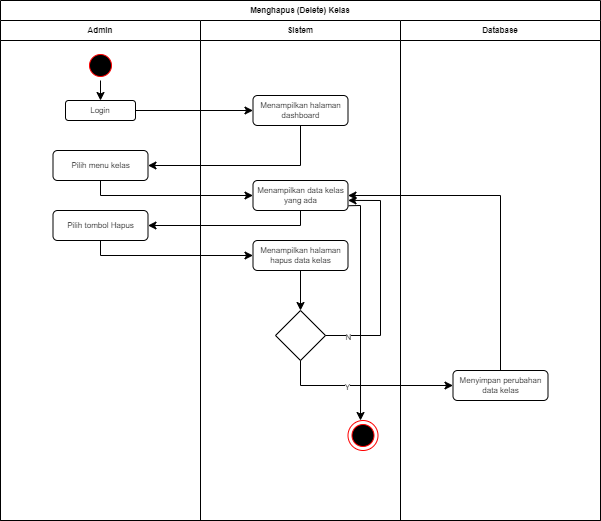
**Gambar 4.** *Activity Diagram* Menambah Data Kelas

Pada gambar 4. Menjelaskan proses admin untuk menambah data kelss. Hal pertama yang dilakukan oleh admin adalah login ke dalam sistem, kemudian masuk ke dalam menu kelas dan tambah data kelas. Masukkan data kelas yang valid dan secara otomatis akan tersimpan ke dalam database.



**Gambar 4.** *Activity Diagram* Mengedit Data Kelas

Pada gambar 4. Menjelaskan proses admin untuk mengedit data kelas. Hal pertama yang dilakukan oleh admin adalah login ke dalam sistem, kemudian masuk ke dalam menu kelas dan edit data kelas. Masukkan data kelas yang ingin dirubah dan secara otomatis akan tersimpan ke dalam database.

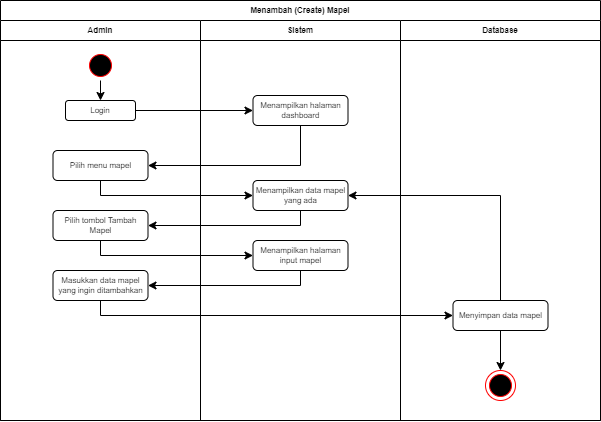


**Gambar 4.** *Activity Diagram* Menghapus Data Kelas

Pada gambar 4. Menjelaskan proses admin untuk menghapus data kelas. Hal pertama yang dilakukan oleh admin adalah login ke dalam sistem, kemudian masuk ke dalam menu kelas dan hapus data kelas. Pastikan data kelas yang ingin dihapus sudah cek kembali, agar tidak terjadi kesalahan yang tidak diinginkan dan secara otomatis system akan menghapus data kelas di dalam database.

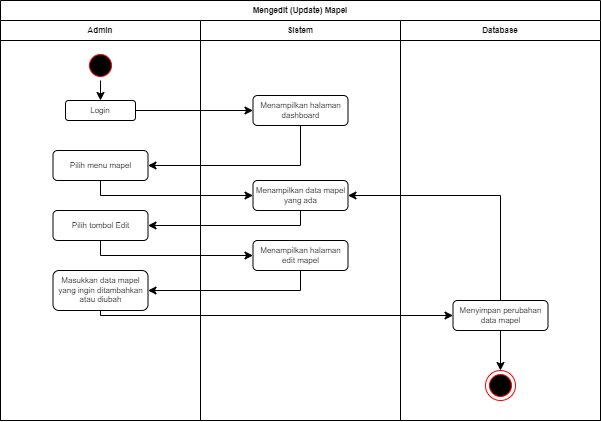
1. Kelola data mapel (mata pelajaran) oleh admin

Admin dapat mengelola data mapel yang sudah terdata sebelumnya ke dalam system. Mulai dari melihat, menambah, mengedit, dan menghapus data mapel.



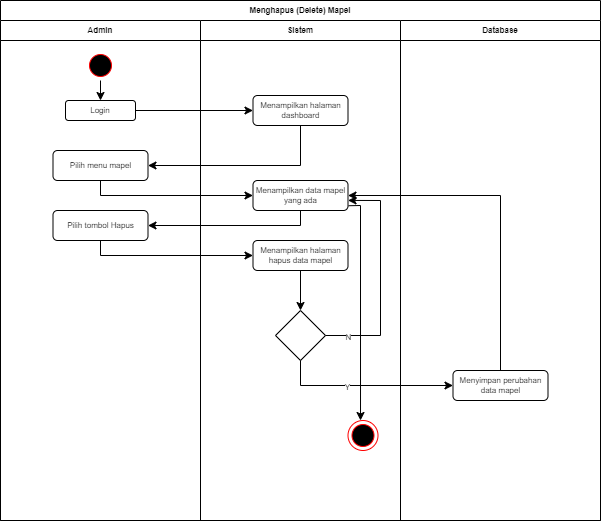
**Gambar 4.** *Activity Diagram* Menambah Mapel

Pada gambar 4. Menjelaskan proses admin untuk menambah data mata pelajaran. Hal pertama yang dilakukan oleh admin adalah login ke dalam sistem, kemudian masuk ke dalam menu mata pelajaran dan tambah data mata pelajaran. Masukkan data mata pelajaranyang valid dan secara otomatis akan tersimpan ke dalam database.



**Gambar 4.** *Activity Diagram* Mengedit Data Mapel

Pada gambar 4. Menjelaskan proses admin untuk mengedit data mata pelajaran. Hal pertama yang dilakukan oleh admin adalah login ke dalam sistem, kemudian masuk ke dalam menu mata pelajaran dan edit data mata pelajaran. Masukkan data mata pelajaran yang ingin dirubah dan secara otomatis akan tersimpan ke dalam database.

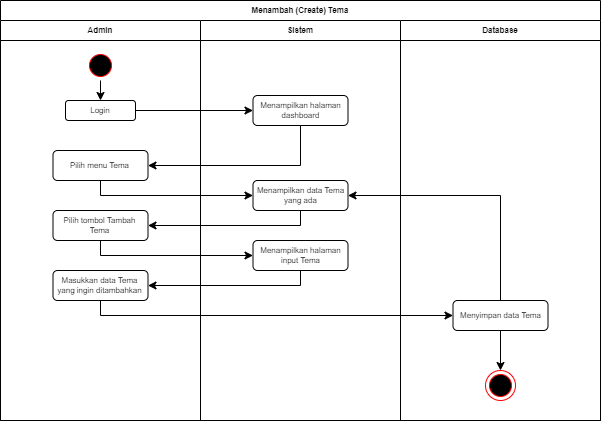


**Gambar 4.** *Acitivity Diagram* Menghapus Data Mapel

Pada gambar 4. Menjelaskan proses admin untuk menghapus data mata pelajaran. Hal pertama yang dilakukan oleh admin adalah login ke dalam sistem, kemudian masuk ke dalam menu mata pelajaran dan hapus data mata pelajaran. Pastikan data mata pelajaran yang ingin dihapus sudah cek kembali, agar tidak terjadi kesalahan yang tidak diinginkan dan secara otomatis system akan menghapus data mata pelajaran di dalam database.

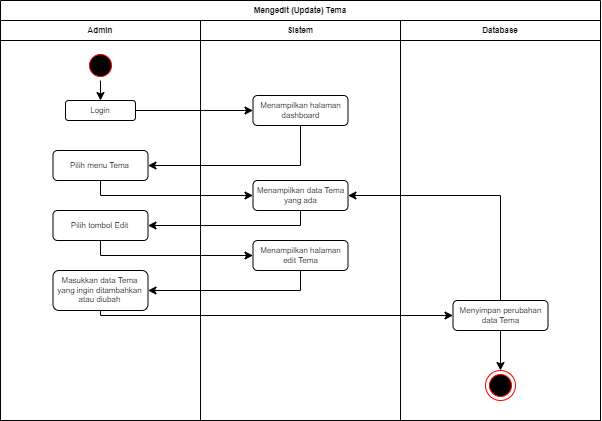
1. Kelola data Tema oleh admin

Admin dapat mengelola data Tema yang sudah terdata sebelumnya ke dalam system. Mulai dari melihat, menambah, mengedit, dan menghapus data Tema.



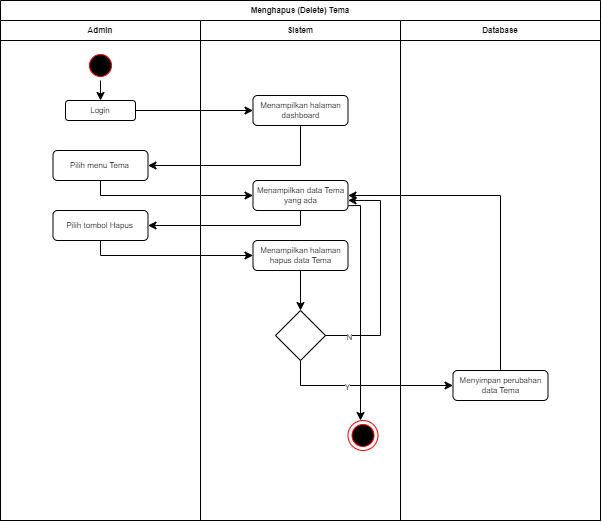
**Gambar 4.** *Acitivity Diagram* Menambah Data Tema

Pada gambar 4. Menjelaskan proses admin untuk menambah data tema. Hal pertama yang dilakukan oleh admin adalah login ke dalam sistem, kemudian masuk ke dalam menu tema dan tambah data tema. Masukkan data tema yang valid dan secara otomatis akan tersimpan ke dalam database.



**Gambar 4.** *Acitivity Diagram* Mengedit Data Tema

Pada gambar 4. Menjelaskan proses admin untuk mengedit data tema. Hal pertama yang dilakukan oleh admin adalah login ke dalam sistem, kemudian masuk ke dalam menu tema dan edit data tema. Masukkan data tema yang ingin dirubah dan secara otomatis akan tersimpan ke dalam database.

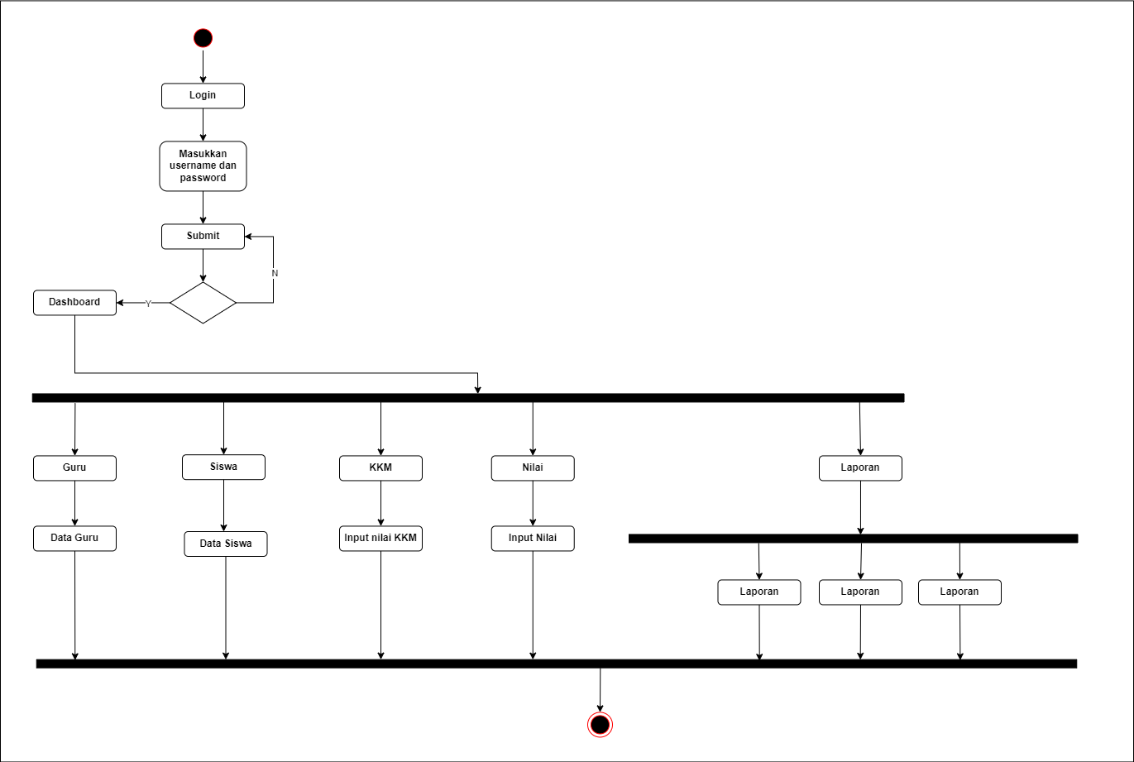


**Gambar 4.** *Acitivity Diagram* Menghapus Data Tema

Pada gambar 4. Menjelaskan proses admin untuk menghapus data tema. Hal pertama yang dilakukan oleh admin adalah login ke dalam sistem, kemudian masuk ke dalam menu tema dan hapus data tema. Pastikan data tema yang ingin dihapus sudah cek kembali, agar tidak terjadi kesalahan yang tidak diinginkan dan secara otomatis system akan menghapus data tema di dalam database.

### **4.4.2 Activity Diagram Guru**

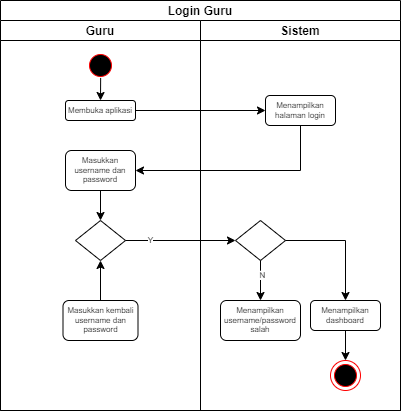
Acrivity diagram guru berisikan aktifitas didalam system informasi pengolahan nilai harian siswa, dimana pada system tersebut terdapat 5 menu utama yaitu dashboard, guru, siswa, kkm, nilai, dan laporan. Untuk gambaran aktifitas guru dapat dilihat pada gambar 4.



**Gambar 4.** *Diagram Activity* Guru

Pada gambar 4. bagian dashboard, guru, dan siswa, guru hanya dapat melihat data yang disediakan saja. Pada bagian kkm dan nilai, guru dapat mengelola data secara keseluruhan seperti menambah data, edit, dan hapus. Pada bagian laporan, guru dapat mencetak data menjadi dua tipe data yaitu data excel dan data pdf.

1. Aktifitas login atau masuk ke dalam aplikasi oleh Guru.

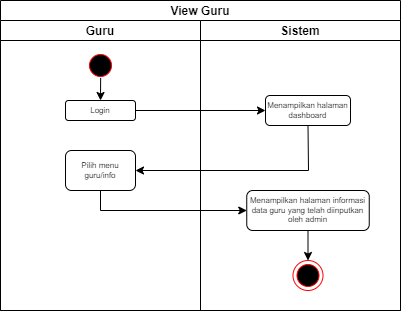


**Gambar 4.** *Activity Diagram* Login Guru

Jika dilihat dari diagram pada gambar 4., guru memasukkan username dan password yang sudah terdaftar, dimana jika data valid maka akan langsung diteruskan oleh system ke dalam halaman dashboard, tetapi jika tidak maka tetap berada di halaman login dan pemberitahuan jika username dan password yang dimasukkan salah.

1. Melihat data diri Guru

Guru dapat melihat data dirinya di dalam system yang telah diinputkan oleh admin sebelumnya.

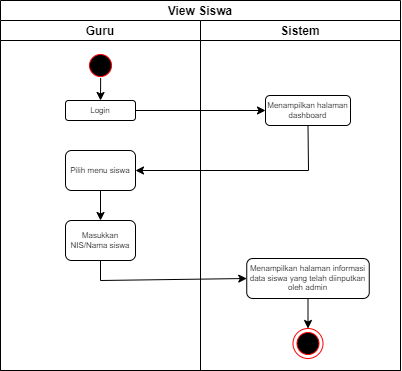


**Gambar 4.** *Activity Diagram* *View* Data Guru

Pada gambar 4. Menjelaskan proses guru untuk melihat data guru. Hal pertama yang dilakukan oleh guru adalah login ke dalam sistem, kemudian masuk ke dalam menu guru. Maka pada halaman tersebut akan ditampilkan data yang sesuai dengan guru yang login saat ini dan data tersebut diambil dari database sistem.

1. Melihat data siswa oleh guru

Guru dapat melihat data siswa yang terdaftar di sekolah pada menu siswa dan guru juga dapat melakukan pencarian data siswa, sesuai dengan kebutuhan.

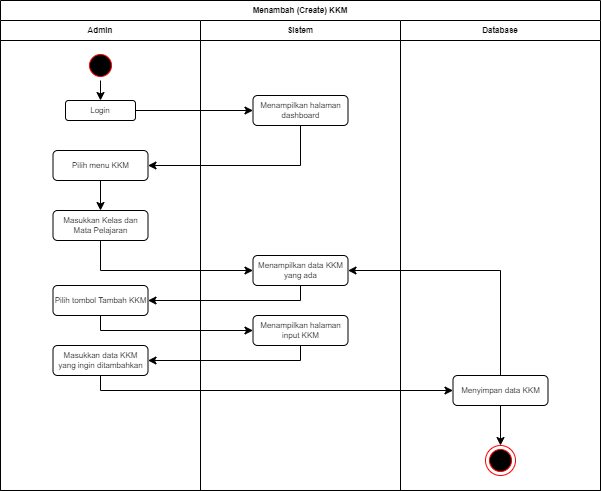


**Gambar 4.** *Activity Diagram* *View* Data Siswa

Pada gambar 4. Menjelaskan proses guru untuk melihat data siswa. Hal pertama yang dilakukan oleh guru adalah login ke dalam sistem, kemudian masuk ke dalam menu siswa. Maka pada halaman tersebut akan ditampilkan data siswa yang mana data tersebut diambil dari database sistem.

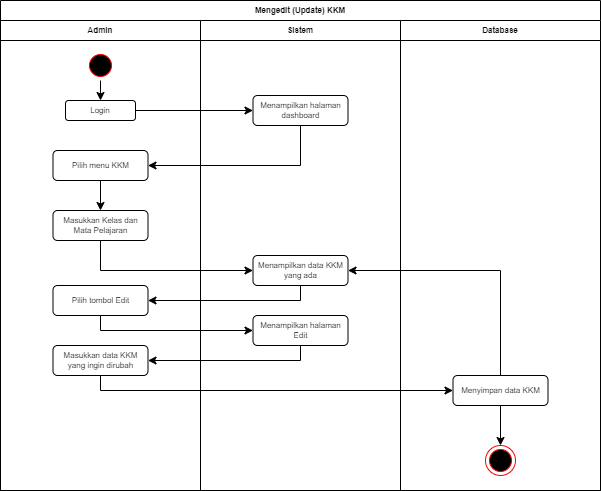
1. Kelola data KKM oleh guru

Guru dapat mengelola data KKM nilai setiap mata pelajaran. Mulai dari melihat, menambah, mengedit, dan menghapus data KKM.



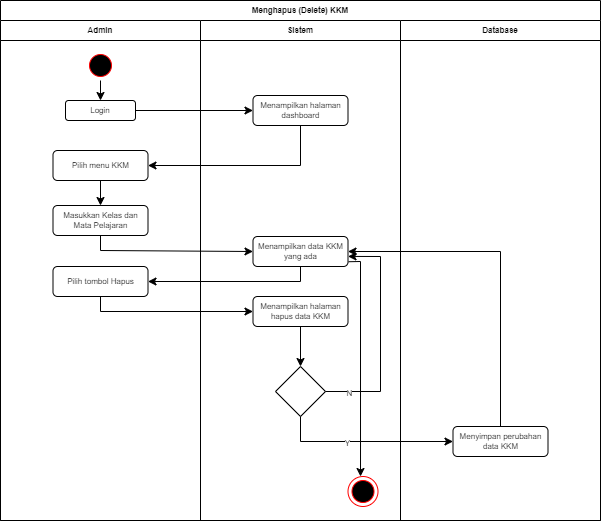
**Gambar 4.** *Activity Diagram* Menambah KKM

Pada gambar 4. Menjelaskan proses guru untuk menambah data KKM. Hal pertama yang dilakukan oleh guru adalah login ke dalam sistem, kemudian masuk ke dalam menu KKM dan tambah data KKM. Masukkan data KKM yang valid dan secara otomatis akan tersimpan ke dalam database.



**Gambar 4.** *Activity Diagram* Mengedit KKM

Pada gambar 4. Menjelaskan proses guru untuk mengedit data KKM. Hal pertama yang dilakukan oleh guru adalah login ke dalam sistem, kemudian masuk ke dalam menu KKM dan edit data KKM. Masukkan data KKM yang ingin dirubah dan secara otomatis akan tersimpan ke dalam database.

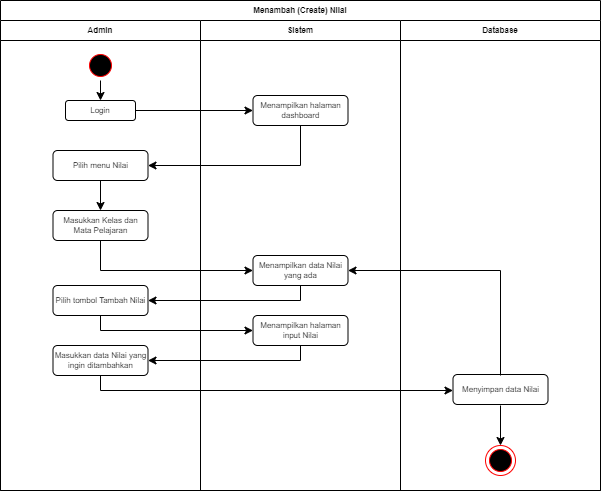


**Gambar 4.** *Activity Diagram* Menghapus KKM

Pada gambar 4. Menjelaskan proses guru untuk menghapus data KKM. Hal pertama yang dilakukan oleh guru adalah login ke dalam sistem, kemudian masuk ke dalam menu KKM dan hapus data KKM. Pastikan data KKM yang ingin dihapus sudah cek kembali, agar tidak terjadi kesalahan yang tidak diinginkan dan secara otomatis system akan menghapus data guru di dalam database.

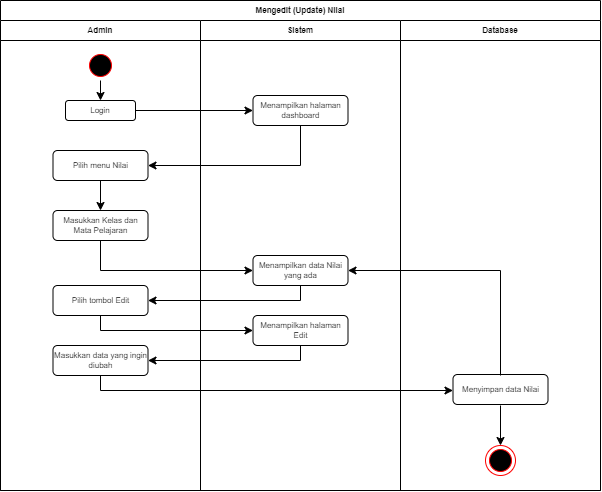
1. Kelola data nilai oleh guru

Guru dapat mengelola data Nilai setiap mata pelajaran. Mulai dari melihat, menambah, mengedit, dan menghapus data nilai.



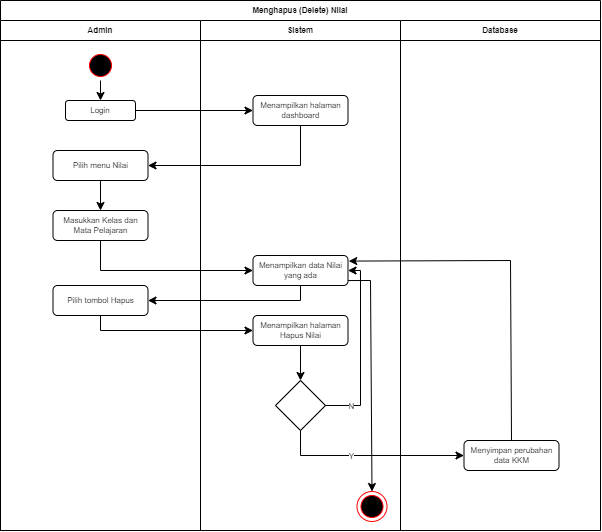
**Gambar 4.** Activity Diagram Menambah Nilai

Pada gambar 4. Menjelaskan proses guru untuk menambah data nilai. Hal pertama yang dilakukan oleh guru adalah login ke dalam sistem, kemudian masuk ke dalam menu nilai dan tambah data nilai. Masukkan data nilai yang valid dan secara otomatis akan tersimpan ke dalam database.



**Gambar 4.** *Activity Diagram* Mengedit Nilai

Pada gambar 4. Menjelaskan proses guru untuk mengedit data nilai. Hal pertama yang dilakukan oleh guru adalah login ke dalam sistem, kemudian masuk ke dalam menu nilai dan edit data nilai. Masukkan data nilai yang ingin dirubah dan secara otomatis akan tersimpan ke dalam database.

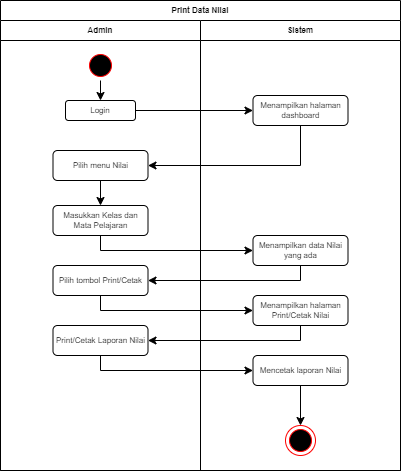


**Gambar 4.** *Activity Diagram* Menghapus Nilai

Pada gambar 4. Menjelaskan proses guru untuk menghapus data nilai. Hal pertama yang dilakukan oleh guru adalah login ke dalam sistem, kemudian masuk ke dalam menu nilai dan hapus data nilai. Pastikan data nilai yang ingin dihapus sudah cek kembali, agar tidak terjadi kesalahan yang tidak diinginkan dan secara otomatis system akan menghapus data nilai di dalam database.

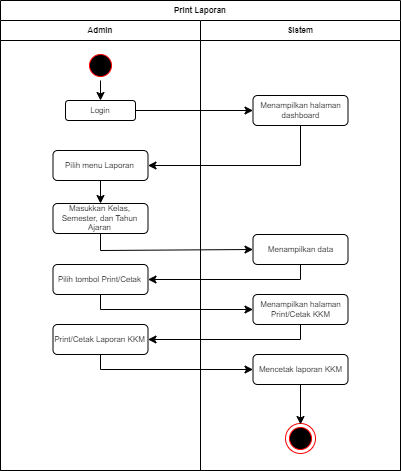
1. Mencetak atau print laporan oleh guru

Guru juga dapat memprint atau mencetak data yang ada untuk dijadikan laporan.



**Gambar 4.** *Activity Diagram* Print Nilai

Pada gambar 4. Menjelaskan proses guru untuk mencetak laporan Nilai. Hal pertama yang dilakukan oleh guru adalah login ke dalam sistem, kemudian masuk ke dalam menu laporan dan cetak data Nilai. Pada bagian laporan, guru dapat mencetak data menjadi dua tipe data yaitu data excel dan data pdf.

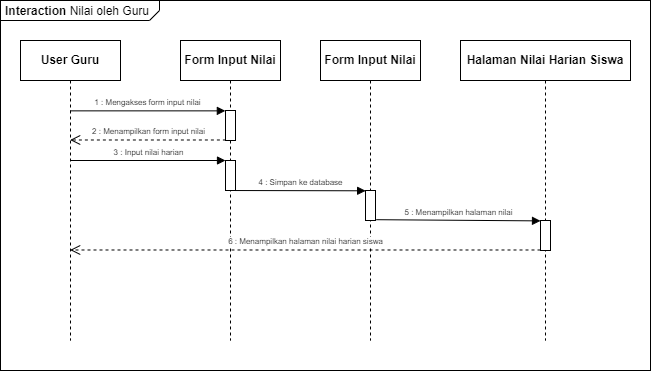


**Gambar 4.** *Activity Diagram* Print Laporan

Pada gambar 4. Menjelaskan proses guru untuk mencetak laporan KKM. Hal pertama yang dilakukan oleh guru adalah login ke dalam sistem, kemudian masuk ke dalam menu laporan dan cetak data KKM. Pada bagian laporan, guru dapat mencetak data menjadi dua tipe data yaitu data excel dan data pdf.

## **4.5 Sequence Diagram**

Pada sistem informasi pengolahan nilai harian siswa, sequence diagram digunakan untuk menggambarkan proses input nilai oleh guru. Diagram ini akan menunjukkan interaksi antara objek-objek pada sistem seperti halaman input nilai, database, dan objek-objek lain yang terkait dengan proses input nilai, proses input nilai oleh guru dapat dilihat pada **Gambar 4.**



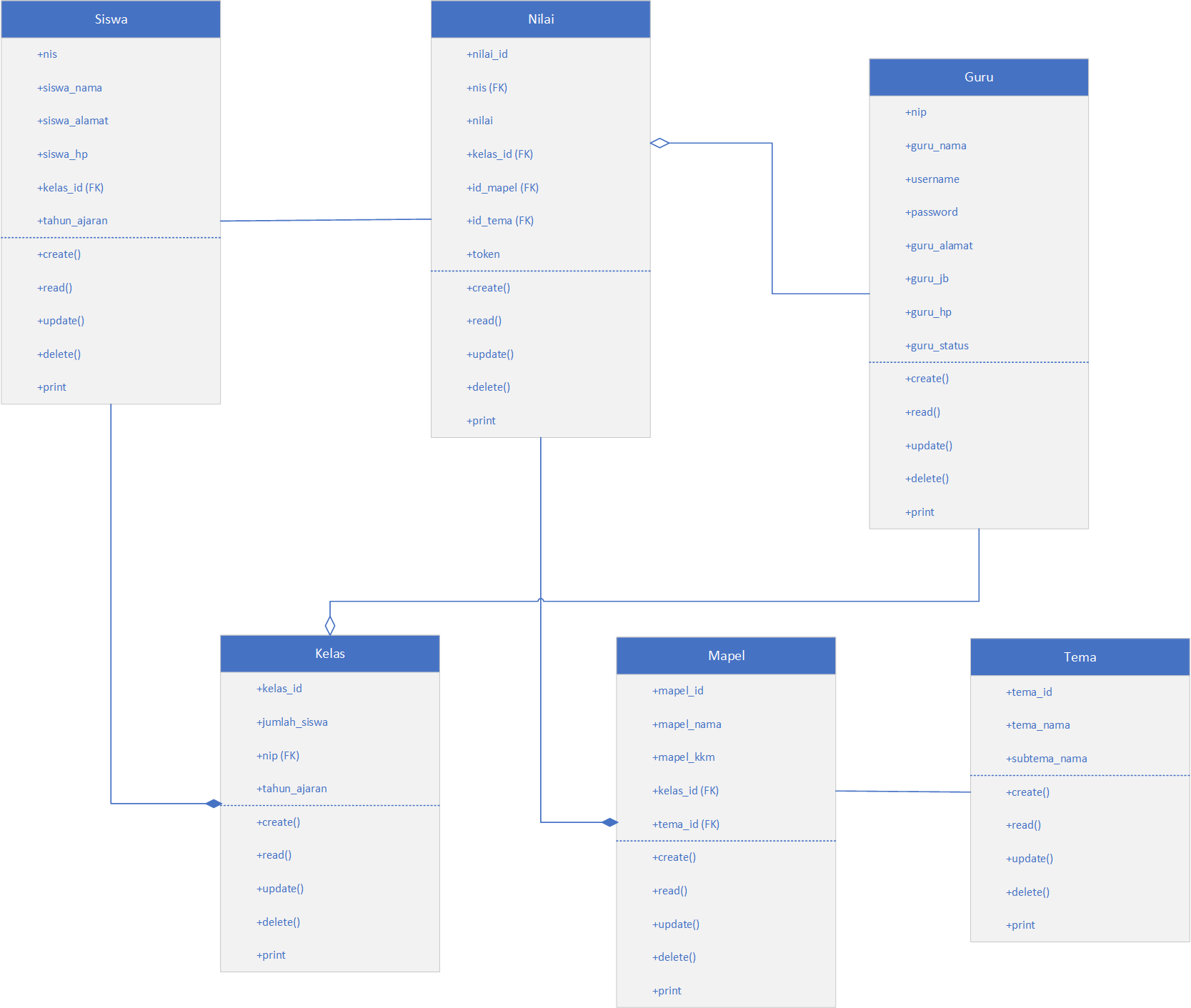
**Gambar 4.** *Sequence Diagram* Input Nilai Siswa oleh Guru

Gambar 4. Menjelaskan proses guru menginputkan nilai siswa pada aplikasi, berikut deskripsi lengkapnya:

1. Sequence diagram dimulai ketika seorang guru ingin memasukkan nilai harian siswa ke dalam sistem.
2. Guru membuka halaman nilai pada sistem informasi pengolahan nilai harian siswa.
3. Sistem menampilkan halaman nilai yang terdiri dari kolom pencarian, menu tambah, edit, hapus, dan print. Untuk menambah nilai, guru masuk ke dalam menu tambah nilai. Sistem menampilkan halaman input nilai yang terdiri dari beberapa kolom yang harus diisi oleh guru.
4. Guru mengisi form tersebut dengan data nilai harian siswa yang akan dimasukkan ke dalam sistem. Setelah selesai mengisi form tersebut, guru menekan tombol "simpan" pada halaman input nilai.
5. Sistem menerima data yang diinput oleh guru dan melakukan validasi terhadap data tersebut.
6. Jika data yang dimasukkan valid, sistem menyimpan data nilai harian siswa ke dalam database dan menampilkan pesan sukses ke layar.
7. Jika terdapat kesalahan pada data yang dimasukkan, sistem menampilkan pesan error yang menjelaskan kesalahan tersebut ke layar dan meminta guru untuk memperbaiki data yang dimasukkan.
8. Setelah selesai memasukkan nilai untuk siswa pertama, guru dapat memasukkan nilai untuk siswa lain dengan cara yang sama.
9. Ketika guru telah selesai memasukkan nilai untuk semua siswa, guru dapat melihat semua nilai yang sudah diinputkan pada halaman nilai.

## **4.6 Class Diagram**

Class Diagram Aplikasi Pengolahan Nilai Harian Siswa adalah suatu diagram yang digunakan untuk merepresentasikan hubungan antar kelas dalam suatu aplikasi pengolahan nilai harian siswa. Class diagram pada Gambar 4. dibawah ini menggambarkan hubungan antara entity class yang dibuat dalam pembangunan Sistem Pengolahan Nilai Siswa :



**Gambar 4.** *Class Diagram* Aplikasi Pengolahan Nilai Harian Siswa

Berikut adalah kelas-kelas yang terdapat dalam class diagram tersebut:

1. Siswa merupakan kelas yang merepresentasikan siswa-siswa yang terdaftar pada aplikasi pengolahan nilai harian. Kelas ini memiliki atribut seperti nama, kelas, nomor induk siswa, alamat, no. hp, dan tahun ajaran.
2. Guru merupakan kelas yang merepresentasikan guru-guru yang mengajar di sekolah pada aplikasi pengolahan nilai harian. Kelas ini dapat memiliki atribut seperti nomot induk pendidikan, nama, username, password, jabatan, no. hp, alamat, dan status.
3. Nilai merupakan kelas yang merepresentasikan nilai-nilai harian yang diperoleh oleh siswa pada suatu mata pelajaran tertentu. Kelas ini dapat memiliki atribut seperti nomor induk siswa, nama, nilai, mata pelajaran, tema, dan sub tema.
4. Kelas merupakan kelas yang merepresentasikan kelas-kelas yang terdaftar pada aplikasi pengolahan nilai harian. Kelas ini dapat memiliki atribut seperti kelas, jumlah siswa, wali kelas, dan tahun ajaran.
5. Mata Pelajaran: merupakan kelas yang merepresentasikan mata pelajaran yang diajarkan pada suatu kelas. Kelas ini dapat memiliki atribut seperti nama mata pelajaran, jumlah jam pelajaran, dan sebagainya.
6. Tema merupakan kelas yang merepresentasikan tema-tema yang dibahas pada suatu periode tertentu. Kelas ini dapat memiliki atribut seperti tema dan sub tema.

Dalam class diagram ini, terdapat beberapa hubungan antara kelas-kelas tersebut, seperti:

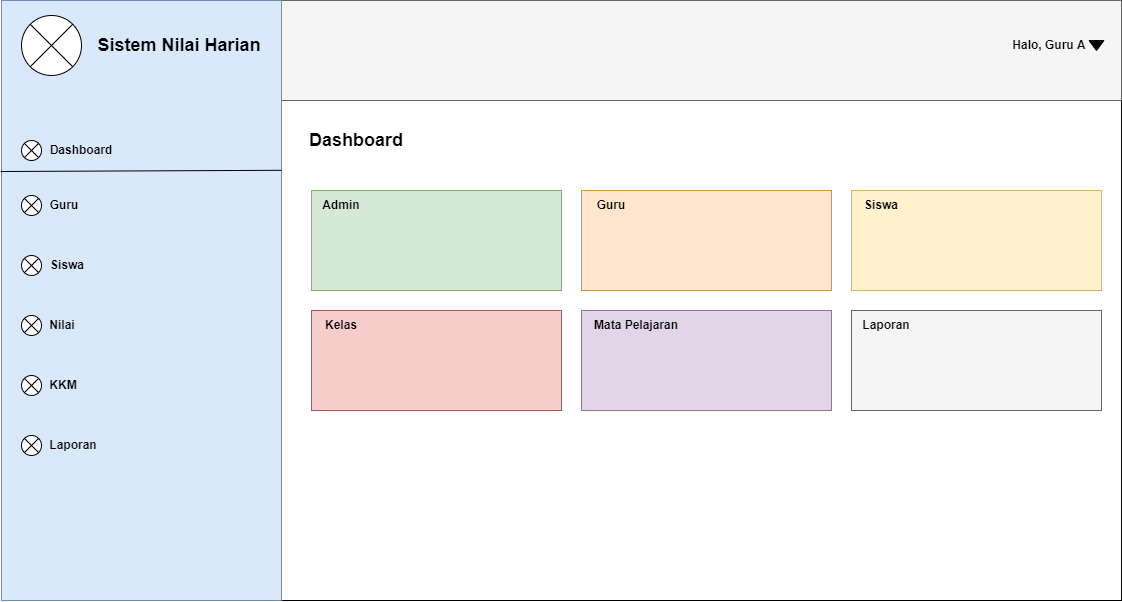
* Siswa dapat terdaftar pada suatu kelas.
* Guru dapat memberikan nilai pada mata pelajaran tertentu.
* Nilai dapat diperoleh oleh siswa pada suatu mata pelajaran tertentu.
* Kelas dapat memiliki beberapa siswa dan mata pelajaran.
* Mata pelajaran dapat diajarkan oleh beberapa guru.
* Tema dapat dibahas pada beberapa mata pelajaran.

Dengan adanya class diagram ini, system informasi pengolahan nilai harian siswa dapat dibangun dengan lebih mudah dan terstruktur, karena hubungan antara kelas-kelas tersebut sudah terdefinisi dengan jelas.

## **4.6 Perancangan Sistem**

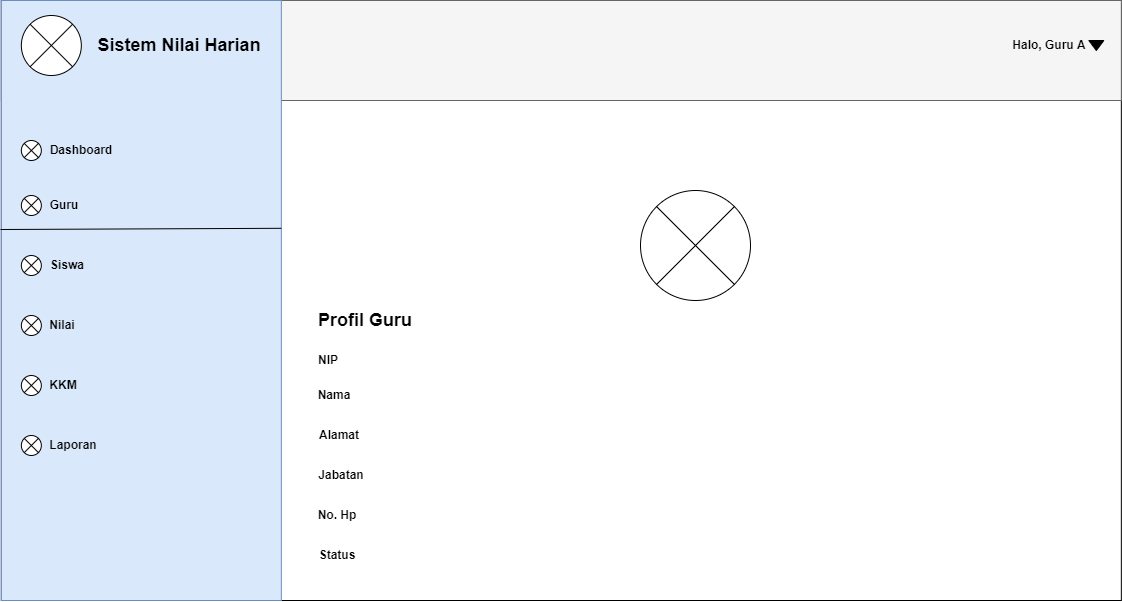
### **4.6.1 Perancangan Sistem Guru**

1. Dashboard Guru



**Gambar 4.** Perancangan Dashboard Guru

1. Profil Guru



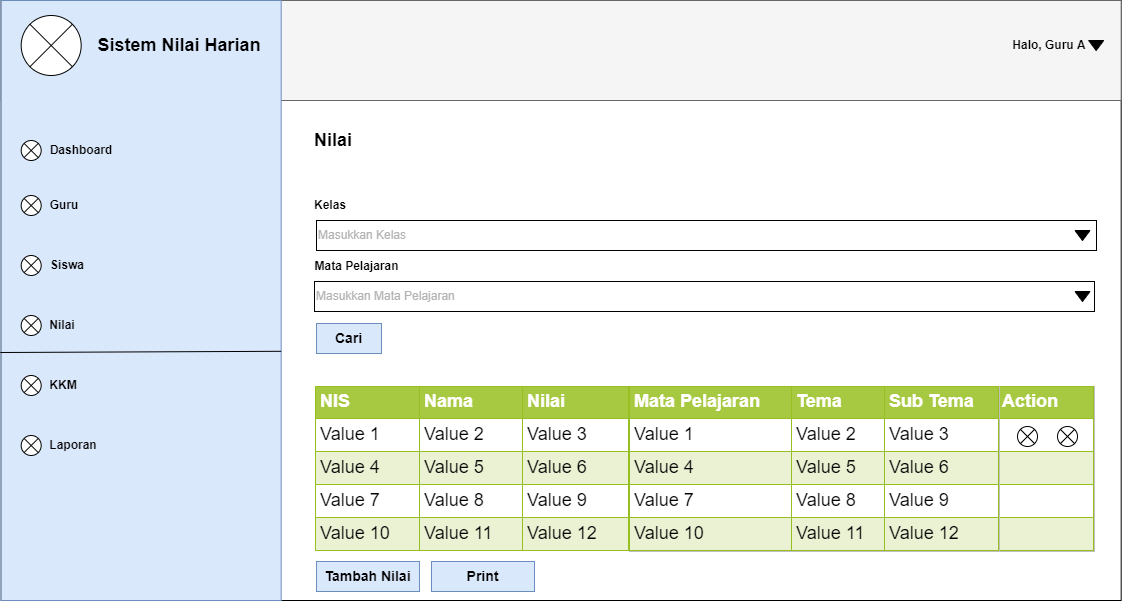
**Gambar 4.** Perancangan Profil Guru

1. Siswa



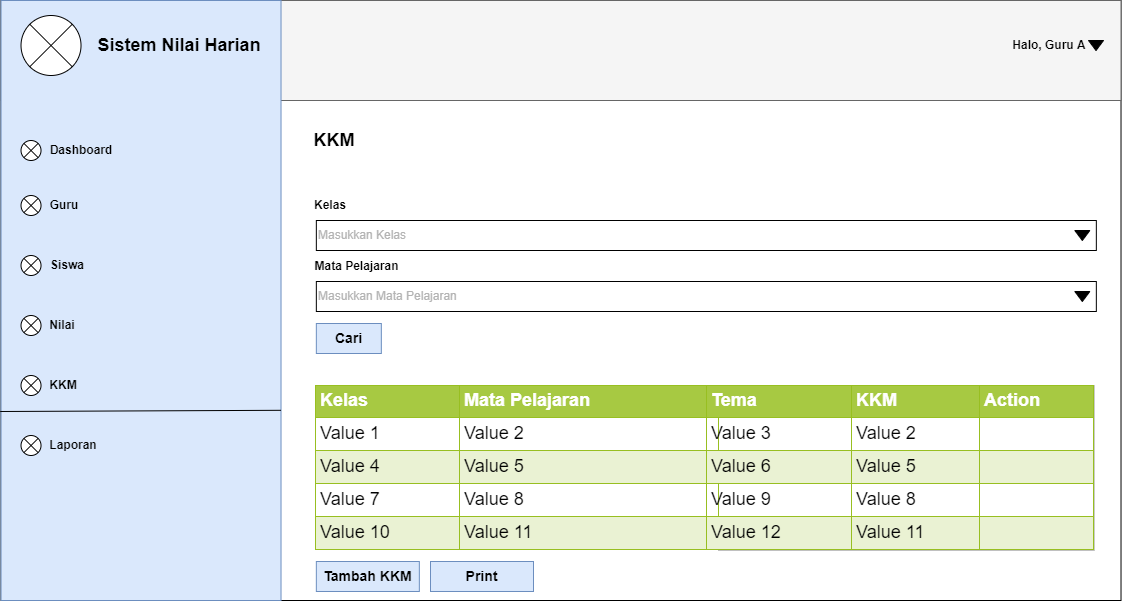
**Gambar 4.** Perancangan Siswa

1. Nilai



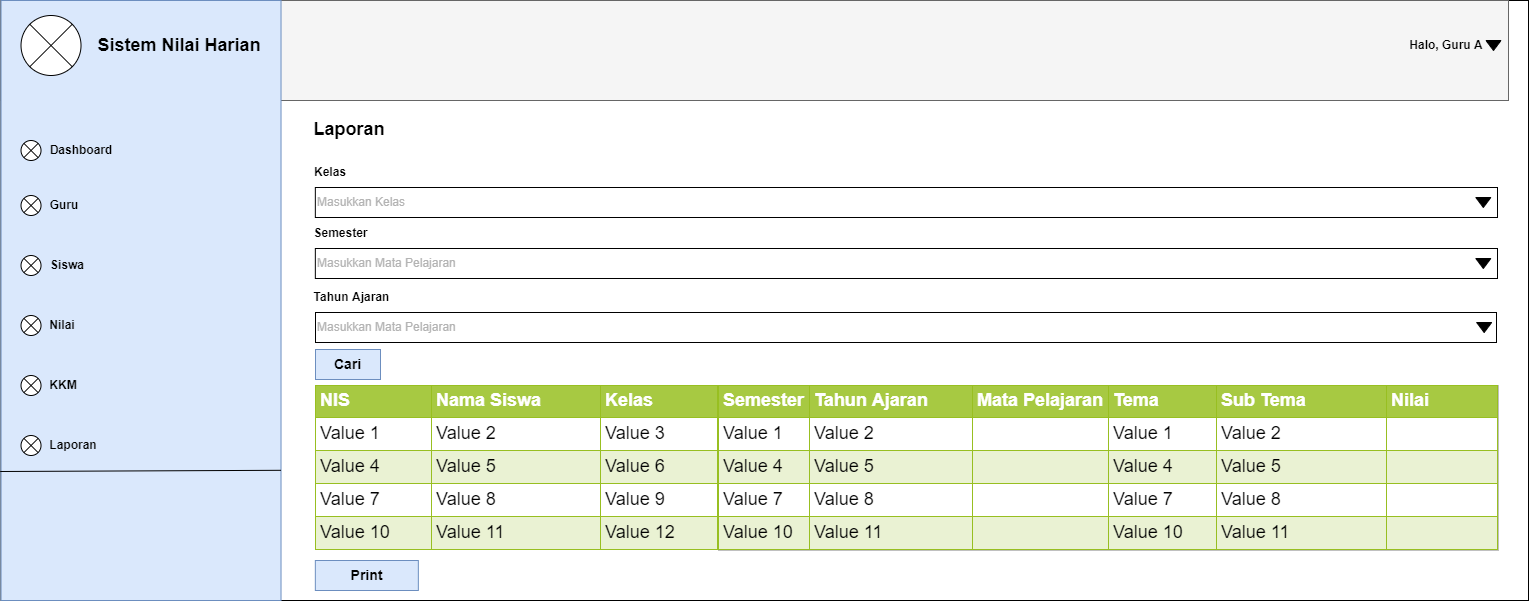
**Gambar 4.** Perancangan Nilai

1. KKM



**Gambar 4.** Perancangan KKM

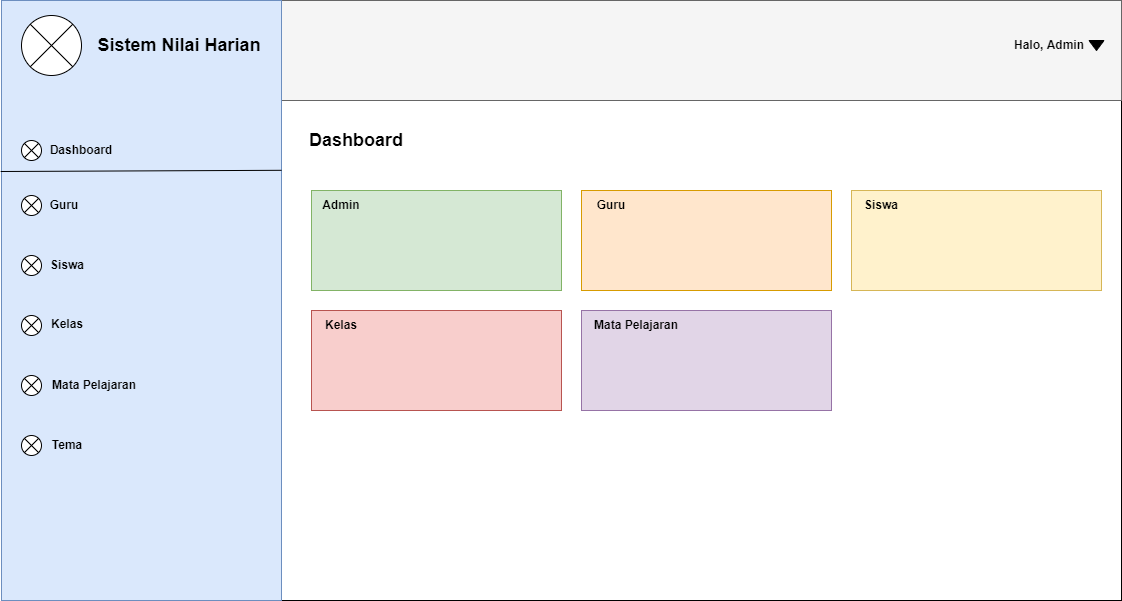
1. Laporan



**Gambar 4.** Perancangan Laporan

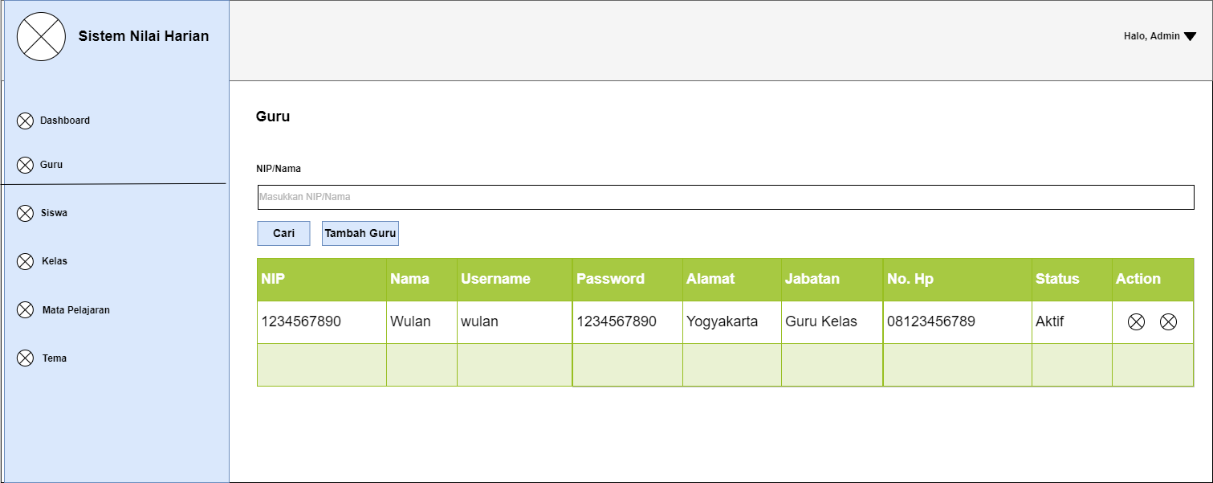
### **4.6.2 Perancangan Sistem Admin**

1. Dashboard Admin



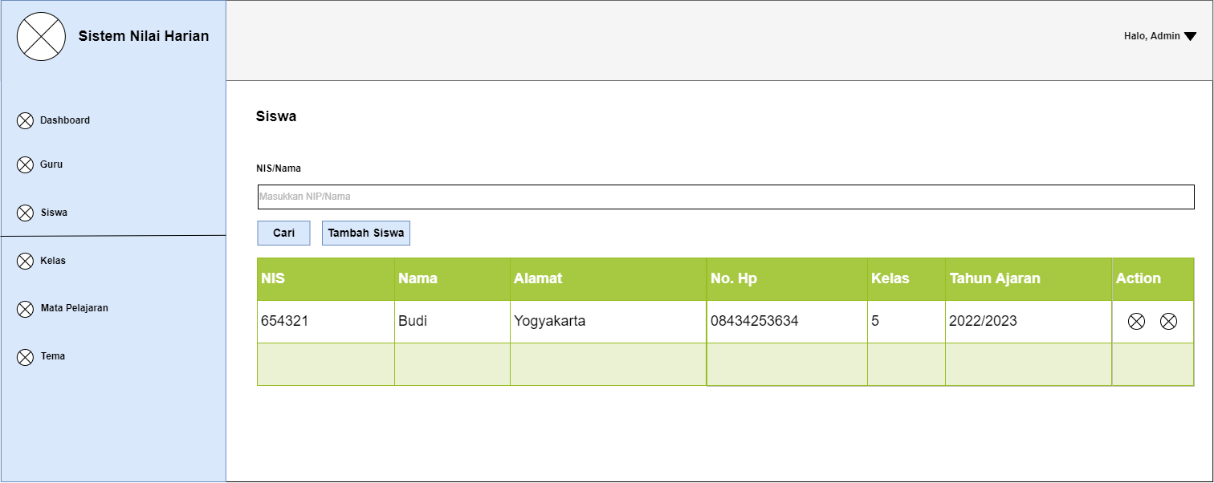
**Gambar 4.** Perancangan Dashboard Admin

1. Guru



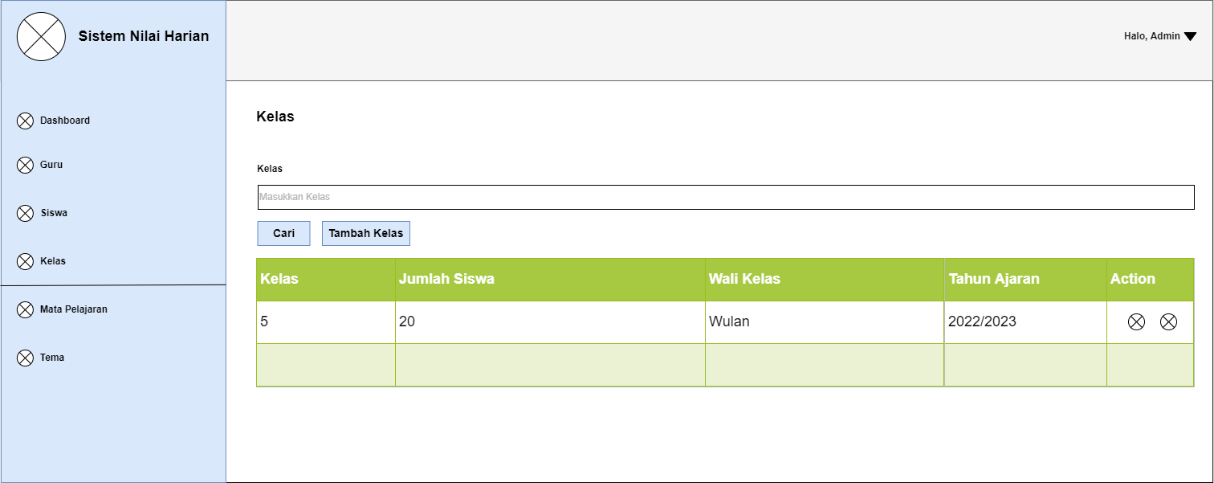
**Gambar 4.** Perancangan Guru

1. Siswa



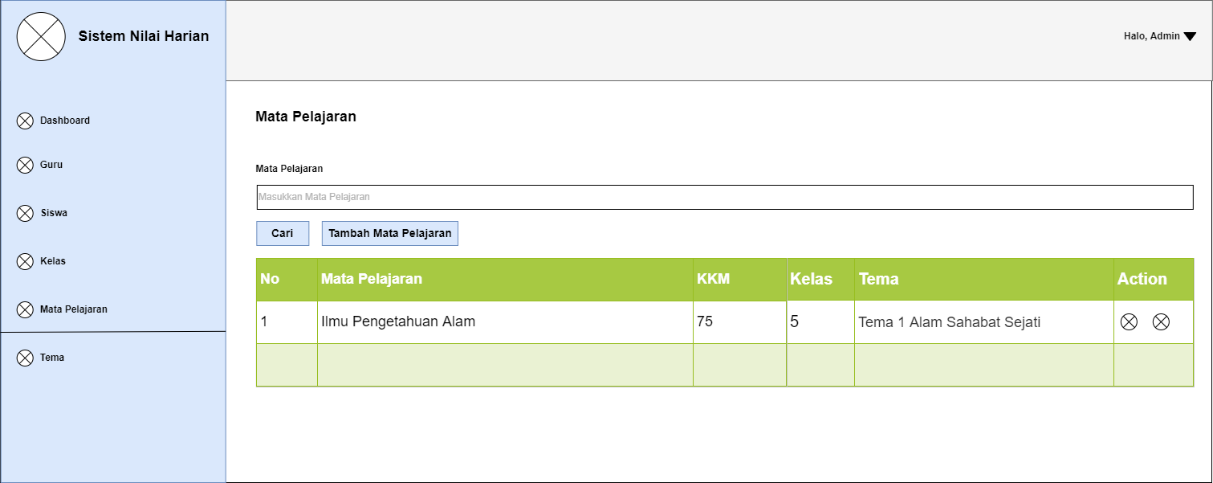
**Gambar 4.** Perancangan Siswa

1. Kelas



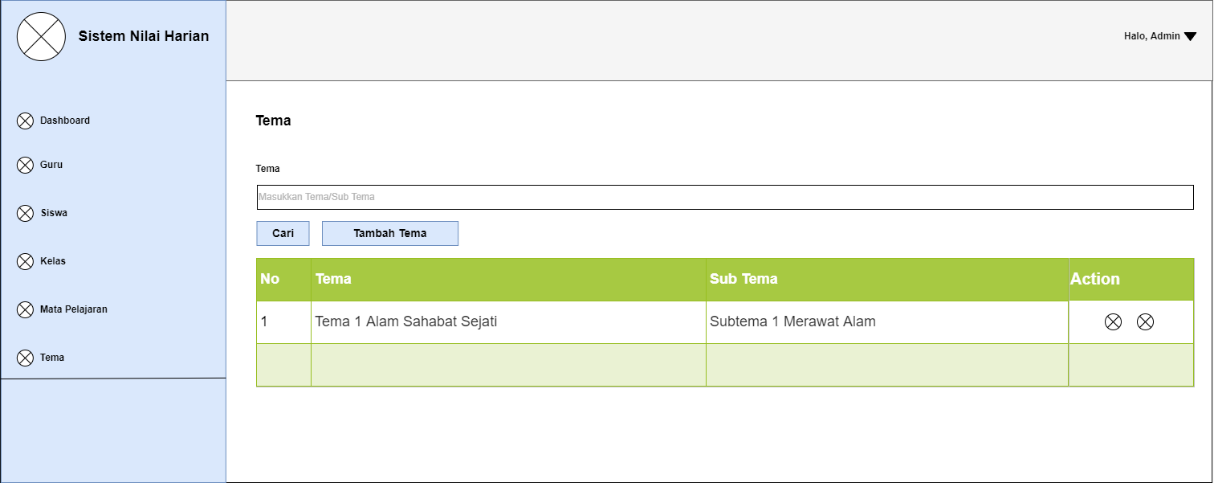
**Gambar 4.** Perancangan Kelas

1. Mata Pelajaran



**Gambar 4.** Perancangan Dashboard Guru

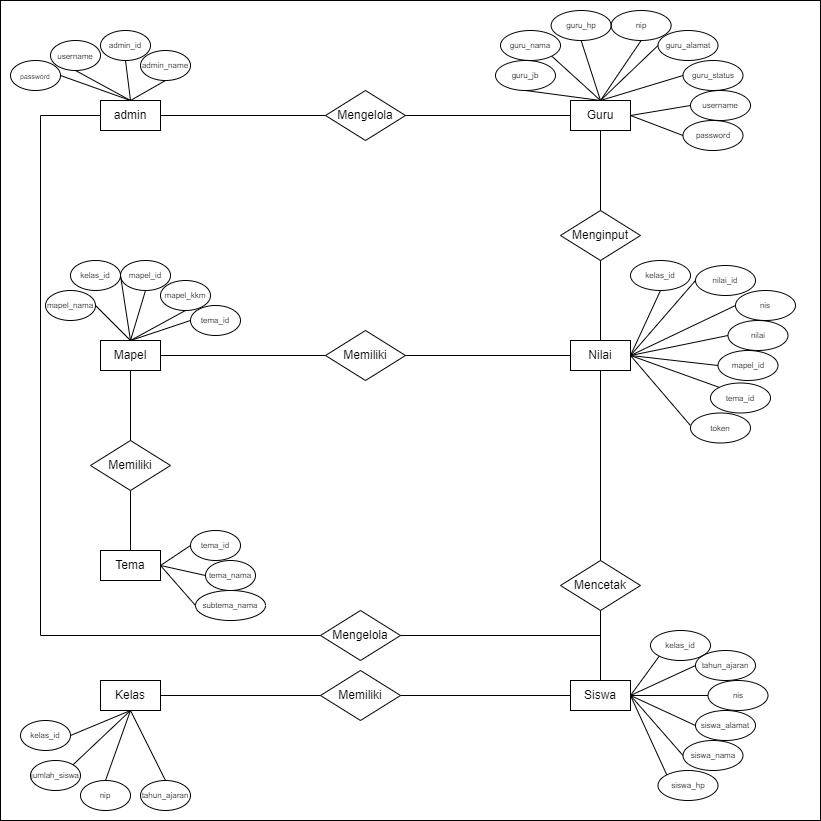
1. Tema



**Gambar 4.** Perancangan Dashboard Guru

## **4.7 Perancangan Database**

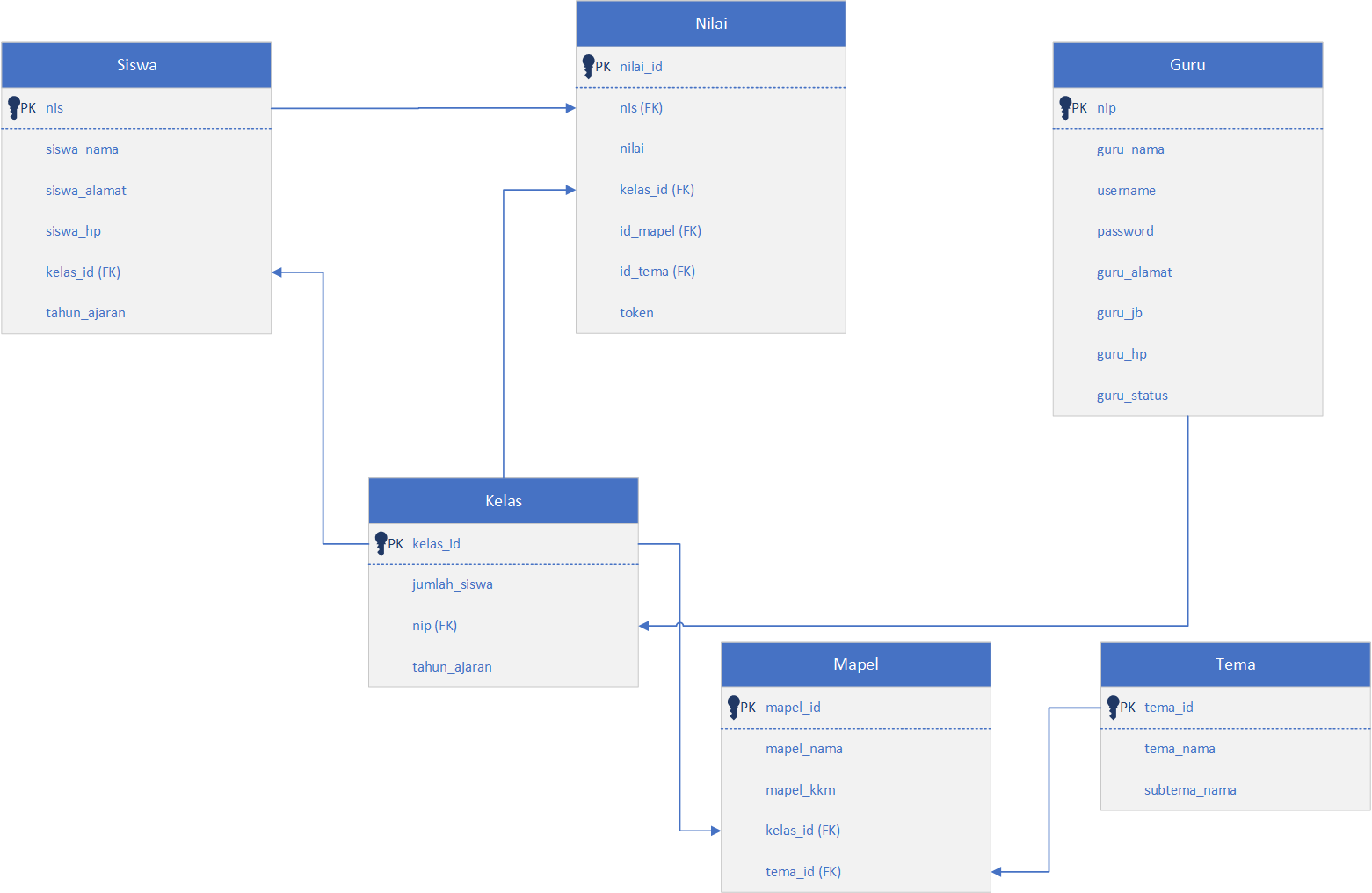
Pada tahap perancangan basis data, pengembang perlu mempertimbangkan entitas-entitas yang akan digunakan dalam sistem seperti siswa, guru, nilai, kelas, dan sebagainya. Kemudian, tabel-tabel dibuat untuk merepresentasikan entitas-entitas tersebut. Tabel-tabel ini akan memiliki kolom-kolom yang merepresentasikan atribut-atribut dari setiap entitas. Perancangan basis data pada Sistem Pengolahan Nilai Harian Siswa berbasis Web menghasilkan pemetaan tabel-tabel yang digambarkan dengan *Entity Relationship Diagram* (ERD) dan *Logical Record Structure* (LRS) yang dapat dilihat pada **gambar 4.** dibawah ini:



**Gambar 4.** *Entity Relationship Diagram* Sistem Pengolahan Data Nilai

Pada gambar 4. Terlihat beberapa tabel yang saling berhubungan satu sama lainnya, seperti tabel siswa yang berhubungan dengan tabel kelas. Dengan memiliki basis data yang baik dan ERD yang jelas, sistem pengolahan nilai harian siswa berbasis web dapat bekerja dengan baik dan efektif. Sistem ini dapat membantu pengelolaan nilai siswa dengan lebih mudah dan efisien serta meningkatkan kualitas pengajaran di sekolah.

Berikut gambaran *Logical Record Structure* (LRS) dari Sistem Pengelohan Nilai Siswa :



**Gambar 4.** *Logical Relationship Stucture* (LRS) Sistem Pengolahan Nilai Harian Siswa

Dari gambar 4. menjelaskan tentang suatu sistem pengolahan nilai harian siswa yang menggunakan struktur hubungan logis (Logical Relationship Structure/LRS) dan berisi data-data seperti:

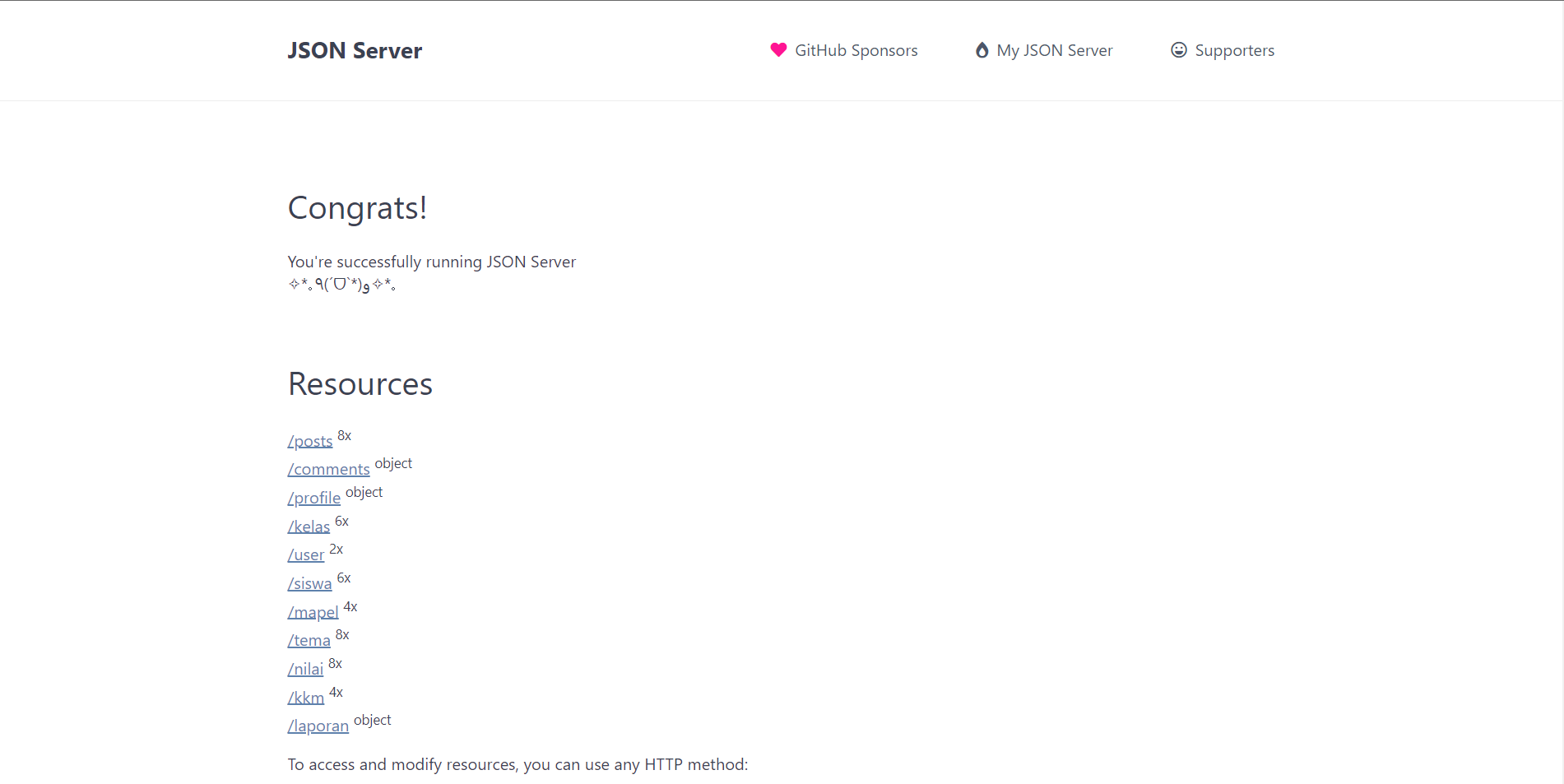
1. Data siswa, mencakup nama siswa, nomor induk siswa, kelas, no. hp, alamat, dan tahun ajaran yang relevan dengan identitas siswa.
2. Data nilai, mencakup nilai-nilai yang diperoleh oleh siswa pada suatu periode pembelajaran tertentu. Ini bisa mencakup nilai untuk tugas dan pekerjaan rumah.
3. Data guru, mencakup nomor induk pendidik, nama, username, password, alamat, no. hp, jabatan, dan status yang relevan dengan identitas guru.
4. Data kelas, mencakup kelas, jumlah siswa, wali kelas, dan tahun ajaran yang relevan dengan identitas kelas.
5. Data mata pelajaran, mencakup nama mata pelajaran, kelas, kkm, tema, dan sub tema yang relevan dengan mata pelajaran yang diajarkan.
6. Data tema, yang mungkin mencakup topik atau tema pembelajaran yang diajarkan dalam suatu periode tertentu.

## **4.8 Implementasi Sistem**

Dalam implementasi Sistem Informasi Pengolahan Nilai Harian SDN 1 Warga Mulya Berbasis Web menggunakan Visual Studio Code sebagai text editor untuk pengambangan sistem, serta JSON server untuk proses pengolahan database.

### **4.6.1 Basis Data**

Berikut ini adalah basis data yang digunakan untuk mengolah data dan informasi pada Sistem Informasi Pengolahan Nilai Harian SDN 1 Warga Mulya Berbasis Web dengan menggunakan JSON server sebagai pengolahan basis data.



**Gambar 4.** *Database* *JSON Server* Aplikasi Pengolahan Nilai Harian Siswa

Dari gambar diatas merupakan list dari beberapa data atau tabel yang terdapat pada JSON Server, dimana data yang disimpan pada setiap tabel tersebut adalah array json. Pemilihan JSON Server ini bukan tanpa alasan, selain penggunaannya yang mudah JSON Server cocok digunakan untuk menguji fungsionalitas dasar pada sebuah aplikasi. JSON Server menyediakan simulasi server REST API dengan menggunakan file JSON sebagai database sementara. Tabel-tabel tersebut merupakan representasi dari data yang jika simpan ke dalam JSON Server, dimana data yang diolah melalui *model* dalam aplikasi Angular. Berikut adalah nama table dan field yang terdapat dalam database Sistem Informasi Pengolahan Nilai Harian SDN 1 Warga Mulya Berbasis Web.

1. Tabel Siswa

Digunakan untuk menyimpan data siswa.

**Tabel 4.** Tabel Siswa

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Field** | **Tipe Data** | **Keterangan** |
| 1 | nis | string | Primary Key |
| 2 | siswa\_nama | string |  |
| 3 | siswa\_alamat | string |  |
| 4 | siswa\_hp | string |  |
| 5 | kelas\_id | string | Foreign Key |
| 6 | tahun\_ajaran | string |  |

Pada tabel 4. Diatas dalam model angular diberi nama SiswaModel yang berisi properti sama seperti pada tabel.

**Kode program 4.** SiswaModel

1. export class SiswaModule { *//siswa model*
2. nis!: string;
3. nama!: string ;
4. alamat!: string ;
5. hp!: string **;**
6. kelas!: string ;
7. ta!: string ;
8. }
9. Tabel Guru

Digunakan untuk menyimpan data guru.

**Tabel 4.** Tabel Guru

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Field** | **Tipe Data** | **Keterangan** |
| 1 | nip | string | Primary Key |
| 2 | guru\_nama | string |  |
| 3 | username | string |  |
| 4 | password | string |  |
| 5 | guru\_alamat | string |  |
| 6 | guru\_jb | string |  |
| 7 | guru\_hp | string |  |
| 8 | guru\_status | string |  |

Pada tabel 4. Diatas dalam model angular diberi nama GuruModel yang berisi properti sama seperti pada tabel.

**Kode program 4.** GuruModel

1. export class GuruModule { *//guru model*
2. nip!: string;
3. nama!: string ;
4. username!: string ;
5. pass!: string ;
6. alamat!: string ;
7. jabatan!: string ;
8. hp!: string ;
9. status!: string ;
10. }
11. Tabel Kelas

Digunakan untuk menyimpan data kelas

**Tabel 4.** Tabel Kelas

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Field** | **Tipe Data** | **Keterangan** |
| 1 | kelas\_id | string | Primary Key |
| 2 | jumlah\_siswa | string |  |
| 3 | nip | string | Foreign Key |
| 4 | tahun\_ajaran | string |  |

Pada tabel 4. Diatas dalam model angular diberi nama KelasModel yang berisi properti sama seperti pada tabel.

**Kode program 4.** KelasModel

1. export class KelasModule { *//kelas model*
2. kelas!: string;
3. jumlah\_siswa!: string ;
4. wali\_kelas!: string ;
5. **ta!: string ;**
6. }
7. Tabel Mapel

Digunakan untuk menyimpan data mapel (mata pelajaran).

**Tabel 4.** Tabel Mapel

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Field** | **Tipe Data** | **Keterangan** |
| 1 | mapel\_id | string | Primary Key |
| 2 | maple | string |  |

Pada tabel 4. Diatas dalam model angular diberi nama MapelModel yang berisi properti sama seperti pada tabel.

**Kode program 4.** MapelModel

1. export class MapelModule { *//mapel model*
2. mapel!: string;
3. }
4. Tabel Tema

Digunakan untuk menyimpan data tema

**Tabel 4.** Tabel Tema

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Field** | **Tipe Data** | **Keterangan** |
| 1 | tema\_id | string | Primary Key |
| 2 | mapel | string |  |
| 3 | tema | string |  |
| 4 | sub\_tema | string |  |

Pada tabel 4. Diatas dalam model angular diberi nama TemaModel yang berisi properti sama seperti pada tabel.

**Kode program 4.** TemaModel

1. export class TemaModule { *//tema model*
2. mapel!: string;
3. tema!: string;
4. sub\_tema!: string ;
5. **}**
6. Tabel Nilai

Digunakan untuk menyimpan data nilai

**Tabel 4.** *Database* Nilai

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Field** | **Tipe Data** | **Keterangan** |
| 1 | nilai\_id | string | Primary Key |
| 2 | nis | string | Foreign key |
| 3 | nama | string |  |
| 4 | nilai | string |  |
| 5 | tema | string | Foreign key |
| 6 | sub\_tema | string | Foreign key |
| 7 | token | string |  |

Pada tabel 4. Diatas dalam model angular diberi nama NilaiModel yang berisi properti sama seperti pada tabel.

**Kode program 4.** NilaiModel

1. export class NilaiModule { *//nilai model*
2. nis!: string;
3. nama!: string ;
4. nilai!: number ;
5. mapel!: string ;
6. tema!: string;
7. sub\_tema!: string;
8. }

## **4.9 Pengkodean**

Dalam pembuatan Sistem Informasi Pengolahan Nilai Harian SDN 1 Warga Mulya Berbasis Web, digunakan pengkodean dengan Framework Angular yang terbagi menjadi dua bagian, yaitu Admin dan Guru. Pengkodean aplikasi ini dilakukan menggunakan aplikasi IDE, yakni Visual Studio Code. Berbagai isi kode yang digunakan dalam aplikasi ini dapat dilihat pada bagian kode program.

### **4.8.1 Guru**

1. Kode login

Berikut merupakan kode program halaman login bagian HTML dan bagian Typescript:

**Kode program 4.** Kode HTML Login

1. <**form** [formGroup]="login" (ngSubmit)="loginApp(login)">
2. <**h2**>LOGIN</**h2**>
3. **<mat-form-field class="full-width-input">**
4. <**input** matInput placeholder="Username" formControlName="username" required>
5. <mat-error>
6. Tolong masukkan Username terlebih dahulu.
7. </mat-error>
8. **</mat-form-field>**
9. <mat-form-field class="full-width-input">
10. <**input** matInput type="password" placeholder="Password" formControlName="pass" required>
11. <mat-error>
12. **Tolong masukkan Password terlebih dahulu.**
13. </mat-error>
14. </mat-form-field>
15. <**button** mat-raised-button color="primary" type="submit"> Login</**button**>
16. </**form**>

**Kode program 4.** Kode Typescript Login

1. loginApp(login: FormGroup) {
2. **this**.service.loginCode(**this**.login.value.username).subscribe(res => {
3. **const** usera = res.find((a: any) => {
4. **return** a.username === **this**.login.value.username && a.pass === **this**.login.value.pass
5. **});**
6. **if** (usera) {
7. **if** (**this**.login.value.username == 'admin') {
8. **this**.data = usera;
9. **this**.toastr.success('Login Sebagai Administrator', 'Login Berhasil');
10. **sessionStorage.setItem('currentUser', JSON.stringify(usera));**
11. **this**.login.reset();
12. **this**.router.navigate(['admin/dashboard-admin']);
13. } **else** {
14. **this**.data = usera;
15. **this.toastr.success('Login Sebagai Guru', 'Login Berhasil');**
16. sessionStorage.setItem('currentUser', JSON.stringify(usera));
17. **this**.login.reset();
18. **this**.router.navigate(['guru/dashboard-guru']);
19. }
20. **} else {**
21. **this**.role = **this**.login.value == '';
22. **this**.toastr.warning('Mohon Cek Kembali Username atau Password Anda', 'Login Gagal');
23. **this**.router.navigate(['/']);
24. }
25. **})**
26. }
27. Kode dashboard guru

Berikut merupakan kode program halaman dashboard guru bagian HTML:

**Kode program 4.** Kode HTML Dashboard Guru

1. <mat-sidenav-container class="sidenav-container">
2. <mat-sidenav #drawer class="sidenav" fixedInViewport [attr.role]="(isHandset$ | async) ? 'dialog' : 'navigation'"
3. [mode]="(isHandset$ | async) ? 'over' : 'side'" [opened]="(isHandset$ | async) === false">
4. <mat-toolbar class="toolbar-logo"> <**img** src="assets/images/tut-wuri.png" class="img"></mat-toolbar>
5. **<mat-nav-list>**
6. <**a** mat-list-item routerLink="./dashboard-guru">Dashboard</**a**>
7. <**a** mat-list-item routerLink="./profile-guru">Profil Guru</**a**>
8. <**a** mat-list-item routerLink="./siswa">Siswa</**a**>
9. <**a** mat-list-item routerLink="./nilai">Nilai</**a**>
10. **<a mat-list-item routerLink="./kkm">KKM</a>**
11. <**a** mat-list-item routerLink="./laporan">Laporan</**a**>
12. </mat-nav-list>
13. </mat-sidenav>
14. <mat-sidenav-content>
15. **<mat-toolbar color="primary">**
16. <**button** type="button" aria-label="Toggle sidenav" mat-icon-button (click)="drawer.toggle()">
17. <mat-icon aria-label="Side nav toggle icon">menu</mat-icon>
18. </**button**>
19. <**span**>Sistem Informasi Nilai Harian</**span**>
20. **<button mat-icon-button color="basic">**
21. <**h3**>| Halo, {{data.nama}} </**h3**>
22. </**button**>
23. <**span** class="example-spacer"></**span**>
24. <**div** class="example-flex-container">
25. **<button mat-icon-button color="basic" routerLink="">**
26. <mat-icon>logout</mat-icon>
27. </**button**>
28. </**div**>
29. </mat-toolbar>
30. *<!-- Add Content Here -->*
31. <router-outlet></router-outlet>
32. </mat-sidenav-content>
33. **</mat-sidenav-container>**
34. Kode profil guru

Berikut merupakan kode program halaman profil guru bagian HTML dan bagian Typescript:

**Kode program 4.** Kode HTML Profil Guru

1. <**div** class="padding">
2. <**img** mat-card-image src="assets/images/tut-wuri.png" class="img">
3. <**h1**>Profil Guru</**h1**>
4. <mat-list>
5. **<mat-list-item>NIP : {{data.nip}}</mat-list-item>**
6. <mat-divider></mat-divider>
7. <mat-list-item>Nama : {{data.nama}}</mat-list-item>
8. <mat-divider></mat-divider>
9. <mat-list-item>Alamat : {{data.alamat}}</mat-list-item>
10. **<mat-divider></mat-divider>**
11. <mat-list-item>Jabatan : {{data.jabatan}}</mat-list-item>
12. <mat-divider></mat-divider>
13. <mat-list-item>No. Hp : {{data.hp}}</mat-list-item>
14. <mat-divider></mat-divider>
15. **<mat-list-item>Status : {{data.status}}</mat-list-item>**
16. </mat-list>
17. </**div**>

Kode program 4. Kode Typescript Profil Guru

1. export class ProfileGuruComponent implements OnInit{
2. data: any;
3. user: any;
4. **constructor(private http: HttpClient,**
5. private api: ApiService,
6. ) { }
7. ngOnInit(): **void** {
8. ***// Mengambil data dari local storage***
9. **const** userData = JSON.parse(sessionStorage.getItem('currentUser')!);
10. *// Menyimpan data di property data*
11. **this**.data = userData;
12. }
13. **}**
14. Kode siswa

Berikut merupakan kode program halaman profile guru bagian HTML dan bagian Typescript:

**Kode program 4.** Kode HTML Siswa

1. ...
2. *<!-- NIS Column -->*
3. <ng-container matColumnDef="nis">
4. <**th** mat-header-cell mat-sort-header="nis" \*matHeaderCellDef mat-sort-header> NIS </**th**>
5. **<td mat-cell \*matCellDef="let element"> {{element.nis}} </td>**
6. </ng-container>
8. *<!-- Nama Column -->*
9. <ng-container matColumnDef="nama">
10. **<th mat-header-cell mat-sort-header="nama" \*matHeaderCellDef mat-sort-header> Nama </th>**
11. <**td** mat-cell \*matCellDef="let element"> {{element.nama}} </**td**>
12. </ng-container>
14. *<!-- Alamat Column -->*
15. **<ng-container matColumnDef="alamat">**
16. <**th** mat-header-cell mat-sort-header="alamat" \*matHeaderCellDef mat-sort-header> Alamat </**th**>
17. <**td** mat-cell \*matCellDef="let element"> {{element.alamat}} </**td**>
18. </ng-container>
19. ...

**Kode program 4.** Kode Typescript Siswa

1. ...
2. getDataSiswa() { *//untuk ambil data siswa*
3. **this**.api.ambilDataSiswa()
4. .subscribe(res => {
5. ***// console.log(res);***
6. **this**.siswas = res;
7. **this**.dataSiswa = **new** MatTableDataSource(**this**.siswas);
8. **this**.dataSiswa.paginator = **this**.paginator;
9. **this**.dataSiswa.sort = **this**.sort;
10. **})**
11. }
12. ...
13. Kode nilai

Berikut merupakan kode program halaman nilai bagian HTML dan bagian Typescript:

**Kode program 4.** Kode HTML Nilai

1. ...
2. *<!-- NIS Column -->*
3. <ng-container matColumnDef="nis">
4. <**th** mat-header-cell \*matHeaderCellDef> NIS </**th**>
5. **<td mat-cell \*matCellDef="let element"> {{element.nis}} </td>**
6. </ng-container>
8. *<!-- Nama Column -->*
9. <ng-container matColumnDef="nama">
10. **<th mat-header-cell \*matHeaderCellDef> Nama </th>**
11. <**td** mat-cell \*matCellDef="let element"> {{element.nama}} </**td**>
12. </ng-container>
14. *<!-- Nilai Column -->*
15. **<ng-container matColumnDef="nilai">**
16. <**th** mat-header-cell \*matHeaderCellDef> Nilai </**th**>
17. <**td** mat-cell \*matCellDef="let element" [ngStyle]="{'color': element.nilai < 65 ? 'red' : 'black'}">{{element.nilai}}</**td**>
18. </ng-container>
19. ...

**Kode program 4.** Kode Typescript Nilai

1. ...
2. getDataNilai() { *//untuk ambil data*
3. **this**.api.ambilDataNilai()
4. .subscribe(res => {
5. **this.nilais = res;**
6. console.log();
7. **this**.dataNilai = **new** MatTableDataSource(**this**.nilais);
8. **this**.dataNilai.paginator = **this**.paginator;
9. **this**.dataNilai.sort = **this**.sort;
10. **})**
11. }
12. ...

**Kode program 4.** Kode HTML Tambah Nilai

1. <**form** [formGroup]="dataNilaiForm" (ngSubmit)="submit()">
2. ...
3. <**div**>
4. <mat-form-field appearance="outline" class="field">
5. **<mat-label>Nilai</mat-label>**
6. <**input** matInput type="text" formControlName="nilai" required>
7. <mat-error>Masukkan Nilai terlebih dahulu</mat-error>
8. </mat-form-field>
9. </**div**>
10. **...**
11. </**form**>

**Kode program 4.** Kode Typescript Tambah Nilai

1. ...
2. submit() {
3. *// Validasi form*
4. **if** (**this**.dataNilaiForm.invalid) {
5. *// Tampilkan pesan kesalahan*
6. **this.toastr.error('Gagal Menambah Data!!!', 'Data Nilai');**
7. **return**;
8. }
10. **const** newData = **this**.dataNilaiForm.value;
11. **const temaInput = newData.tema;**
12. **const** subTemaInput = newData.sub\_tema;
13. **const** nisInput = newData.nis;
14. **this**.api.getNilaiId(temaInput)
15. .subscribe(existingData => {
16. **const isExisting = existingData.some(data => {**
17. **const** temaData = data.tema;
18. **const** subTemaData = data.sub\_tema;
19. **const** nisData = data.nis;
20. **return** (
21. **encodeURIComponent(temaData) === encodeURIComponent(temaInput) &&**
22. encodeURIComponent(subTemaData) === encodeURIComponent(subTemaInput) &&
23. nisData === nisInput
24. );
25. });
27. **if** (isExisting) {
28. *// Nilai dengan Tema yang sama sudah ada, tampilkan pesan kesalahan*
29. **this**.toastr.error('Sub Tema pada Tema Mata Pelajaran sudah diberi Nilai. Data tidak dapat ditambahkan.', 'Data Nilai');
30. } **else** {
31. **this.simpan(); *// Lakukan simpan data***
32. }
33. }, error => {
34. console.log(error);
35. });
36. **}**
38. simpan() {
39. **const** nilai = **this**.dataNilaiForm?.**get**('nilai')?.value ?? 0;
40. **const** kkm = **this**.kkm.find((item: { kkm: any; }) => item.kkm === **this**.dataNilaiForm.**get**('kkm')?.value)?.kkm ?? 65;
41. **if (kkm !== 0 && nilai < kkm) {**
42. **this**.toastr.error(`Nilai yang dimasukkan (${nilai}) kurang dari KKM (${kkm})`, 'Masuk Kategori Remidial');
43. **this**.api.tambahdataNilai(**this**.dataNilaiForm.value)
44. .subscribe(res => {
45. **this**.toastr.success('Berhasil Mengupdate Data!!!', 'Data Nilai');
46. **this.dialogref.close();**
47. setTimeout(() => {
48. window.location.reload();
49. }, 5500);
50. })
51. **return;**
52. } **else** {
53. **this**.api.tambahdataNilai(**this**.dataNilaiForm.value)
54. .subscribe(res => {
55. **this**.toastr.success('Berhasil Menambah Data!!!', 'Data Nilai');
56. **this.dialogref.close();**
57. setTimeout(() => {
58. window.location.reload();
59. }, 5500);
60. })
61. **}**
62. ...

**Kode program 4.** Kode HTML Edit Nilai

1. <**form** [formGroup]="dataNilaiForm" (ngSubmit)="submit()">
2. ...
3. <**div**>
4. <mat-form-field appearance="outline" class="field">
5. **<mat-label>Nilai</mat-label>**
6. <**input** matInput type="text" formControlName="nilai" required>
7. <mat-error>Masukkan Nilai terlebih dahulu</mat-error>
8. </mat-form-field>
9. </**div**>
10. **...**
11. </**form**>

**Kode program 4.** Kode Typescript Edit Nilai

1. ...
2. submit() {
3. *// Validasi form*
4. **if** (**this**.dataNilaiForm.invalid) {
5. ***// Tampilkan pesan kesalahan***
6. **this**.toastr.error('Gagal Menambah Data!!!', 'Data Nilai');
7. **return**;
8. }
9. **this**.simpan(); *// Lakukan simpan data*
10. **}**
12. simpan() {
13. **const** nilai = **this**.dataNilaiForm?.**get**('nilai')?.value ?? 0;
14. **const** kkm = **this**.kkm.find((item: { kkm: any; }) => item.kkm === **this**.dataNilaiForm.**get**('kkm')?.value)?.kkm ?? 65;
15. **if (kkm !== 0 && nilai < kkm) {**
16. **this**.toastr.error(`Nilai yang dimasukkan (${nilai}) kurang dari KKM (${kkm})`, 'Masuk Kategori Remidial');
17. **this**.api.ubahDataNilai(**this**.data.id, **this**.dataNilaiForm.value)
18. .subscribe(res => {
19. **this**.toastr.success('Berhasil Mengupdate Data!!!', 'Data Nilai');
20. **this.dialogref.close();**
21. setTimeout(() => {
22. window.location.reload();
23. }, 5500);
24. })
25. **return;**
26. } **else** {
27. **this**.api.ubahDataNilai(**this**.data.id, **this**.dataNilaiForm.value)
28. .subscribe(res => {
29. **this**.toastr.success('Berhasil Mengupdate Data!!!', 'Data Nilai');
30. **this.dialogref.close();**
31. setTimeout(() => {
32. window.location.reload();
33. }, 5500);
34. })
35. **}**
36. }
37. ...

Kode program 4. Kode HTML Hapus Nilai

1. <**div** mat-dialog-content class="content">
2. <**img** [src]="myimage" class="ihapus">
3. <**h1**>Apakah anda yakin ingin menghapus data?</**h1**>
4. </**div**>
5. **<div mat-dialog-actions class="action">**
6. <**button** mat-raised-button color="basic" class="btn-flex" [mat-dialog-close]="true">Batal</**button**>
7. <**button** mat-raised-button color="warn" class="btn-flex" type="submit" (click)="hapusDataNilai()">Hapus</**button**>
8. </**div**>

**Kode program 4.** Kode Typescript Hapus Nilai

1. <**div** mat-dialog-content class="content">
2. <**img** [src]="myimage" class="ihapus">
3. <**h1**>Apakah anda yakin ingin menghapus data?</**h1**>
4. </**div**>
5. **<div mat-dialog-actions class="action">**
6. <**button** mat-raised-button color="basic" class="btn-flex" [mat-dialog-close]="true">Batal</**button**>
7. <**button** mat-raised-button color="warn" class="btn-flex" type="submit" (click)="hapusDataNilai()">Hapus</**button**>
8. </**div**>
9. Kode KKM

Berikut merupakan kode program halaman KKM bagian HTML dan bagian Typescript:

**Kode program 4.** Kode HTML KKM

1. ...
2. <**table** mat-table [dataSource]="dataKkm" class="mat-elevation-z8">
3. ...
4. *<!-- KKM Column -->*
5. **<ng-container matColumnDef="kkm">**
6. <**th** mat-header-cell \*matHeaderCellDef> KKM </**th**>
7. <**td** mat-cell \*matCellDef="let element"> {{element.kkm}} </**td**>
8. </ng-container>
9. ...
10. **</table>**
11. ...

**Kode program 4.** Kode Typescript KKM

1. ...
2. getDataKkm() { *//untuk ambil data*
3. **this**.api.ambilDataKkm()
4. .subscribe(res => {
5. **this.kkms = res;**
6. console.log();
7. **this**.dataKkm = **new** MatTableDataSource(**this**.kkms);
8. **this**.dataKkm.paginator = **this**.paginator;
9. **this**.dataKkm.sort = **this**.sort;
11. })
12. }
13. ...

**Kode program 4.** Kode HTML Tambah KKM

1. <form [formGroup]="dataKkmForm" (ngSubmit)="submit()">
2. ...
3. <div>
4. <mat-form-field appearance="outline" class="field">
5. **<mat-label>KKM</mat-label>**
6. <input matInput type="text" formControlName="kkm" required>
7. <mat-error>Masukkan nama terlebih dahulu</mat-error>
8. </mat-form-field>
9. </div>
10. **...**
11. </form>

**Kode program 4.** Kode Typescript Tambah KKM

1. ...
2. submit() {
3. *// Validasi form*
4. **if** (**this**.dataKkmForm.invalid) {
5. ***// Tampilkan pesan kesalahan***
6. **this**.toastr.error('Gagal Menambah Data!!!', 'Data KKM');
7. **return**;
8. }
10. **const newData = this.dataKkmForm.value;**
11. **const** temaInput = newData.tema;
12. **const** subTemaInput = newData.sub\_tema;
13. **this**.api.getKkmId(temaInput)
14. .subscribe(existingData => {
15. **const isExisting = existingData.some(data => {**
16. **const** temaData = data.tema;
17. **const** subTemaData = data.sub\_tema;
18. **return** encodeURIComponent(temaData) === encodeURIComponent(temaInput) &&
19. encodeURIComponent(subTemaData) === encodeURIComponent(subTemaInput);
20. **});**
21. **if** (isExisting) {
22. *// Mata Pelajaran dengan Tema yang sama sudah ada, tampilkan pesan kesalahan*
23. **this**.toastr.error('KKM pada Mata Pelajaran sudah ada. Data tidak dapat ditambahkan.', 'Data KKM');
24. } **else** {
25. ***// Mata Pelajaran dengan Tema yang sama belum ada, lakukan simpan data***
26. **this**.api.tambahDataKkm(newData)
27. .subscribe(res => {
28. **this**.toastr.success('Berhasil Menambah Data!!!', 'Data KKM');
29. **this**.dialogref.close();
30. **setTimeout(() => {**
31. window.location.reload();
32. }, 5500);
33. });
34. }
35. **}, error => {**
36. console.log(error);
37. });
38. }
39. ...

**Kode program 4.** Kode HTML Edit KKM

1. <**form** [formGroup]="dataKkmForm" (ngSubmit)="submit()">
2. ...
3. <**div**>
4. <mat-form-field appearance="outline" class="field">
5. **<mat-label>KKM</mat-label>**
6. <**input** matInput type="text" formControlName="kkm" required>
7. <mat-error>Masukkan nama terlebih dahulu</mat-error>
8. </mat-form-field>
9. </**div**>
10. **...**
11. </**form**>

**Kode program 4.** Kode Typescript Edit KKM

1. ...
2. submit() {
3. *// Validasi form*
4. **if** (**this**.dataKkmForm.invalid) {
5. ***// Tampilkan pesan kesalahan***
6. **this**.toastr.error('Gagal Menambah Data!!!', 'Data KKM');
7. **return**;
8. }
9. **this**.simpan(); *// Lakukan simpan data*
10. **}**
12. simpan() {
13. **this**.api.ubahDataKkm(**this**.data.id, **this**.dataKkmForm.value).subscribe(res => {
14. **this**.toastr.success('Berhasil Mengupdate Data!!!', 'Data KKM');
15. **this.dialogref.close();**
16. setTimeout(() => {
17. window.location.reload();
18. }, 5500);
19. },
20. **)**
21. }
22. ...

**Kode program 4.** Kode HTML Hapus KKM

1. <**div** mat-dialog-content class="content">
2. <**img** [src]="myimage" class="ihapus">
3. <**h1**>Apakah anda yakin ingin menghapus data?</**h1**>
4. </**div**>
5. **<div mat-dialog-actions class="action">**
6. <**button** mat-raised-button color="basic" class="btn-flex" [mat-dialog-close]="true">Batal</**button**>
7. <**button** mat-raised-button color="warn" class="btn-flex" type="submit" (click)="hapusDataKkm()">Hapus</**button**>
8. </**div**>

**Kode program 4.** Kode Typescript Hapus KKM

1. ...
2. hapusDataKkm() {
3. **this**.api.hapusDataKkm(**this**.data.id).subscribe(res => {
4. **this**.toastr.success('Berhasil Menghapus Data!!!', 'Data KKM');
5. **this.dialogref.close();**
6. setTimeout(() => {
7. window.location.reload();
8. }, 5500);
9. })
10. **}**
11. ...
12. Kode laporan

Berikut merupakan kode program halaman KKM bagian HTML dan bagian Typescript:

**Kode program 4.** Kode HTML Laporan

1. ...
2. <**table** mat-table [dataSource]="dataLaporan" class="mat-elevation-z8">
3. ...
4. <ng-container matColumnDef="nilaiMapel1">
5. **<th mat-header-cell \*matHeaderCellDef> IPA </th>**
6. <**td** mat-cell \*matCellDef="let element">
7. <**h5** [ngClass]="{'text-danger': element.nilaiPerTema['Tema 1 IPA']?.jumlah < 65}">T1: {{element.nilaiPerTema['Tema 1 IPA']?.jumlah}}</**h5**>
8. <**h5** [ngClass]="{'text-danger': element.nilaiPerTema['Tema 2 IPA']?.jumlah < 65}">T2: {{element.nilaiPerTema['Tema 2 IPA']?.jumlah}}</**h5**>
9. <**h5** [ngClass]="{'text-danger': element.nilaiPerTema['Tema 3 IPA']?.jumlah < 65}">T3: {{element.nilaiPerTema['Tema 3 IPA']?.jumlah}}</**h5**>
10. **<h5 [ngClass]="{'text-danger': element.nilaiPerTema['Tema 4 IPA']?.jumlah < 65}">T4: {{element.nilaiPerTema['Tema 4 IPA']?.jumlah}}</h5>**
11. <**h5** [ngClass]="{'text-danger': element.nilaiPerTema['Tema 5 IPA']?.jumlah < 65}">T5: {{element.nilaiPerTema['Tema 5 IPA']?.jumlah}}</**h5**>
12. <**h5** [ngClass]="{'text-danger': element.nilaiPerTema['Tema 6 IPA']?.jumlah < 65}">T6: {{element.nilaiPerTema['Tema 6 IPA']?.jumlah}}</**h5**>
13. <**h5** [ngClass]="{'text-danger': element.nilaiPerTema['Tema 7 IPA']?.jumlah < 65}">T7: {{element.nilaiPerTema['Tema 7 IPA']?.jumlah}}</**h5**>
14. <**h5** [ngClass]="{'text-danger': element.nilaiPerTema['Tema 8 IPA']?.jumlah < 65}">T8: {{element.nilaiPerTema['Tema 8 IPA']?.jumlah}}</**h5**>
15. **<h5 [ngClass]="{'text-danger': element.nilaiPerTema['Tema 9 IPA']?.jumlah < 65}">T9: {{element.nilaiPerTema['Tema 9 IPA']?.jumlah}}</h5>**
17. Total Nilai: {{element.nilaiPerMapel.IPA.total}} <**br**>
18. </**td**>
19. </ng-container>
20. **...**
21. </**table**>
22. ...

**Kode program 4.** Kode Typescript Laporan

1. ...
2. getDataLaporan() { *//untuk ambil data*
3. **this**.api.ambilDataNilai()
4. .subscribe(res => {
5. **this.laporans = res;**
6. console.log(res);
7. **this**.dataLaporan = **new** MatTableDataSource(**this**.laporans);
8. **this**.dataLaporan.paginator = **this**.paginator;
9. **this**.dataLaporan.sort = **this**.sort;
10. **})**
11. }
12. ...
13. Kode cetak pdf

**Kode program 4.** Kode Typescript Cetak PDF

1. ...
2. cetak() {
3. let pdf = **new** jsPDF('l', 'pt', 'a4');
4. pdf.html(**this**.el.nativeElement, {
5. **callback: (pdf) => {**
6. pdf.save("nilai.pdf")
7. }
8. })
9. }
10. **...**
11. Kode export to excel

**Kode program 4.** Kode Typescript Cetak Excel

1. ...
2. excel() {
3. let element = document.getElementById('nilai');
4. **const** ws: XLSX.WorkSheet = XLSX.utils.table\_to\_sheet(element);
5. **const wb: XLSX.WorkBook = XLSX.utils.book\_new();**
6. XLSX.utils.book\_append\_sheet(wb, ws, 'sheet1');
7. XLSX.writeFile(wb, 'Nilai Rata-Rata Per Tema.xlsx');
8. }
9. ...

### **4.8.2 Admin**

1. Login

**Kode program 4.** Kode HTML Login

1. <**form** [formGroup]="login" (ngSubmit)="loginApp(login)">
2. <**h2**>LOGIN</**h2**>
3. **<mat-form-field class="full-width-input">**
4. <**input** matInput placeholder="Username" formControlName="username" required>
5. <mat-error>
6. Tolong masukkan Username terlebih dahulu.
7. </mat-error>
8. **</mat-form-field>**
9. <mat-form-field class="full-width-input">
10. <**input** matInput type="password" placeholder="Password" formControlName="pass" required>
11. <mat-error>
12. **Tolong masukkan Password terlebih dahulu.**
13. </mat-error>
14. </mat-form-field>
15. <**button** mat-raised-button color="primary" type="submit"> Login</**button**>
16. </**form**>

**Kode program 4.** Kode Typescript Login

1. loginApp(login: FormGroup) {
2. **this**.service.loginCode(**this**.login.value.username).subscribe(res => {
3. **const** usera = res.find((a: any) => {
4. **return** a.username === **this**.login.value.username && a.pass === **this**.login.value.pass
5. **});**
6. **if** (usera) {
7. **if** (**this**.login.value.username == 'admin') {
8. **this**.data = usera;
9. **this**.toastr.success('Login Sebagai Administrator', 'Login Berhasil');
10. **sessionStorage.setItem('currentUser', JSON.stringify(usera));**
11. **this**.login.reset();
12. **this**.router.navigate(['admin/dashboard-admin']);
13. } **else** {
14. **this**.data = usera;
15. **this.toastr.success('Login Sebagai Guru', 'Login Berhasil');**
16. sessionStorage.setItem('currentUser', JSON.stringify(usera));
17. **this**.login.reset();
18. **this**.router.navigate(['guru/dashboard-guru']);
19. }
20. **} else {**
21. **this**.role = **this**.login.value == '';
22. **this**.toastr.warning('Mohon Cek Kembali Username atau Password Anda', 'Login Gagal');
23. **this**.router.navigate(['/']);
24. }
25. **})**
26. }
27. Kode dashboard admin

**Kode program 4.** Kode HTML Dashboard Admin

1. ...
2. <mat-sidenav #drawer class="sidenav" fixedInViewport [attr.role]="(isHandset$ | async) ? 'dialog' : 'navigation'"
3. [mode]="(isHandset$ | async) ? 'over' : 'side'" [opened]="(isHandset$ | async) === false">
4. <mat-toolbar class="toolbar-logo"> <**img** src="assets/images/tut-wuri.png" class="img"></mat-toolbar>
5. **<mat-nav-list>**
6. <**a** mat-list-item routerLink="./dashboard-admin">Dashboard</**a**>
7. <**a** mat-list-item routerLink="./guru">Guru</**a**>
8. <**a** mat-list-item routerLink="./siswa">Siswa</**a**>
9. <**a** mat-list-item routerLink="./kelas">Kelas</**a**>
10. **<a mat-list-item routerLink="./mapel">Mata Pelajaran</a>**
11. <**a** mat-list-item routerLink="./tema">Tema</**a**>
12. </mat-nav-list>
13. </mat-sidenav>
14. ...
15. Kode guru admin

**Kode program 4.** Kode HTML Guru

1. <**table** mat-table [dataSource]="dataSource" class="mat-elevation-z8">
2. ...
3. *<!-- No Column -->*
4. <ng-container matColumnDef="no" >
5. **<th mat-header-cell \*matHeaderCellDef> No </th>**
6. <**td** mat-cell \*matCellDef="let element = index"> {{element + 1}} </**td**>
7. </ng-container>
9. *<!-- NIP Column -->*
10. **<ng-container matColumnDef="nip">**
11. <**th** mat-header-cell \*matHeaderCellDef> NIP </**th**>
12. <**td** mat-cell \*matCellDef="let element"> {{element.nip}} </**td**>
13. </ng-container>
15. ***<!-- Nama Column -->***
16. <ng-container matColumnDef="nama">
17. <**th** mat-header-cell \*matHeaderCellDef> Nama </**th**>
18. <**td** mat-cell \*matCellDef="let element"> {{element.nama}} </**td**>
19. </ng-container>
20. **...**
21. </**table**>

**Kode program 4.** Kode Typescript Guru

1. ...
2. getDataGuru() { *//untuk ambil data*
3. **this**.api.ambilDataGuru()
4. .subscribe(res => {
5. **this.gurus = res;**
6. console.log();
7. **this**.dataSource = **new** MatTableDataSource(**this**.gurus);
8. **this**.dataSource.paginator = **this**.paginator;
9. **this**.dataSource.sort = **this**.sort;
10. **})**
11. }
12. ...
13. Kode tambah guru

**Kode program 4.** Tambah Guru HTML

1. ...
2. <**form** [formGroup]="dataGuruForm" (ngSubmit)="submit()">
3. <**div** mat-dialog-content class="content">
4. <**div**>
5. **<mat-form-field appearance="outline" class="field">**
6. <mat-label>NIP</mat-label>
7. <**input** matInput formControlName="nip" required>
8. <mat-error>Masukkan NIP terlebih dahulu</mat-error>
9. </mat-form-field>
10. **</div>**
11. <**div**>
12. <mat-form-field appearance="outline" class="field">
13. <mat-label>Nama</mat-label>
14. <**input** matInput type="text" formControlName="nama" required>
15. **<mat-error>Masukkan nama terlebih dahulu</mat-error>**
16. </mat-form-field>
17. </**div**>
18. <**div**>
19. <mat-form-field appearance="outline" class="field">
20. **<mat-label>Username</mat-label>**
21. <**input** matInput formControlName="username" required>
22. <mat-error>Masukkan username terlebih dahulu</mat-error>
23. </mat-form-field>
24. </**div**>
25. **...**
26. </**form**>

**Kode program 4.** Tambah Guru Typescript

1. submit() {
2. *// Validasi form*
3. **if** (**this**.dataGuruForm.invalid) {
4. *// Tampilkan pesan kesalahan*
5. **this.toastr.error('Gagal Menambah Data!!!', 'Data Guru');**
6. **return**;
7. }
9. **const** newData = **this**.dataGuruForm.value;
10. **this.api.getGuruNip(newData.nip)**
11. .subscribe(existingData => {
12. **if** (existingData.length > 0) {
13. *// NIP sudah ada, tampilkan pesan kesalahan*
14. **this**.toastr.error('NIP sudah ada. Data tidak dapat ditambahkan.', 'Data Guru');
15. **} else {**
16. *// NIP belum ada, lakukan simpan data*
17. **this**.api.tambahData(newData)
18. .subscribe(res => {
19. **this**.toastr.success('Berhasil Menambah Data!!!', 'Data Guru');
20. **this.dialogref.close();**
21. setTimeout(() => {
22. window.location.reload();
23. }, 5500);
24. });
25. **}**
26. }, error => {
27. console.log(error);
28. });
29. }
30. Kode edit guru

**Kode program 4.** Edit Guru HTML

1. ...
2. <**form** [formGroup]="dataGuruForm" (ngSubmit)="submit()">
3. <**div** mat-dialog-content class="content">
4. <**div**>
5. **<mat-form-field appearance="outline" class="field">**
6. <mat-label>NIP</mat-label>
7. <**input** matInput formControlName="nip" required>
8. <mat-error>Masukkan NIP terlebih dahulu</mat-error>
9. </mat-form-field>
10. **</div>**
11. <**div**>
12. <mat-form-field appearance="outline" class="field">
13. <mat-label>Nama</mat-label>
14. <**input** matInput type="text" formControlName="nama" required>
15. **<mat-error>Masukkan nama terlebih dahulu</mat-error>**
16. </mat-form-field>
17. </**div**>
18. <**div**>
19. <mat-form-field appearance="outline" class="field">
20. **<mat-label>Username</mat-label>**
21. <**input** matInput formControlName="username" required>
22. <mat-error>Masukkan username terlebih dahulu</mat-error>
23. </mat-form-field>
24. </**div**>
25. **...**
26. </**form**>
27. ...

**Kode program 4.** Edit Guru Typescript

1. ...
2. submit() {
3. *// Validasi form*
4. **if** (**this**.dataGuruForm.invalid) {
5. ***// Tampilkan pesan kesalahan***
6. **this**.toastr.error('Gagal Mengupdate Data!!!', 'Data Guru');
7. **return**;
8. }
9. **this**.simpan(); *// Lakukan simpan data*
10. **}**
12. simpan() {
13. **this**.api.ubahDataGuru(**this**.data.id, **this**.dataGuruForm.value).subscribe(res => {
14. **this**.toastr.success('Berhasil Mengupdate Data!!!', 'Data Guru');
15. **this.dialogref.close();**
16. setTimeout(() => {
17. window.location.reload();
18. }, 5500);
19. },
20. **)**
21. }
22. ...
23. Kode hapus guru

**Kode program 4.** Hapus Guru HTML

1. <**div** mat-dialog-content class="content">
2. <**img** [src]="myimage" class="ihapus">
3. <**h1**>Apakah anda yakin ingin menghapus data?</**h1**>
4. </**div**>
5. **<div mat-dialog-actions class="action">**
6. <**button** mat-raised-button color="basic" class="btn-flex" [mat-dialog-close]="true">Batal</**button**>
7. <**button** mat-raised-button color="warn" class="btn-flex" type="submit" (click)="hapusDataGuru()">Hapus</**button**>
8. </**div**>

**Kode program 4.** Hapus Guru Typescript

1. ...
2. hapusDataGuru() {
3. **this**.api.hapusDataGuru(**this**.data.id).subscribe(res => {
4. **this**.toastr.success('Berhasil Menghapus Data!!!', 'Data Guru');
5. **this.dialogref.close();**
6. setTimeout(() => {
7. window.location.reload();
8. }, 5500);
9. })
10. **}**
11. ...
12. Kode siswa

**Kode program 4.** Kode HTML Siswa

1. ...
2. <**table** mat-table [dataSource]="dataSiswa" class="mat-elevation-z8">
3. ...
4. *<!-- No Column -->*
5. **<ng-container matColumnDef="no" >**
6. <**th** mat-header-cell \*matHeaderCellDef> No </**th**>
7. <**td** mat-cell \*matCellDef="let element = index"> {{element + 1}} </**td**>
8. </ng-container>
10. ***<!-- NIS Column -->***
11. <ng-container matColumnDef="nis">
12. <**th** mat-header-cell \*matHeaderCellDef> NIS </**th**>
13. <**td** mat-cell \*matCellDef="let element"> {{element.nis}} </**td**>
14. </ng-container>
16. *<!-- Nama Column -->*
17. <ng-container matColumnDef="nama">
18. <**th** mat-header-cell \*matHeaderCellDef> Nama </**th**>
19. <**td** mat-cell \*matCellDef="let element"> {{element.nama}} </**td**>
20. **</ng-container>**
21. ...
22. </**table**>
23. ...

**Kode program 4.** Kode Typescript Siswa

1. ...
2. getDataSiswa() { *//untuk ambil data bisa dibuat kek gini*
3. **this**.api.ambilDataSiswa()
4. .subscribe(res => {
5. **this.siswas = res;**
6. console.log();
7. **this**.dataSiswa = **new** MatTableDataSource(**this**.siswas);
8. **this**.dataSiswa.paginator = **this**.paginator;
9. **this**.dataSiswa.sort = **this**.sort;
10. **})**
11. }
12. ...
13. Kode tambah siswa

**Kode program 4.** Kode HTML Tambah Siswa

1. ...
2. <**form** [formGroup]="dataSiswaForm" (ngSubmit)="submit()">
3. <**div** mat-dialog-content class="content">
4. <**div**>
5. **<mat-form-field appearance="outline" class="field">**
6. <mat-label>NIS</mat-label>
7. <**input** matInput formControlName="nis" required>
8. <mat-error>Masukkan NIS terlebih dahulu</mat-error>
9. </mat-form-field>
10. **</div>**
11. <**div**>
12. <mat-form-field appearance="outline" class="field">
13. <mat-label>Nama</mat-label>
14. <**input** matInput type="text" formControlName="nama" required>
15. **<mat-error>Masukkan Nama terlebih dahulu</mat-error>**
16. </mat-form-field>
17. </**div**>
18. <**div**>
19. <mat-form-field appearance="outline" class="field">
20. **<mat-label>Alamat</mat-label>**
21. <**input** matInput formControlName="alamat" required>
22. <mat-error>Masukkan Alamat terlebih dahulu</mat-error>
23. </mat-form-field>
24. </**div**>
25. **...**
26. </**form**>

**Kode program 4.** Kode Typescipt Tambah Siswa

1. ...
2. submit() {
3. *// Validasi form*
4. **if** (**this**.dataSiswaForm.invalid) {
5. ***// Tampilkan pesan kesalahan***
6. **this**.toastr.error('Gagal Menambah Data!!!', 'Data Siswa');
7. **return**;
8. }
10. **const newData = this.dataSiswaForm.value;**
11. **this**.api.getSiswaNis(newData.nis)
12. .subscribe(existingData => {
13. **if** (existingData.length > 0) {
14. *// NIS sudah ada, tampilkan pesan kesalahan*
15. **this.toastr.error('NIS sudah ada. Data tidak dapat ditambahkan.', 'Data Siswa');**
16. } **else** {
17. *// NIS belum ada, lakukan simpan data*
18. **this**.api.tambahDataSiswa(newData)
19. .subscribe(res => {
20. **this.toastr.success('Berhasil Menambah Data!!!', 'Data Siswa');**
21. **this**.dialogref.close();
22. setTimeout(() => {
23. window.location.reload();
24. }, 5500);
25. **});**
26. }
27. }, error => {
28. console.log(error);
29. });
30. **}**
31. ...
32. kode edit siswa

**Kode program 4.** Kode HTML Edit Siswa

1. ...
2. <**form** [formGroup]="dataSiswaForm" (ngSubmit)="submit()">
3. <**div** mat-dialog-content class="content">
4. <**div**>
5. **<mat-form-field appearance="outline" class="field">**
6. <mat-label>NIS</mat-label>
7. <**input** matInput formControlName="nis" required>
8. <mat-error>Masukkan NIP terlebih dahulu</mat-error>
9. </mat-form-field>
10. **</div>**
11. <**div**>
12. <mat-form-field appearance="outline" class="field">
13. <mat-label>Nama</mat-label>
14. <**input** matInput type="text" formControlName="nama" required>
15. **<mat-error>Masukkan nama terlebih dahulu</mat-error>**
16. </mat-form-field>
17. </**div**>
18. <**div**>
19. <mat-form-field appearance="outline" class="field">
20. **<mat-label>Alamat</mat-label>**
21. <**input** matInput formControlName="alamat" required>
22. <mat-error>Masukkan username terlebih dahulu</mat-error>
23. </mat-form-field>
24. </**div**>
25. **...**
26. </**form**>

**Kode program 4.** Kode Typrscript Edit Siswa

1. ...
2. submit() {
3. *// Validasi form*
4. **if** (**this**.dataSiswaForm.invalid) {
5. ***// Tampilkan pesan kesalahan***
6. **this**.toastr.error('Gagal Mengupdate Data!!!', 'Data Siswa');
7. **return**;
8. }
9. **this**.simpan(); *// Lakukan simpan data*
10. **}**
12. simpan() {
13. **this**.api.ubahDataSiswa(**this**.data.id, **this**.dataSiswaForm.value).subscribe(res => {
14. **this.toastr.success('Berhasil Mengupdate Data!!!', 'Data Siswa');**
15. **this**.dialogref.close();
16. setTimeout(() => {
17. window.location.reload();
18. }, 5500);
19. **},**
20. )
21. }
22. ...
23. kode hapus siswa

**Kode program 4.** Kode HTML Hapus Siswa

1. ...
2. <**div** mat-dialog-content class="content">
3. <**img** [src]="myimage" class="ihapus">
4. <**h1**>Apakah anda yakin ingin menghapus data?</**h1**>
5. **</div>**
6. <**div** mat-dialog-actions class="action">
7. <**button** mat-raised-button color="basic" class="btn-flex" [mat-dialog-close]="true">Batal</**button**>
8. <**button** mat-raised-button color="warn" class="btn-flex" type="submit" (click)="hapusDataSiswa()">Hapus</**button**>
9. </**div**>
10. **...**

**Kode program 4.** Kode Typescript Hapus Siswa

1. ...
2. hapusDataSiswa() {
3. **this**.api.hapusDataSiswa(**this**.data.id).subscribe(res => {
4. **this**.toastr.success('Berhasil Menghapus Data!!!', 'Data Siswa');
5. **this.dialogref.close();**
6. setTimeout(() => {
7. window.location.reload();
8. }, 5500);
9. })
10. **}**
11. ...
12. Kode Kelas

**Kode program 4.** Kode HTML Kelas

1. ...
2. <**table** mat-table [dataSource]="dataKelas" class="mat-elevation-z8">
4. *<!-- No Column -->*
5. **<ng-container matColumnDef="no" >**
6. <**th** mat-header-cell \*matHeaderCellDef> No </**th**>
7. <**td** mat-cell \*matCellDef="let element = index"> {{element + 1}} </**td**>
8. </ng-container>
10. ***<!-- Kelas Column -->***
11. <ng-container matColumnDef="kelas">
12. <**th** mat-header-cell \*matHeaderCellDef> Kelas </**th**>
13. <**td** mat-cell \*matCellDef="let element"> {{element.kelas}} </**td**>
14. </ng-container>
16. *<!-- Jumlah Siswa Column -->*
17. <ng-container matColumnDef="jumlah\_siswa">
18. <**th** mat-header-cell \*matHeaderCellDef> Jumlah Siswa </**th**>
19. <**td** mat-cell \*matCellDef="let element"> {{element.jumlah\_siswa}} </**td**>
20. **</ng-container>**
21. ...
22. </**table**>
23. ...

**Kode program 4.** Kode Typescipt Kelas

1. ...
2. getDataKelas() { *//untuk ambil data*
3. **this**.api.ambilDataKelas()
4. .subscribe(res => {
5. **this.kelass = res;**
6. console.log();
7. **this**.dataKelas = **new** MatTableDataSource(**this**.kelass);
8. **this**.dataKelas.paginator = **this**.paginator;
9. **this**.dataKelas.sort = **this**.sort;
10. **})**
11. }
12. ...
13. Kode tambah kelas

**Kode program 4.** Kode HTML Tambah Kelas

1. ...
2. <**form** [formGroup]="dataKelasForm" (ngSubmit)="submit()">
3. <**div** mat-dialog-content class="content">
4. <**div**>
5. **<mat-form-field appearance="outline" class="field">**
6. <mat-label>Kelas</mat-label>
7. <mat-select formControlName="kelas" required>
8. <mat-option value="1">1</mat-option>
9. <mat-option value="2">2</mat-option>
10. **<mat-option value="3">3</mat-option>**
11. <mat-option value="4">4</mat-option>
12. <mat-option value="5">5</mat-option>
13. <mat-option value="6">6</mat-option>
14. </mat-select>
15. **<mat-error>Masukkan Kelas terlebih dahulu</mat-error>**
16. </mat-form-field>
17. </**div**>
18. <**div**>
19. <mat-form-field appearance="outline" class="field">
20. **<mat-label>Jumlah Siswa</mat-label>**
21. <**input** matInput type="text" formControlName="jumlah\_siswa" required>
22. <mat-error>Masukkan Jumlah Siswa terlebih dahulu</mat-error>
23. </mat-form-field>
24. </**div**>
25. **...**
26. </**form**>

**Kode program 4.** Kode Typescript Tambah Kelas

1. ...
2. submit() {
3. *// Validasi form*
4. **if** (**this**.dataKelasForm.invalid) {
5. ***// Tampilkan pesan kesalahan***
6. **this**.toastr.error('Gagal Menambah Data!!!', 'Data Kelas');
7. **return**;
8. }
10. **const newData = this.dataKelasForm.value;**
11. **this**.api.getKelasId(newData.kelas)
12. .subscribe(existingData => {
13. **if** (existingData.length > 0) {
14. *// kelas sudah ada, tampilkan pesan kesalahan*
15. **this.toastr.error('Kelas sudah ada. Data tidak dapat ditambahkan.', 'Data Kelas');**
16. } **else** {
17. *// kelas belum ada, lakukan simpan data*
18. **this**.api.tambahDataKelas(newData)
19. .subscribe(res => {
20. **this.toastr.success('Berhasil Menambah Data!!!', 'Data Kelas');**
21. **this**.dialogref.close();
22. setTimeout(() => {
23. window.location.reload();
24. }, 5500);
25. **});**
26. }
27. }, error => {
28. console.log(error);
29. });
30. **}**
31. ...
32. Kode edit kelas

**Kode program 4.** Kode HTML Edit Kelas

1. ...
2. <**form** [formGroup]="dataKelasForm" (ngSubmit)="submit()">
3. <**div** mat-dialog-content class="content">
4. <**div**>
5. **<mat-form-field appearance="outline" class="field">**
6. <mat-label>Kelas</mat-label>
7. <**input** matInput formControlName="kelas" required>
8. <mat-error>Masukkan NIP terlebih dahulu</mat-error>
9. </mat-form-field>
10. **</div>**
11. <**div**>
12. <mat-form-field appearance="outline" class="field">
13. <mat-label>Jumlah Siswa</mat-label>
14. <**input** matInput type="text" formControlName="jumlah\_siswa" required>
15. **<mat-error>Masukkan nama terlebih dahulu</mat-error>**
16. </mat-form-field>
17. </**div**>
18. ...
19. </**form**>

**Kode program 4.** Kode Typescript Edit Kelas

1. ...
2. submit() {
3. *// Validasi form*
4. **if** (**this**.dataKelasForm.invalid) {
5. ***// Tampilkan pesan kesalahan***
6. **this**.toastr.error('Gagal Menambah Data!!!', 'Data Kelas');
7. **return**;
8. }
9. **this**.simpan(); *// Lakukan simpan data*
10. **}**
12. simpan() {
13. **this**.api.ubahDataKelas(**this**.data.id, **this**.dataKelasForm.value).subscribe(res => {
14. **this**.toastr.success('Berhasil Mengupdate Data!!!', 'Data Kelas');
15. **this.dialogref.close();**
16. setTimeout(() => {
17. window.location.reload();
18. }, 5500);
19. },
20. **)**
21. }
22. ...
23. Kode hapus kelas

**Kode program 4.** Hapus HTML Kelas

1. <**div** mat-dialog-content class="content">
2. <**img** [src]="myimage" class="ihapus">
3. <**h1**>Apakah anda yakin ingin menghapus data?</**h1**>
4. </**div**>
5. **<div mat-dialog-actions class="action">**
6. <**button** mat-raised-button color="basic" class="btn-flex" [mat-dialog-close]="true">Batal</**button**>
7. <**button** mat-raised-button color="warn" class="btn-flex" type="submit" (click)="hapusDataKelas()">Hapus</**button**>
8. </**div**>

**Kode program 4.** Kode Typescript Kelas

1. ...
2. hapusDataKelas() {
3. **this**.api.hapusDataKelas(**this**.data.id).subscribe(res => {
4. **this**.toastr.success('Berhasil Menghapus Data!!!', 'Data Kelas');
5. **this.dialogref.close();**
6. setTimeout(() => {
7. window.location.reload();
8. }, 5500);
9. })
10. **}**
11. ...
12. Kode Mata Pelajaran

**Kode program 4.** Kode HTML Mata Pelajaran

1. ...
2. <**table** mat-table [dataSource]="dataMapel" class="mat-elevation-z8">
4. <ng-container matColumnDef="no" >
5. **<th mat-header-cell \*matHeaderCellDef> No </th>**
6. <**td** mat-cell \*matCellDef="let element = index"> {{element + 1}} </**td**>
7. </ng-container>
9. <ng-container matColumnDef="mapel">
10. **<th mat-header-cell \*matHeaderCellDef> Mata Pelajaran </th>**
11. <**td** mat-cell \*matCellDef="let element"> {{element.mapel}} </**td**>
12. </ng-container>
13. ...
14. </**table**>
15. **...**

**Kode program 4.** Kode Typescript Mata Pelajaran

1. ...
2. getDataMapel() { *//untuk ambil data bisa dibuat kek gini*
3. **this**.api.ambilDataMapel()
4. .subscribe(res => {
5. **this.mapels = res;**
6. console.log();
7. **this**.dataMapel = **new** MatTableDataSource(**this**.mapels);
8. **this**.dataMapel.paginator = **this**.paginator;
9. **this**.dataMapel.sort = **this**.sort;
11. })
12. }
13. ...
14. Kode Tambah Mata Pelajaran

**Kode program 4.** Kode HTML Tambah Mata Pelajaran

1. ...
2. <**form** [formGroup]="dataMapelForm" (ngSubmit)="submit()">
3. <**div** mat-dialog-content class="content">
4. <**div**>
5. **<mat-form-field appearance="outline" class="field">**
6. <mat-label>Mata Pelajaran</mat-label>
7. <**input** matInput formControlName="mapel" required>
8. <mat-error>Masukkan Mata Pelajaran terlebih dahulu</mat-error>
9. </mat-form-field>
10. **</div>**
11. ...
12. </**form**>

**Kode program 4.** Kode Typescript Tambah Mata Pelajaran

1. ...
2. submit() {
3. *// Validasi form*
4. **if** (**this**.dataMapelForm.invalid) {
5. ***// Tampilkan pesan kesalahan***
6. **this**.toastr.error('Gagal Menambah Data!!!', 'Data Mata Pelajaran');
7. **return**;
8. }
10. **const newData = this.dataMapelForm.value;**
11. **const** mapelInput = newData.mapel.toUpperCase(); *// Ubah input menjadi lowercase*
12. **this**.api.getMapelId(mapelInput)
13. .subscribe(existingData => {
14. **if** (existingData.length > 0) {
15. ***// Mata Pelajaran sudah ada, tampilkan pesan kesalahan***
16. **this**.toastr.error('Mata Pelajaran sudah ada. Data tidak dapat ditambahkan.', 'Data Mata Pelajaran');
17. } **else** {
18. *// Mata Pelajaran belum ada, lakukan simpan data*
19. **this**.api.tambahDataMapel(newData)
20. **.subscribe(res => {**
21. **this**.toastr.success('Berhasil Menambah Data!!!', 'Data Mata Pelajaran');
22. **this**.dialogref.close();
23. setTimeout(() => {
24. window.location.reload();
25. **}, 5500);**
26. });
27. }
28. }, error => {
29. console.log(error);
30. **});**
31. }
32. ...
33. Kode Edit Mata Pelajaran

**Kode program 4.** Kode HTML Edit Mata Pelajaran

1. ...
2. <**form** [formGroup]="dataMapelForm" (ngSubmit)="submit()">
3. <**div** mat-dialog-content class="content">
4. <**div**>
5. **<mat-form-field appearance="outline" class="field">**
6. <mat-label>Mata Pelajaran</mat-label>
7. <**input** matInput formControlName="mapel" required>
8. <mat-error>Masukkan Mata Palajaran terlebih dahulu</mat-error>
9. </mat-form-field>
10. **</div>**
11. ...
12. </**form**>

**Kode program 4.** Kode Typescript Edit Mata Pelajaran

1. ...
2. submit() {
3. *// Validasi form*
4. **if** (**this**.dataMapelForm.invalid) {
5. ***// Tampilkan pesan kesalahan***
6. **this**.toastr.error('Gagal Mengupdate Data!!!', 'Data Data Mata Pelajaran');
7. **return**;
8. }
9. **this**.simpan(); *// Lakukan simpan data*
10. **}**
12. simpan() {
13. **this**.api.ubahDataMapel(**this**.data.id, **this**.dataMapelForm.value).subscribe(res => {
14. **this**.toastr.success('Berhasil Mengupdate Data!!!', 'Data Data Mata Pelajaran');
15. **this.dialogref.close();**
16. setTimeout(() => {
17. window.location.reload();
18. }, 5500);
19. },
20. **)**
21. }
22. ...
23. Kode Hapus Mata Pelajaran

**Kode program 4.** Kode HTML Hapus Mata Pelajaran

1. <**div** mat-dialog-content class="content">
2. <**img** [src]="myimage" class="ihapus">
3. <**h1**>Apakah anda yakin ingin menghapus data?</**h1**>
4. </**div**>
5. **<div mat-dialog-actions class="action">**
6. <**button** mat-raised-button color="basic" class="btn-flex" [mat-dialog-close]="true">Batal</**button**>
7. <**button** mat-raised-button color="warn" class="btn-flex" type="submit" (click)="hapusDataMapel()">Hapus</**button**>
8. </**div**>

**Kode program 4.** Kode Typescript Hapus Mata Pelajaran

1. ...
2. hapusDataMapel() {
3. **this**.api.hapusDataMapel(**this**.data.id).subscribe(res => {
4. **this**.toastr.success('Berhasil Menghapus Data!!!', 'Data Mata Pelajaran');
5. **this.dialogref.close();**
6. setTimeout(() => {
7. window.location.reload();
8. }, 5500);
9. })
10. **}**
11. ...
12. Kode Tema

**Kode program 4.** Kode HTML Tema

1. ...
2. <**table** mat-table [dataSource]="dataTema" class="mat-elevation-z8">
4. <ng-container matColumnDef="no" >
5. **<th mat-header-cell \*matHeaderCellDef> No </th>**
6. <**td** mat-cell \*matCellDef="let element = index"> {{element + 1}} </**td**>
7. </ng-container>
9. <ng-container matColumnDef="mapel">
10. **<th mat-header-cell \*matHeaderCellDef> Mata Pelajaran </th>**
11. <**td** mat-cell \*matCellDef="let element"> {{element.mapel}} </**td**>
12. </ng-container>
14. <ng-container matColumnDef="tema">
15. **<th mat-header-cell \*matHeaderCellDef> Tema </th>**
16. <**td** mat-cell \*matCellDef="let element"> {{element.tema}} </**td**>
17. </ng-container>
18. ...
19. </**table**>
20. **...**

**Kode program 4.** Kode Typescript Tema

1. ...
2. getDataTema() { *//untuk ambil data*
3. **this**.api.ambilDataTema()
4. .subscribe(res => {
5. **this.temas = res;**
6. console.log();
7. **this**.dataTema = **new** MatTableDataSource(**this**.temas);
8. **this**.dataTema.paginator = **this**.paginator;
9. **this**.dataTema.sort = **this**.sort;
10. **})**
11. }
12. ...
13. Kode Tambah Tema

**Kode program 4.** Kode HTML Tambah Tema

1. ...
2. <**form** [formGroup]="dataTemaForm" (ngSubmit)="submit()">
3. <**div** mat-dialog-content class="content">
4. <**div**>
5. **<mat-form-field appearance="outline" class="field">**
6. <mat-label>Mata Pelajaran</mat-label>
7. <mat-select formControlName="mapel" required >
8. <mat-option \*ngFor="let um of unikMapel" [value]="um">{{um}}</mat-option>
9. </mat-select>
10. **<mat-error>Masukkan Mata Pelajaran terlebih dahulu</mat-error>**
11. </mat-form-field>
12. </**div**>
13. ...
14. </**form**>

**Kode program 4.** Kode Typescript Tambah Tema

1. ...
2. submit() {
3. *// Validasi form*
4. **if** (**this**.dataTemaForm.invalid) {
5. ***// Tampilkan pesan kesalahan***
6. **this**.toastr.error('Gagal Menambah Data!!!', 'Data Tema');
7. **return**;
8. }
10. **const newData = this.dataTemaForm.value;**
11. **const** temaInput = newData.tema;
12. **const** subTemaInput = newData.sub\_tema;
13. **this**.api.getTemaId(temaInput)
14. .subscribe(existingData => {
15. **const isExisting = existingData.some(data => {**
16. **const** temaData = data.tema;
17. **const** subTemaData = data.sub\_tema;
18. **return** encodeURIComponent(temaData) === encodeURIComponent(temaInput) && encodeURIComponent(subTemaData) === encodeURIComponent(subTemaInput);
19. });
20. **if (isExisting) {**
21. *// Mata Pelajaran dengan Tema yang sama sudah ada, tampilkan pesan kesalahan*
22. **this**.toastr.error('Tema dengan Sub Tema yang sama sudah ada. Data tidak dapat ditambahkan.', 'Data Tema);
23. } **else** {
24. *// Mata Pelajaran dengan Tema yang sama belum ada, lakukan simpan data*
25. **this.api.tambahDataTema(newData)**
26. .subscribe(res => {
27. **this**.toastr.success('Berhasil Menambah Data!!!', 'Data Tema’);
28. **this**.dialogref.close();
29. setTimeout(() => {
30. **window.location.reload();**
31. }, 5500);
32. });
33. }
34. }, error => {
35. **console.log(error);**
36. });
37. }
38. ...
39. Kode Edit Tema

**Kode program 4.** Kode HTML Edit Tema

1. ...
2. <**form** [formGroup]="dataTemaForm">
3. ...
4. <**div**>
5. **<mat-form-field appearance="outline" class="field">**
6. <mat-label>Tema</mat-label>
7. <**input** matInput formControlName="tema" [matAutocomplete]="autoTema" required ngDefaultControl>
8. <mat-autocomplete #autoTema="matAutocomplete">
9. <mat-option \*ngFor="let ut of unikTema" [value]="ut">
10. **{{ut}}**
11. </mat-option>
12. </mat-autocomplete>
13. <mat-error>Masukkan Jabatan terlebih dahulu</mat-error>
14. </mat-form-field>
15. **</div>**
16. ...
17. </**form**>

**Kode program 4.** Kode Typescript Edit Tema

1. ...
2. getTema() {
3. **this**.api.ambilDataTema().subscribe(res => {
4. **this**.tema = res;
5. **this.filterTema();**
6. })
7. }
9. simpan() {
10. **this.api.ubahDataTema(this.data.id, this.dataTemaForm.value).subscribe(res => {**
11. **this**.toastr.success('Berhasil Mengupdate Data!!!', 'Data Tema');
12. **this**.dialogref.close();
13. setTimeout(() => {
14. window.location.reload();
15. **}, 5500);**
16. },
17. )
18. }
19. ...
20. Kode Hapus Tema

**Kode program 4.** Kode HTML Hapus Tema

1. <**div** mat-dialog-content class="content">
2. <**img** [src]="myimage" class="ihapus">
3. <**h1**>Apakah anda yakin ingin menghapus data?</**h1**>
4. </**div**>
5. **<div mat-dialog-actions class="action">**
6. <**button** mat-raised-button color="basic" class="btn-flex" [mat-dialog-close]="true">Batal</**button**>
7. <**button** mat-raised-button color="warn" class="btn-flex" type="submit" (click)="hapusDataMapel()">Hapus</**button**>
8. </**div**>

**Kode program 4.** Kode Typescript Hapus Tema

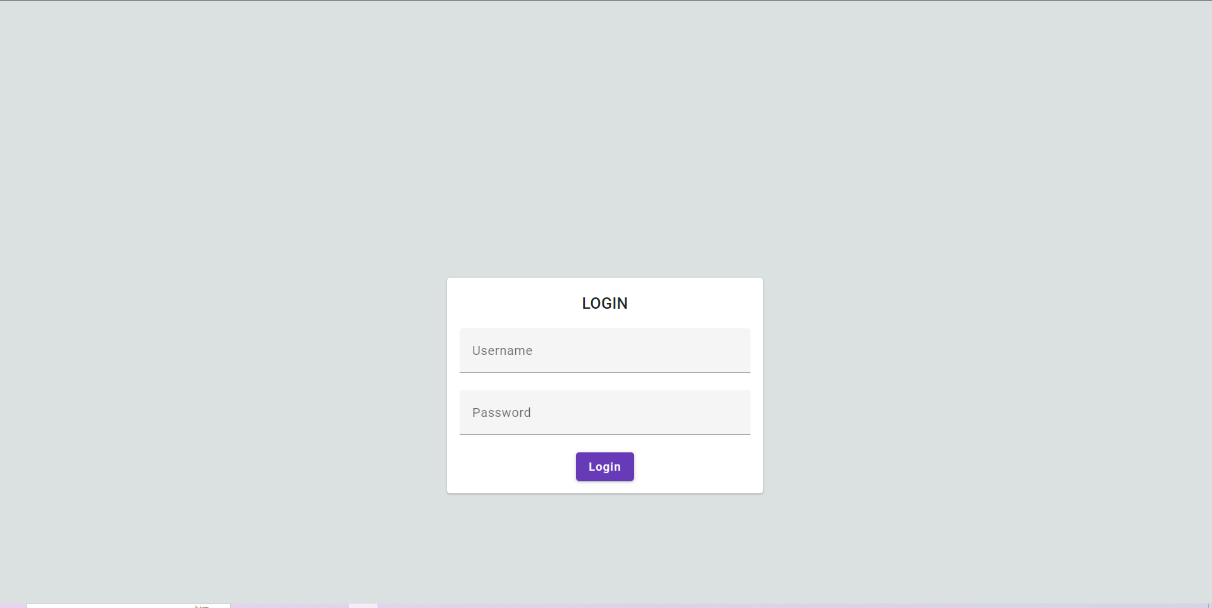
1. ...
2. hapusDataTema() {
3. **this**.api.hapusDataTema(**this**.data.id).subscribe(res => {
4. **this**.toastr.success('Berhasil Menghapus Data!!!', 'Data Tema');
5. **this.dialogref.close();**
6. setTimeout(() => {
7. window.location.reload();
8. }, 5500);
9. })
10. **}**
11. ...

## **4.10 Hasil Implementasi**

Berikut ini merupakan hasil implementasi dari perancangan wireframe dan kode program yang telah diranca ng seelumnya. Hasil implementasi dari perancangan tersebut dapat dilihat dengan menjalankan project angular menggunakan web browser. Berikut adalah hasilnya :

### **4.9.1 Guru**

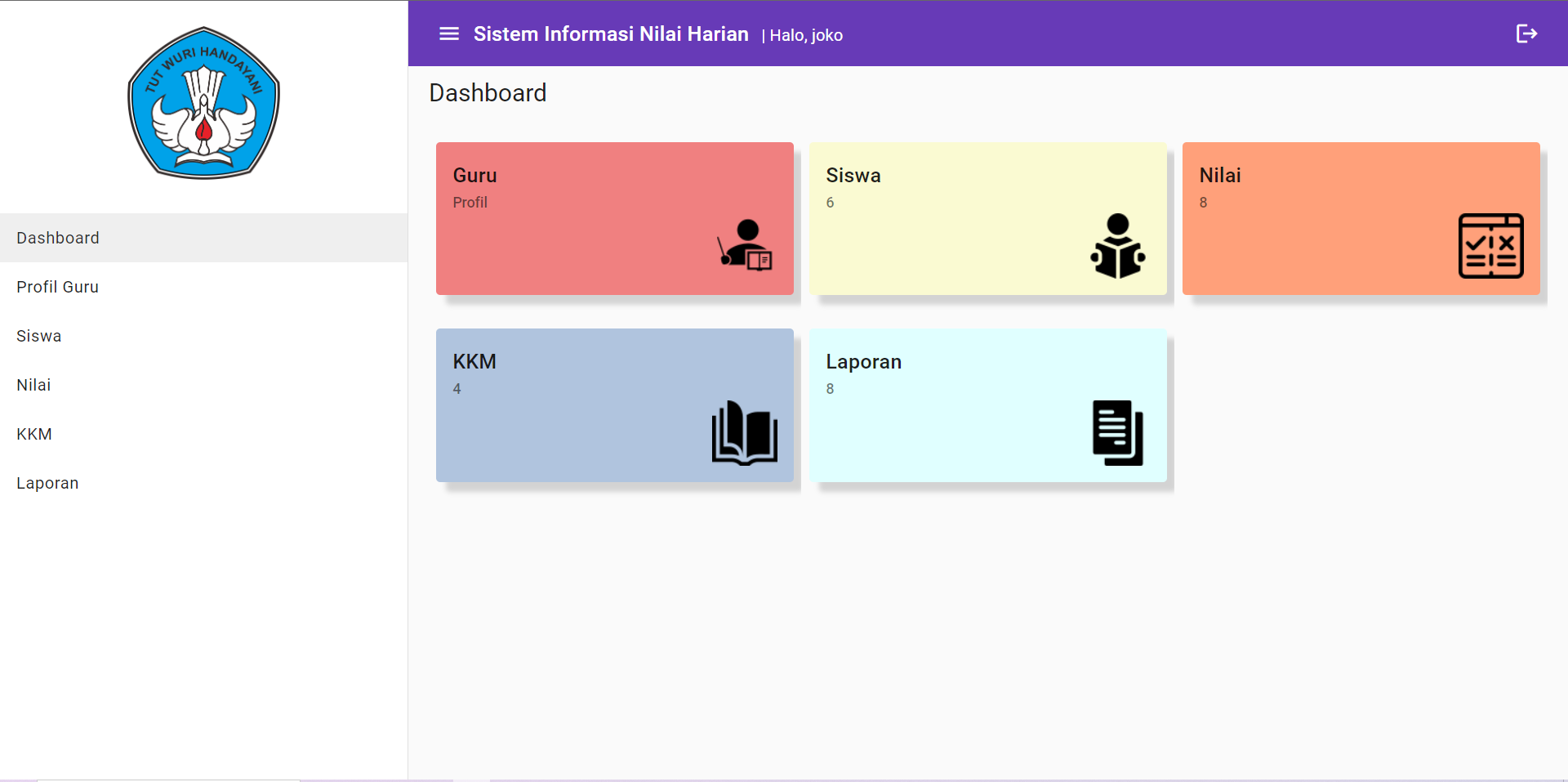
1. Halaman login



**Gambar 4.** Halaman Login

Dari **gambar 4.** Diatas merupakan halaman login yang berisi field username dan password, serta tombol login.

1. Halaman Dashboard Guru



**Gambar 4.** Halaman Dashboard Guru

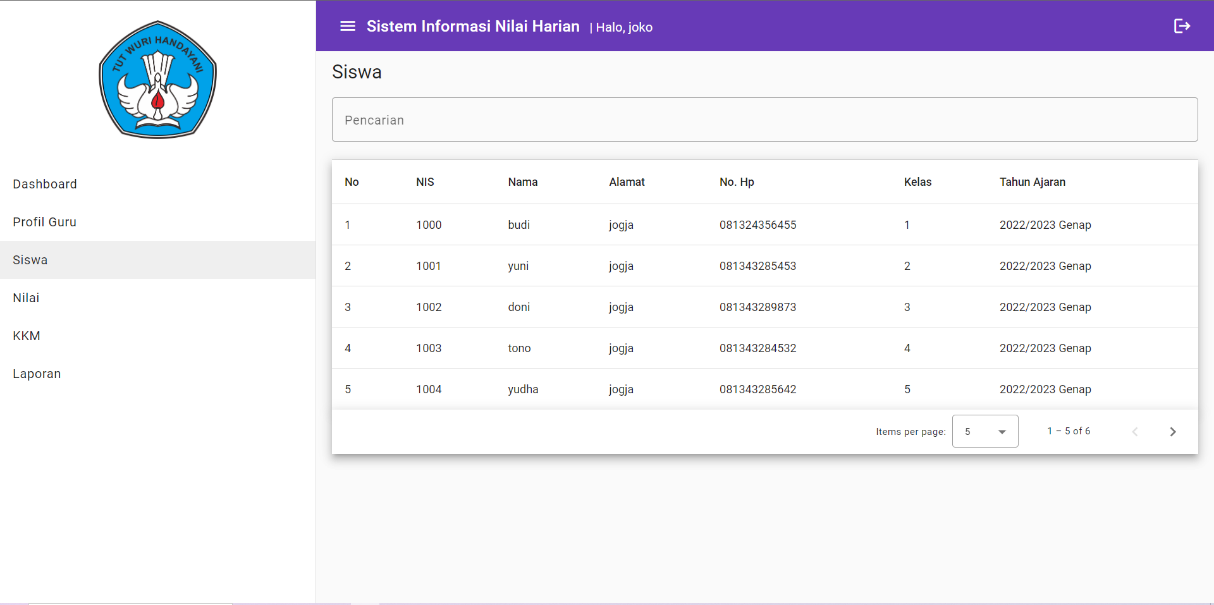
Dari **gambar 4.** Diatas merupakan halaman dashboard dari guru, dimana terdapat beberapa card yang berisi jumlah data dalam system yang terapat pada menu di sidebar.

1. Halaman Profil Guru

**Gambar 4.** Halaman Profil Guru

Dari **gambar 4.** Diatas merupakan halaman yang menampilkan data profil dari guru yang sedang menggunakan aplikasi.

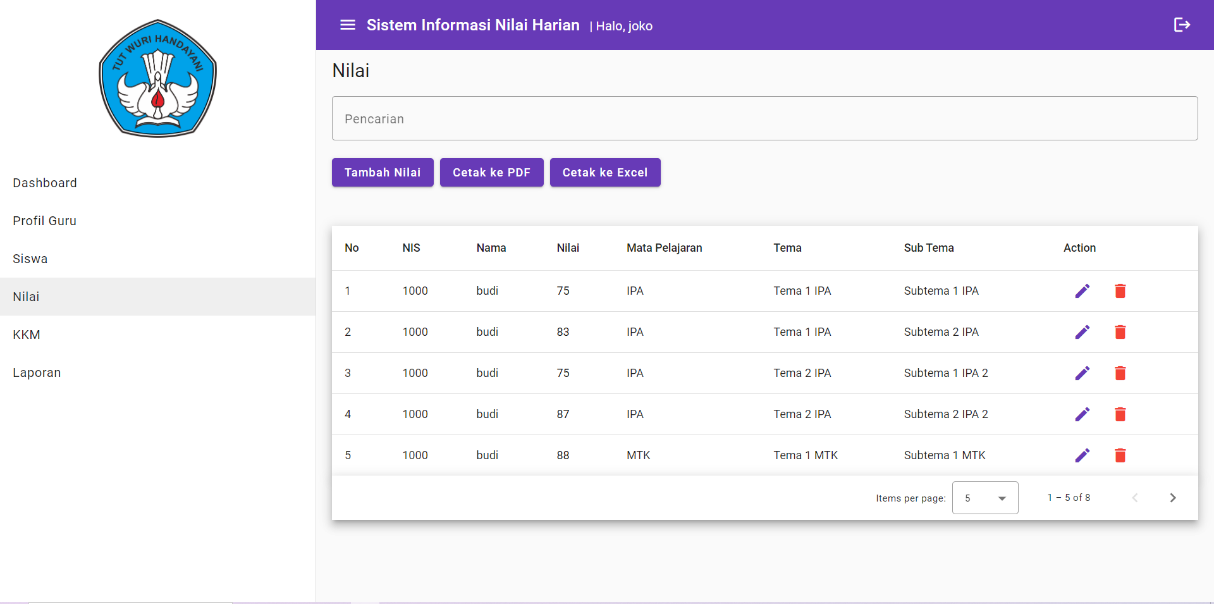
1. Halaman siswa



**Gambar 4.** Halaman Siswa

Dari **gambar 4.** Diatas merupakan halaman siswa yang berisi pencarian dan tabel siswa.

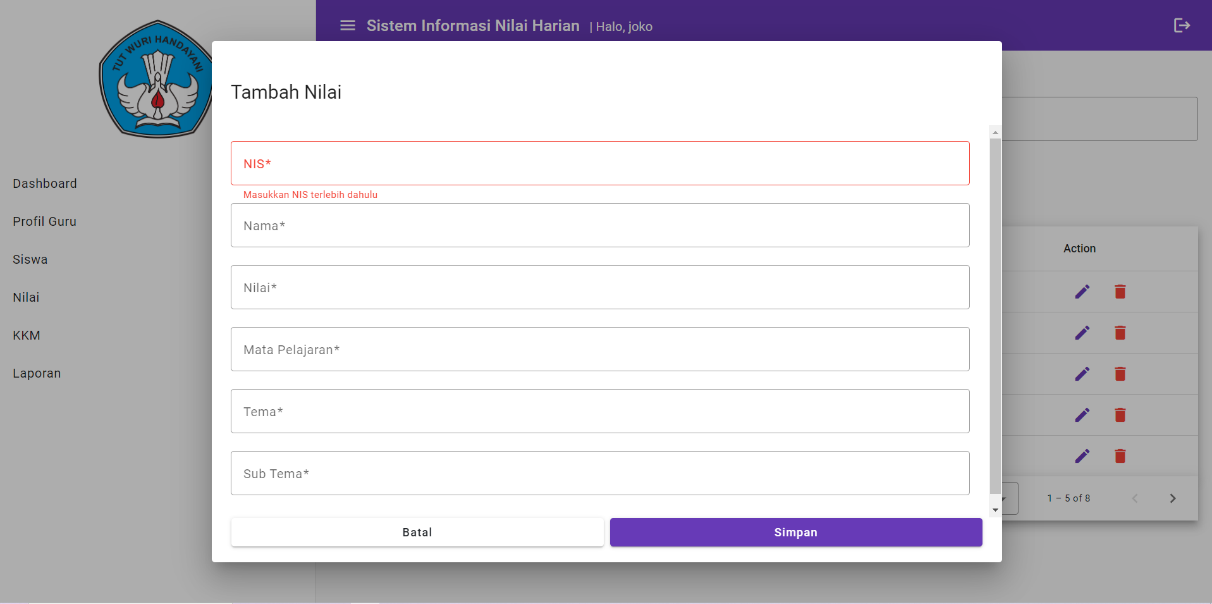
1. Halaman nilai



**Gambar 4.** Halaman Nilai

Dari **gambar 4.** Diatas merupakan halaman nilai yang berisi pencarian, tabel nilai, tambah nilai, edit nilai, hapus nilai, cetak pdf, dan cetak excel.

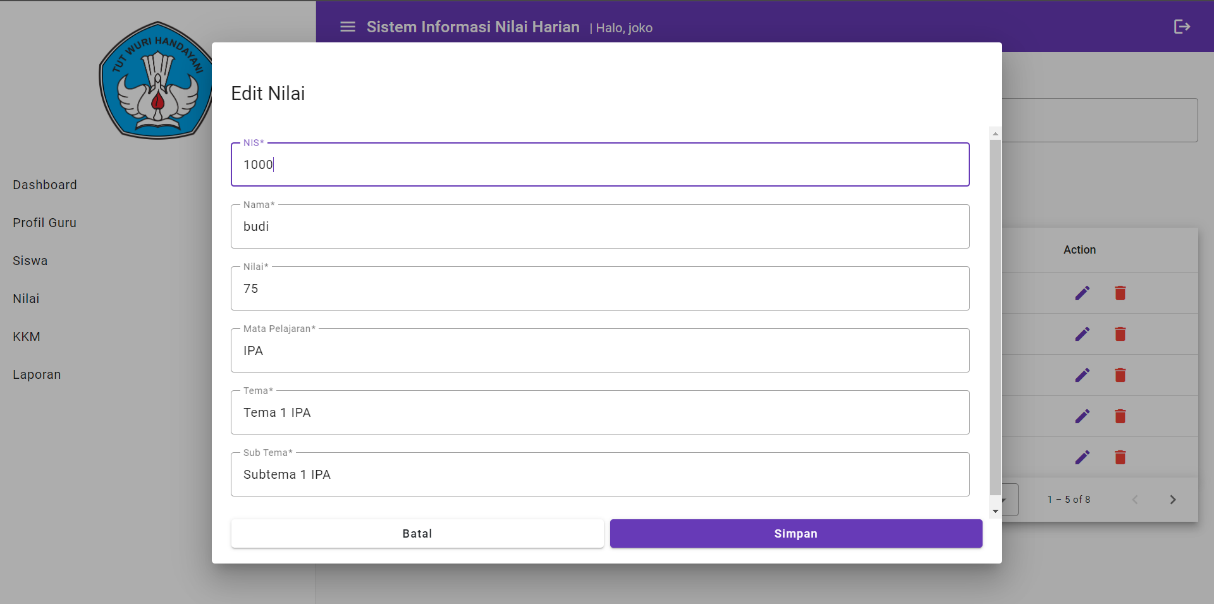
1. Halaman tambah nilai



**Gambar 4.** Halaman Tambah Nilai

Dari **gambar 4.** Diatas merupakan halaman tambah nilai, yang berisikan beberapa field yang perlu diisi dan memiliki indicator jika field belum diisi. Terdapat pilihan batal dan simpan.

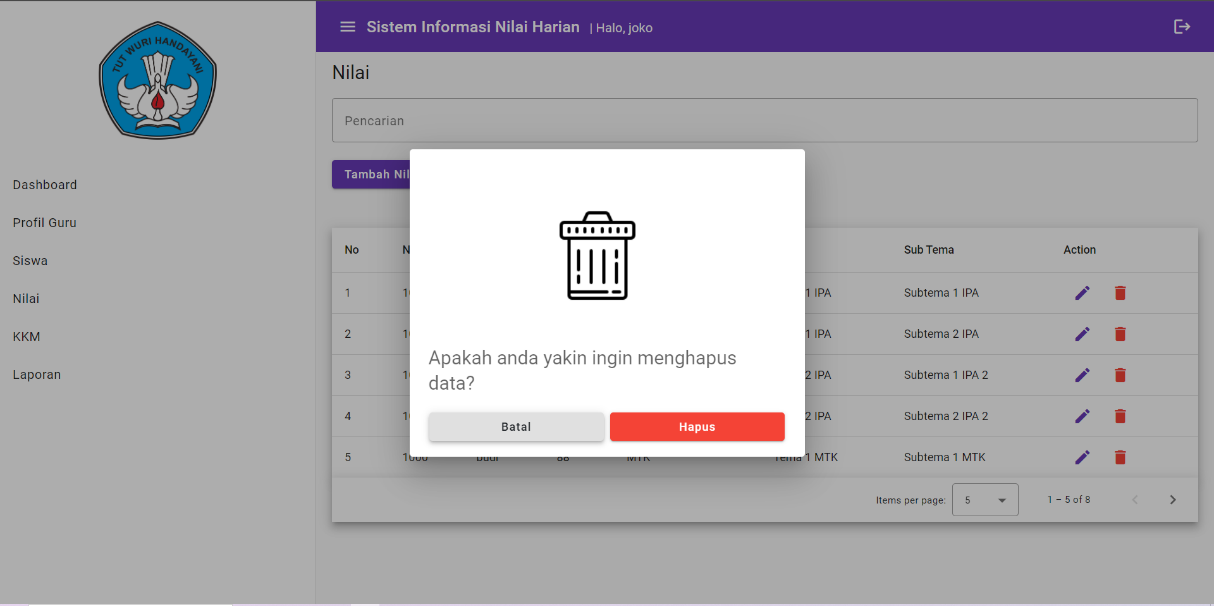
1. Halaman Edit Nilai



**Gambar 4.** Halaman Edit Nilai

Dari **gambar 4.** Diatas merupakan halaman edit nilai, data pada field sudah terisi sesuai dengan data yang kita pilih. Terdapat pilihan batal dan simpan.

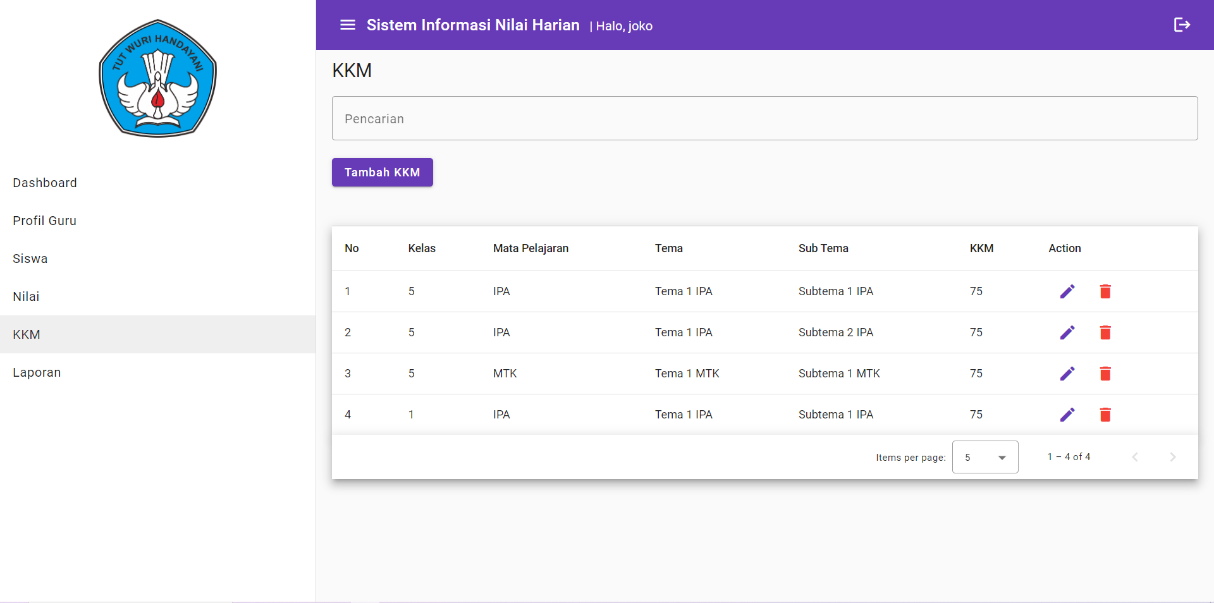
1. Halaman Hapus Nilai



**Gambar 4.** Halaman Hapus Nilai

Dari **gambar 4.** Diatas merupakan halaman yang muncul ketika kita ingin menghapus data nilai, terdapat pilihan batal dan hapus.

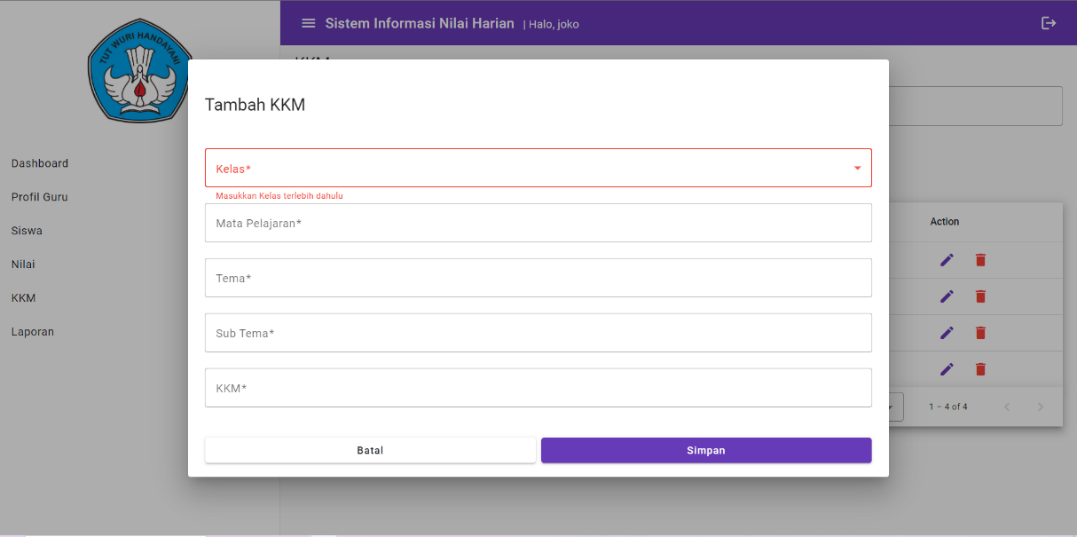
1. Halaman KKM



**Gambar 4.** Halaman KKM

Dari **gambar 4.** Diatas merupakan halaman kkm yang berisikan pencarian, tabel data kkm, tambah kkm, edit kkm, dan hapus kkm

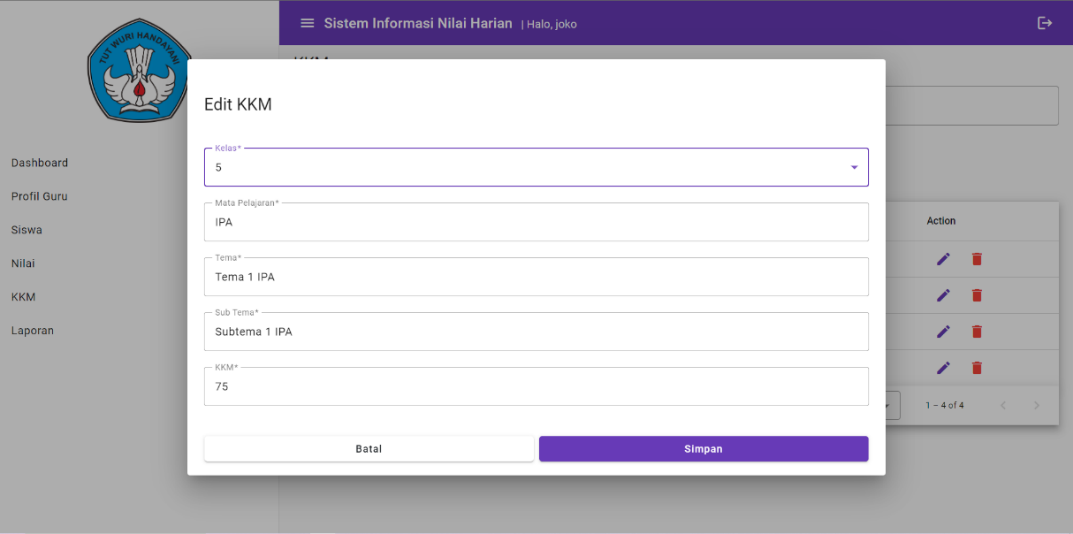
1. Halaman Tambah Kkm



**Gambar 4.** Halaman Tambah KKM

Dari **gambar 4.** Diatas merupakan halaman tambah kkn, yang berisikan beberapa field yang perlu diisi dan memiliki indicator jika field belum diisi. Terdapat pilihan batal dan simpan.

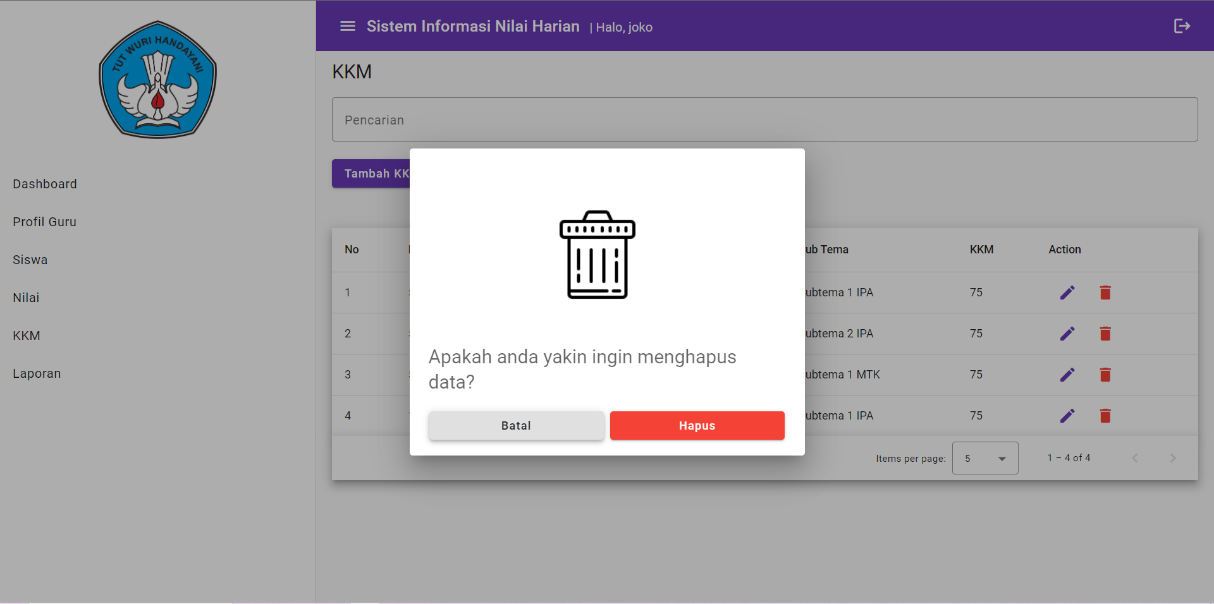
1. Halaman edit kkm



**Gambar 4.** Halaman Edit KKM

Dari **gambar 4.** Diatas merupakan halaman edit kkm, data pada field sudah terisi sesuai dengan data yang kita pilih. Terdapat pilihan batal dan simpan.

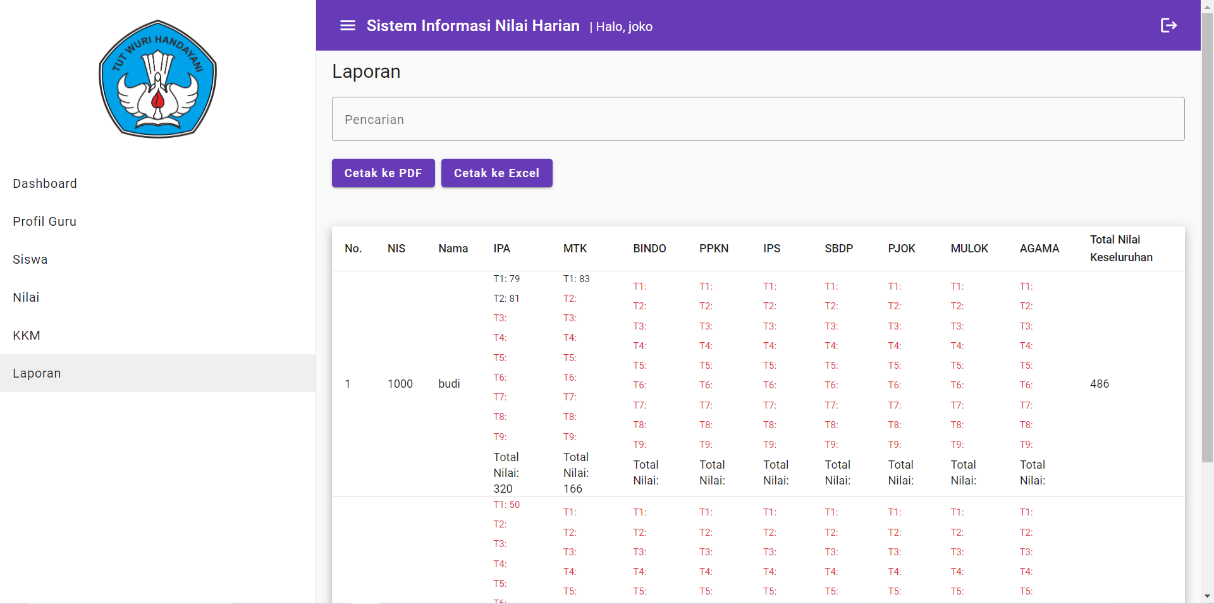
1. Halaman Hapus KKM



**Gambar 4.** Halaman Hapus KKM

Dari **gambar 4.** Diatas merupakan halaman yang muncul ketika kita ingin menghapus data KKM, terdapat pilihan batal dan hapus.

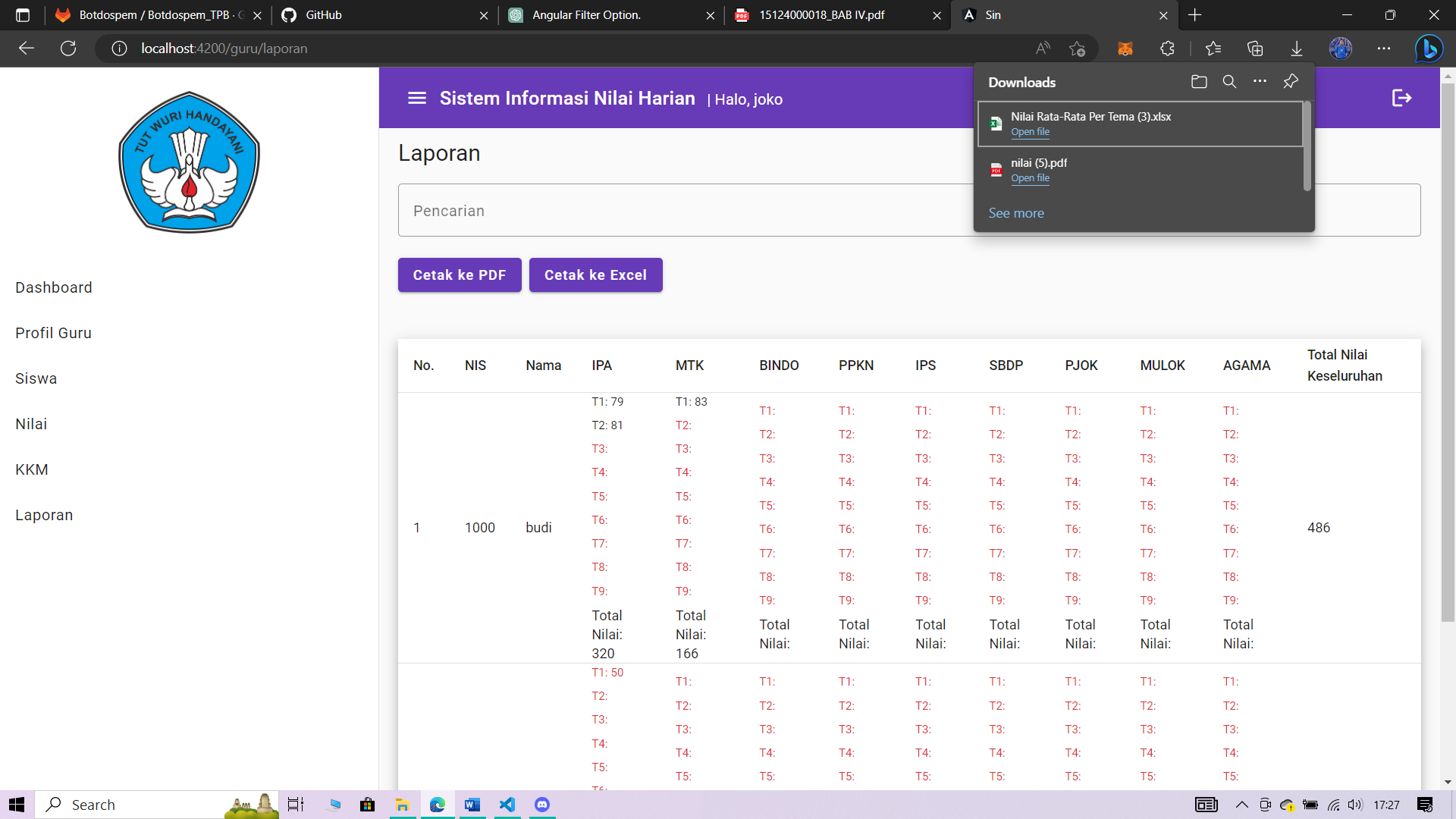
1. Halaman Laporan



**Gambar 4.** Halaman Laporan

Dari **gambar 4.** Diatas merupakan halaman bagian laporan yang berisikan data rata-rata nilai per tema dan per mata pelajaran. Dalam halaman laporan kita dapat melakukan pencarian data, cetak pdf, cetak excel, dan tabel nilai rata-rata. Pada tabel nilai berwarna hitam berarti berada diatas kkm dan nilai berwarna merah berarti dibawah kkm atau belum ada nilai.

1. Cetak PDF dan Cetak Excel

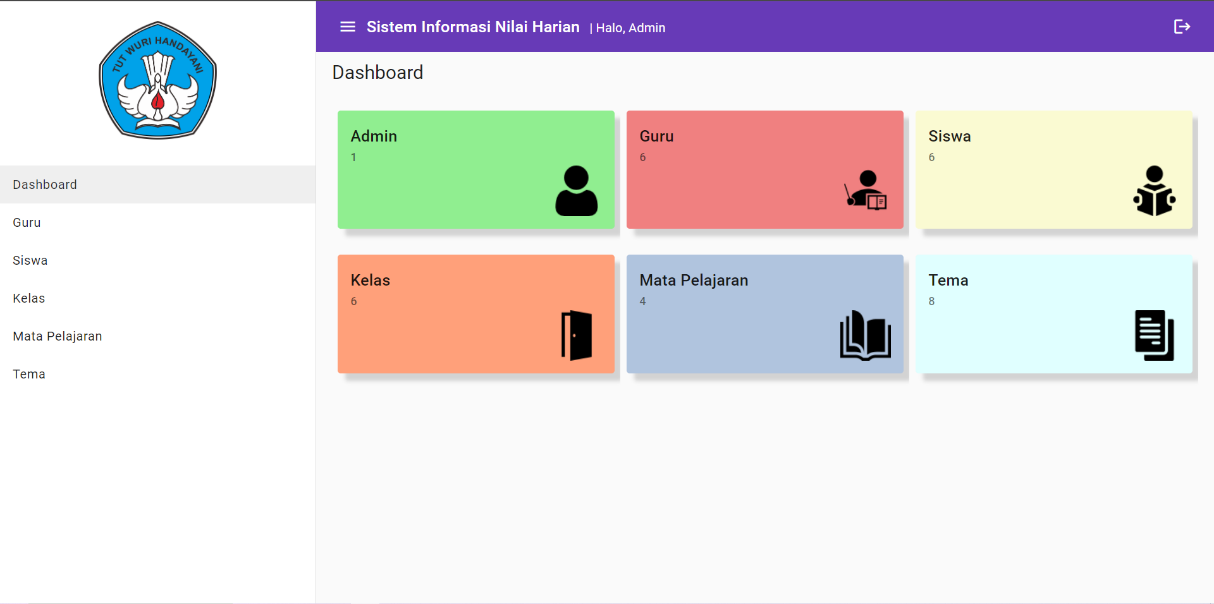


**Gambar 4.** Cetak PDF dan Cetak Excel

Dari **gambar 4.** Merupakan proses cetak data ke dalam bentuk pdf dan excel, sebagai contoh mencetak data laporan nilai. Saat button cetak pdf atau cetak excel di tekan, maka secara otomatis data laporan nilai akan terkonversi menjadi pdf atau excel.

### **4.9.2 Admin**

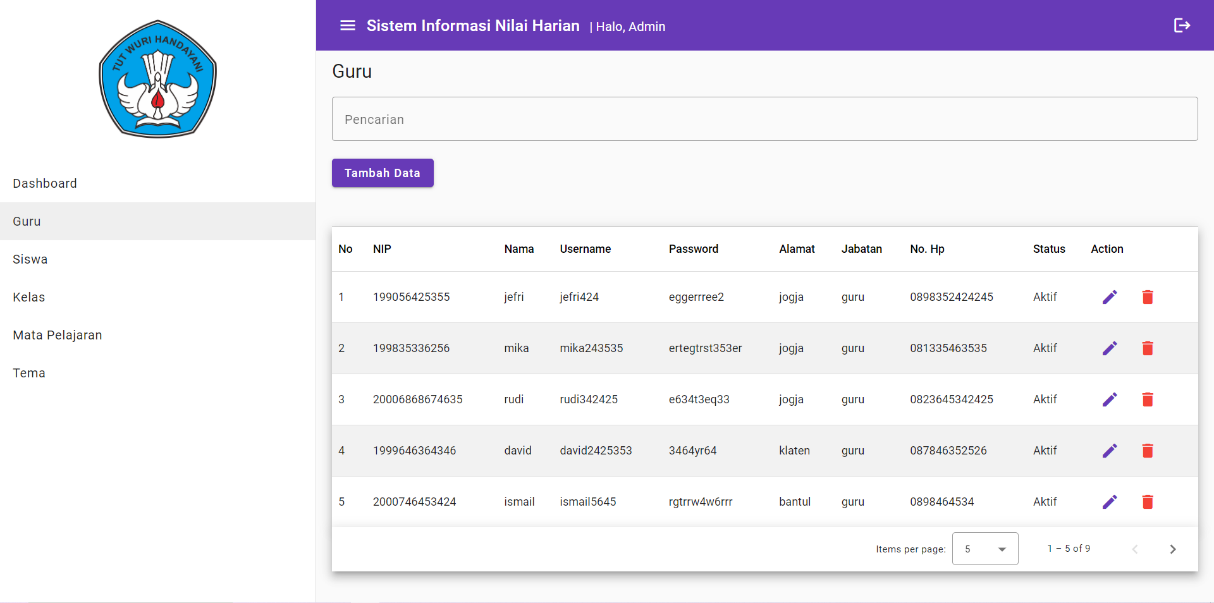
1. Halaman Dashboard Admin



**Gambar 4.** Halaman Dashboard Admin

Dari **gambar 4.** Diatas merupakan halaman dashboard dari admin, dimana terdapat beberapa card yang berisi jumlah data dalam system dan terdapat beberapa menu pada sidebar.

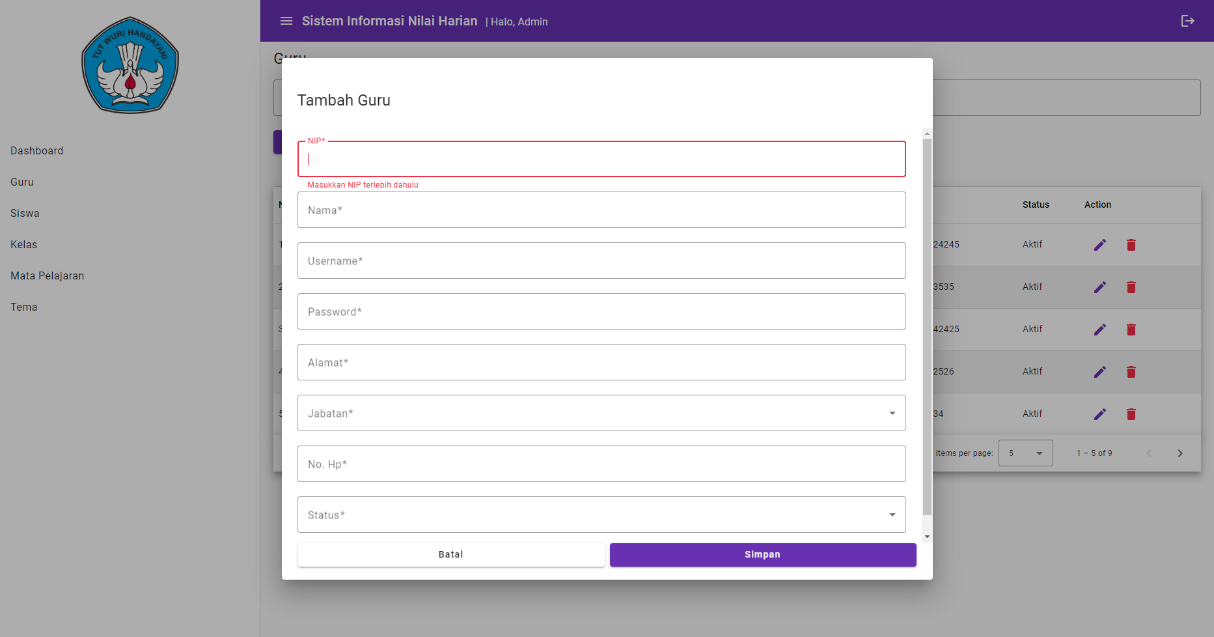
1. Halaman Guru



**Gambar 4.** Halaman Guru Admin

Pada **gambar 4.** Diatas merupakan halaman guru, dimana terdapat beberapa fitur yang tersedia yaitu pencarian, kelola data guru, dan tabel daftar guru.

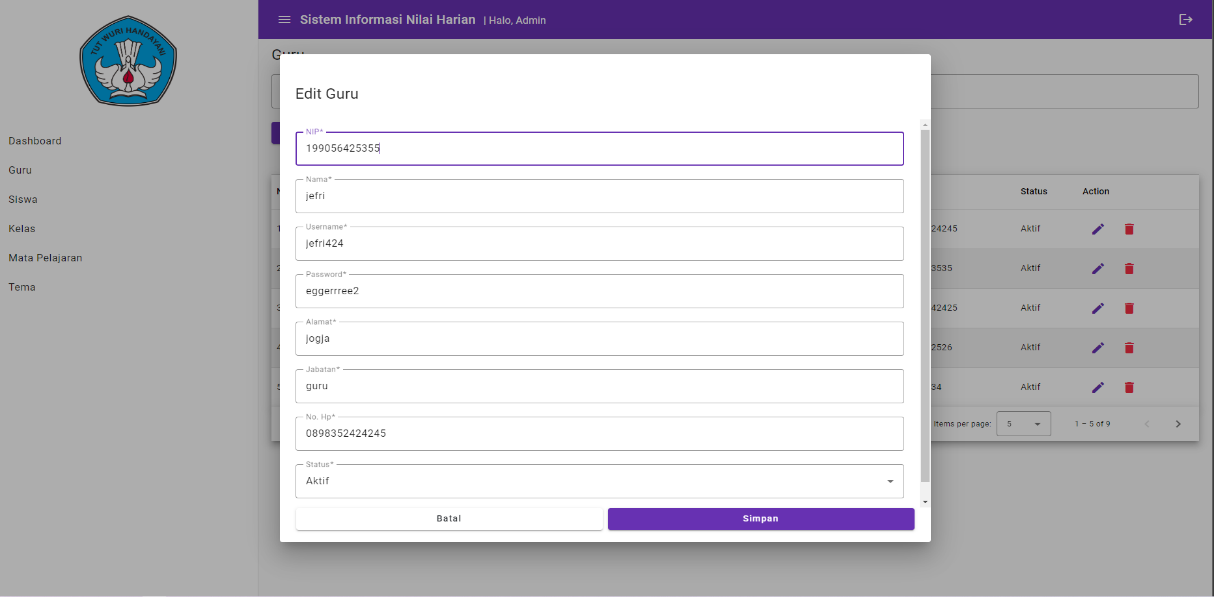
1. Halaman Tambah Guru



**Gambar 4.** Halaman Tambah Guru

Pada **gambar 4.** Diatas merupakan halaman tambah guru, dimana terdapat beberapa field yang harus disini terlebih dahulu mengenai data diri guru dan terdapat tombol batal dan simpan.

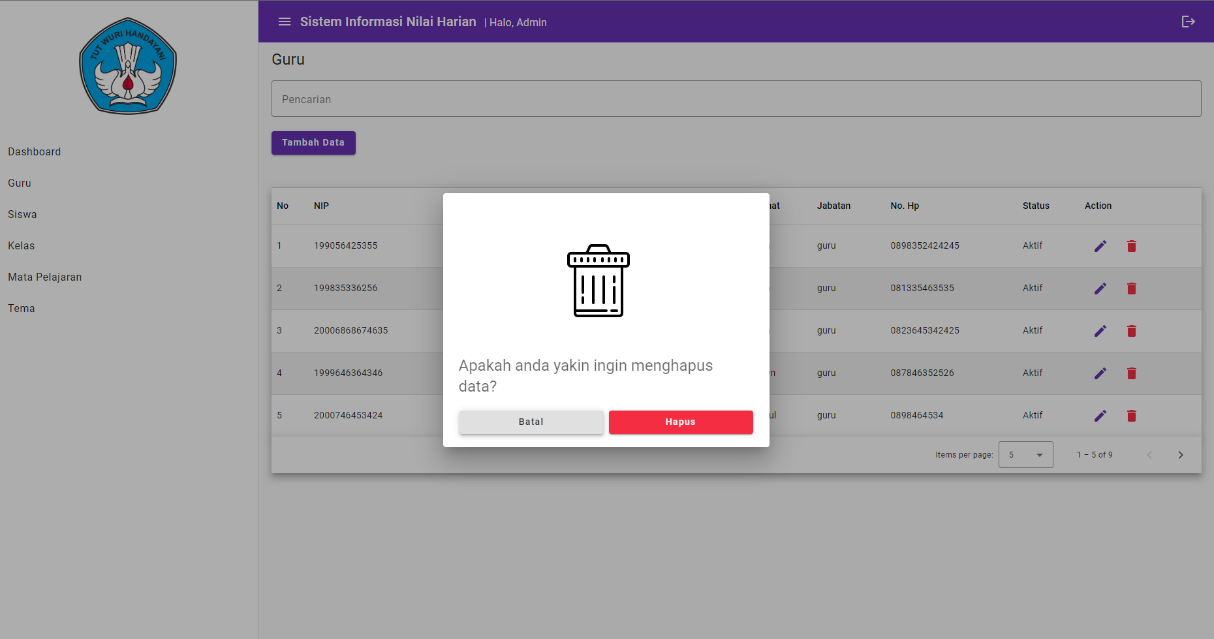
1. Halaman Edit Guru



**Gambar 4.** Halaman Edit Guru

Pada **gambar 4.** Diatas merupakan halaman edit guru, dimana terdapat data guru yang ingin diubah atau diupdate, serta terdapat tombol batal dan simpan.

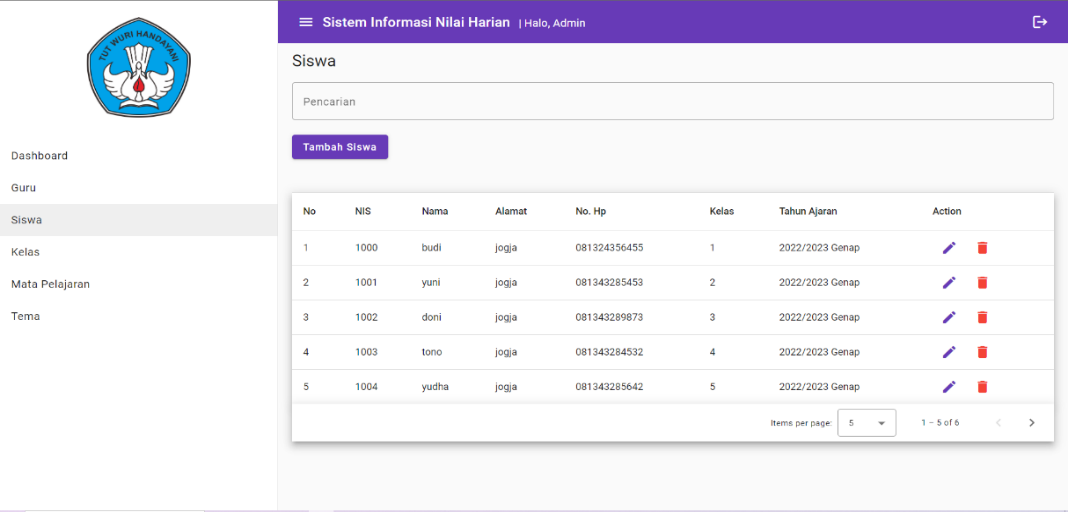
1. Halaman Hapus Guru



**Gambar 4.** Halaman Hapus Guru

Pada **gambar 4.** Diatas merupakan halaman hapus guru, dimana untuk konfirmasi dalam melakukan hapus data, serta terdapat tombol batal dan simpan.

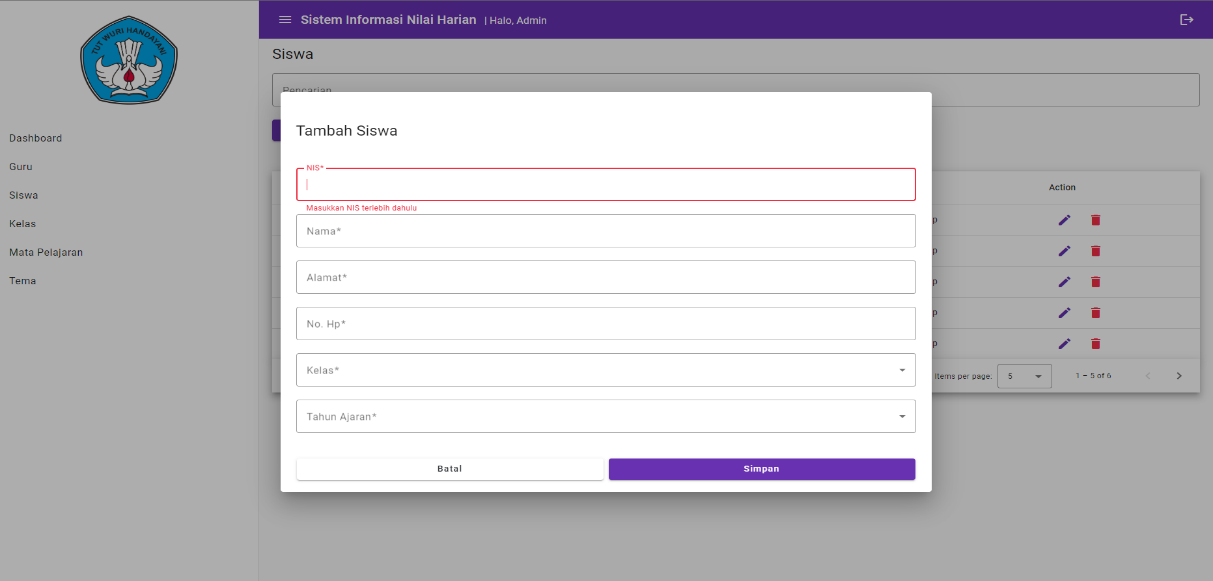
1. Halaman Siswa Admin



**Gambar 4.** Halaman Siswa Admin

Pada **gambar 4.** Diatas merupakan halaman siswa, dimana terdapat beberapa fitur yang tersedia yaitu pencarian, kelola data siswa, dan tabel daftar siswa.

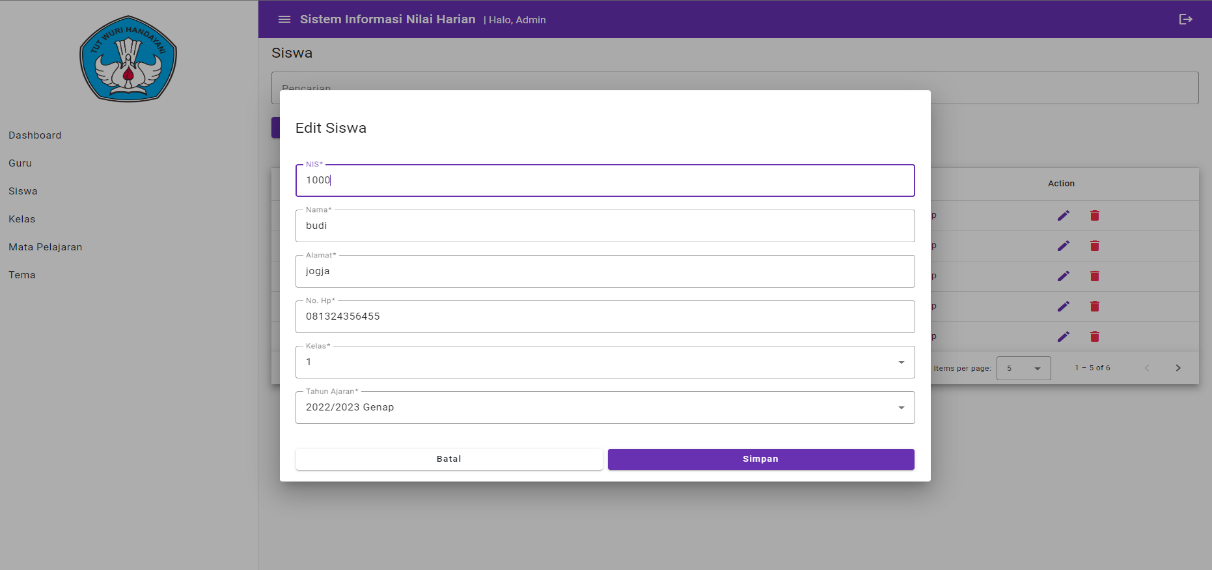
1. Halaman Tambah Siswa



**Gambar 4.** Halaman Tambah Siswa

Pada **gambar 4.** Diatas merupakan halaman tambah siswa, dimana terdapat beberapa field yang harus disini terlebih dahulu mengenai data diri siswa dan terdapat tombol batal dan simpan.

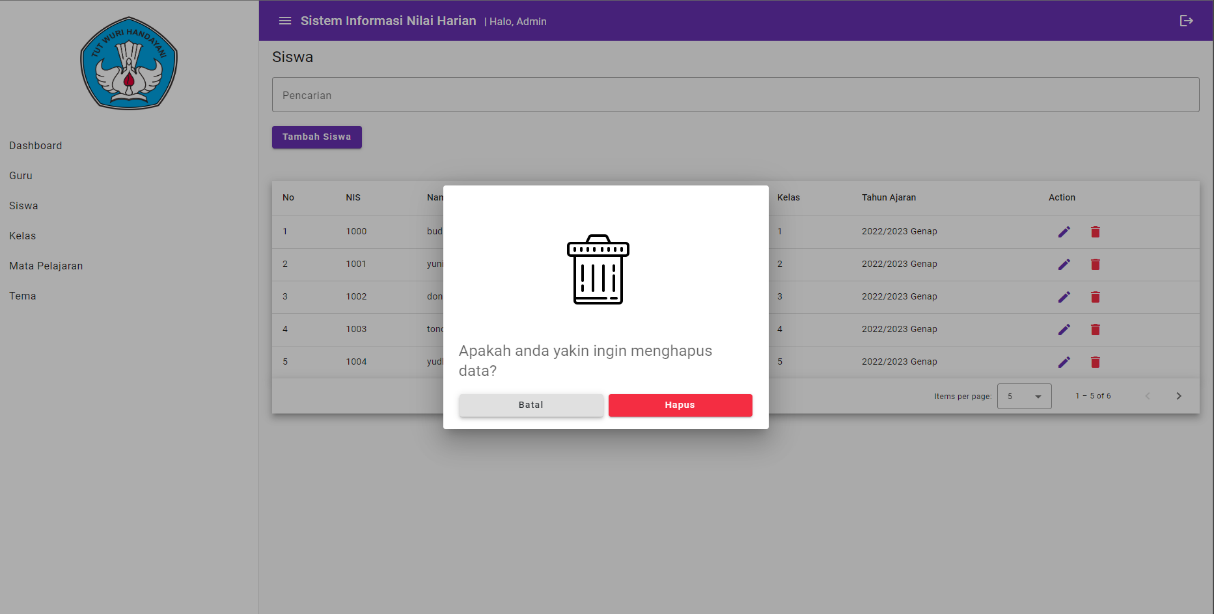
1. Halaman Edit Siswa



**Gambar 4.** Halaman Edit Siswa

Pada **gambar 4.** Diatas merupakan halaman edit siswa, dimana terdapat data guru yang ingin diubah atau diupdate, serta terdapat tombol batal dan simpan.

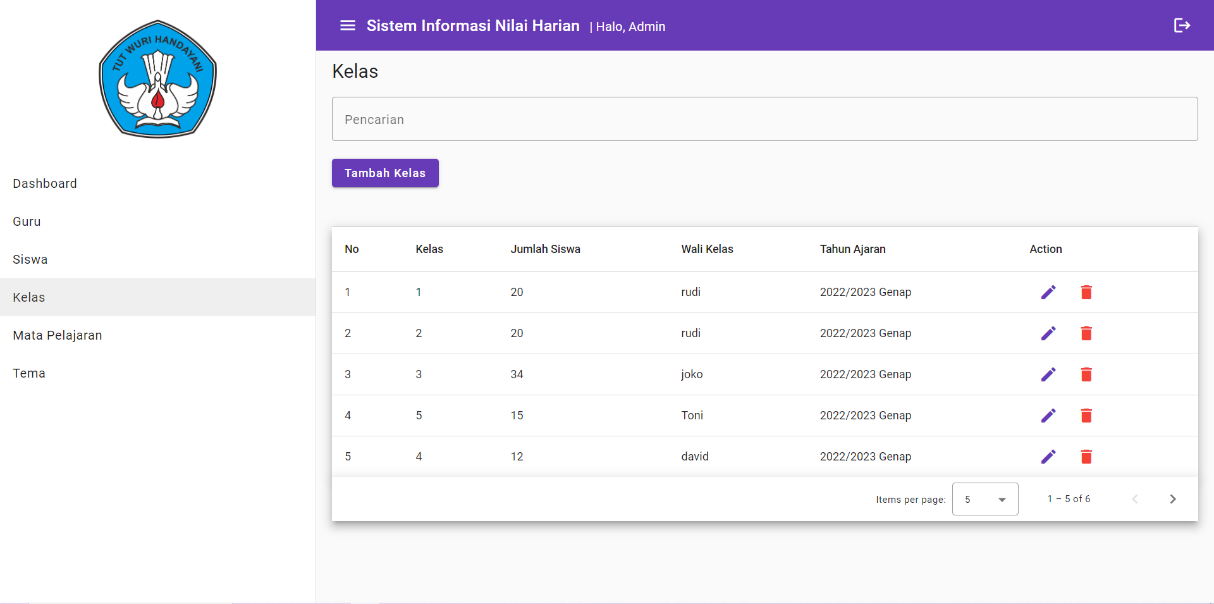
1. Halaman Hapus Siswa



**Gambar 4.** Halaman Hapus Siswa

Pada **gambar 4.** Diatas merupakan halaman hapus siswa, dimana untuk konfirmasi dalam melakukan hapus data, serta terdapat tombol batal dan simpan.

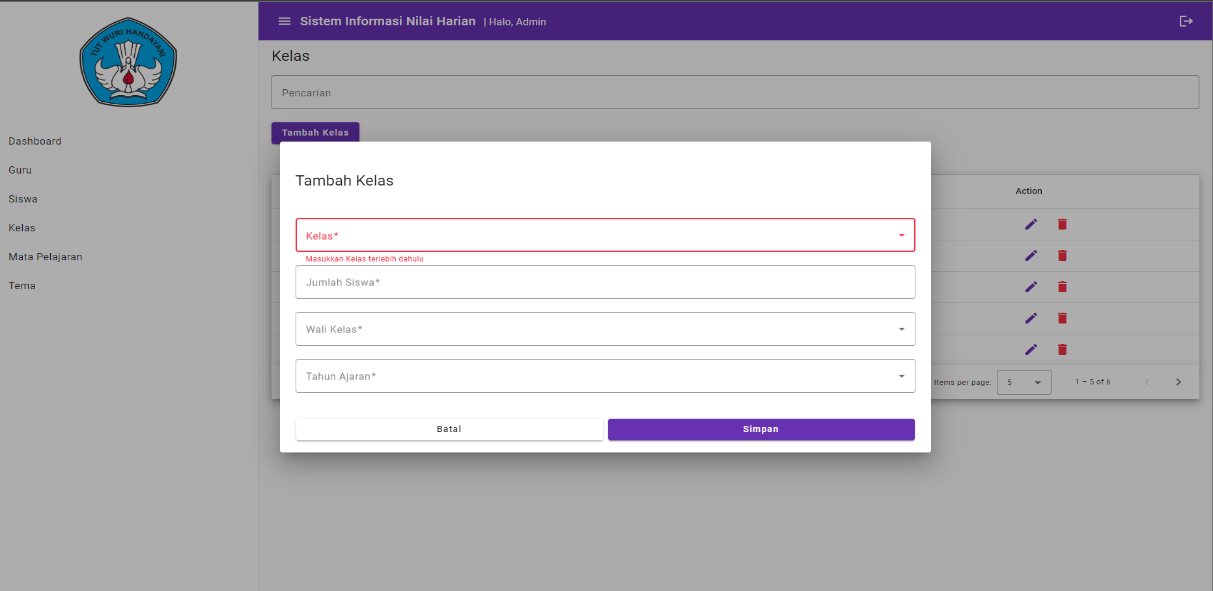
1. Halaman Kelas Admin



**Gambar 4.** Halaman Kelas Admin

Pada **gambar 4.** Diatas merupakan halaman kelas, dimana terdapat beberapa fitur yang tersedia yaitu pencarian, kelola data kelas, dan tabel daftar kelas.

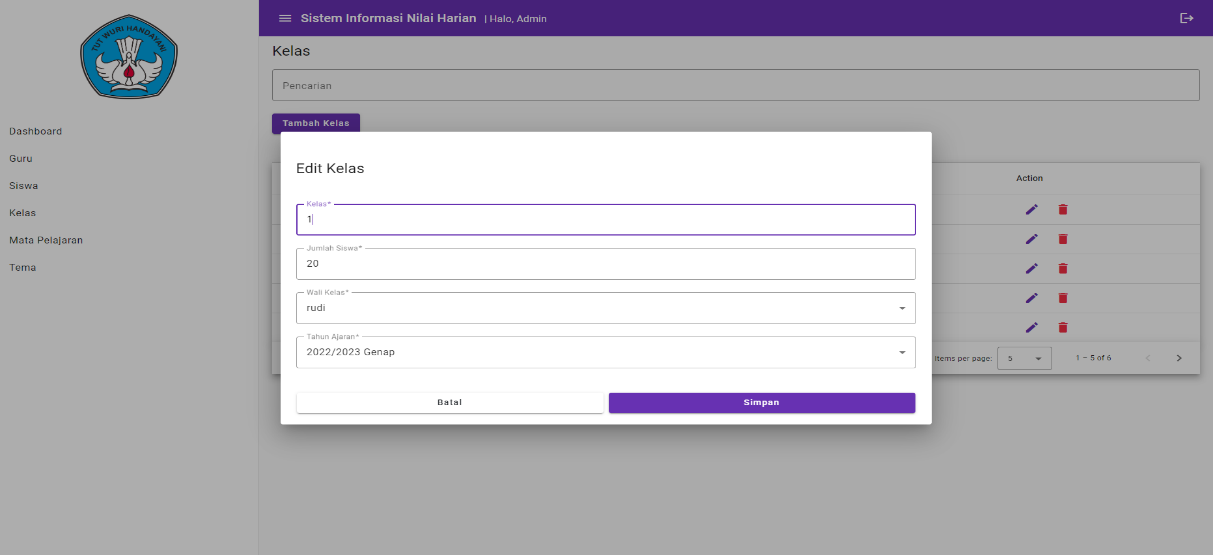
1. Halaman Tambah Kelas



**Gambar 4.** Halaman Tambah Kelas

Pada **gambar 4.** Diatas merupakan halaman tambah kelas, dimana terdapat beberapa field yang harus disini terlebih dahulu mengenai data kelas dan terdapat tombol batal dan simpan.

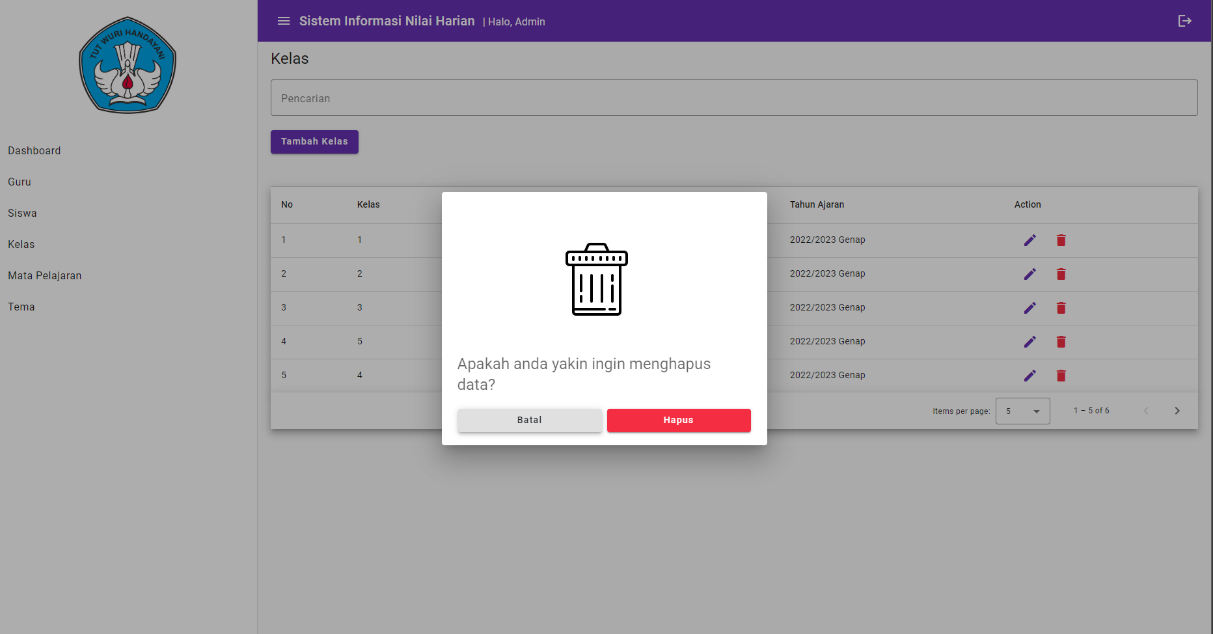
1. Halaman Edit Kelas



**Gambar 4.** Halaman Edit Kelas

Pada **gambar 4.** Diatas merupakan halaman edit kelas, dimana terdapat data guru yang ingin diubah atau diupdate, serta terdapat tombol batal dan simpan.

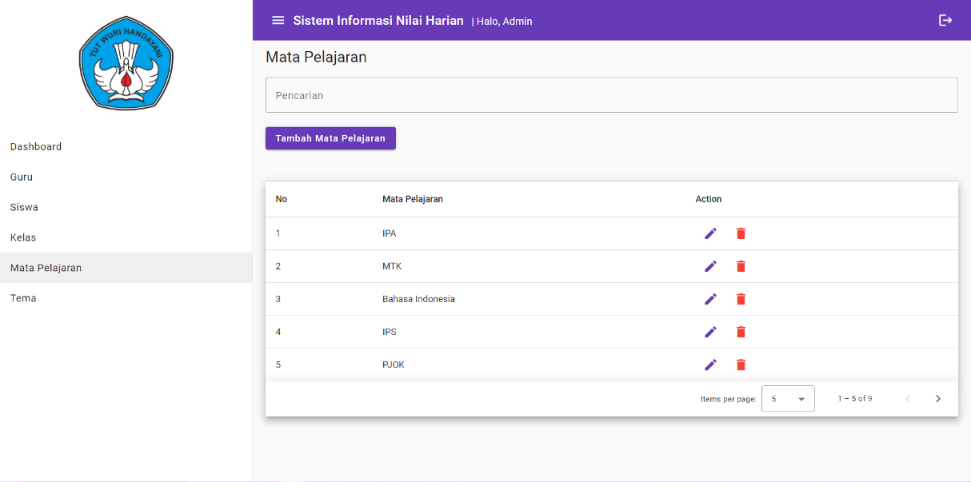
1. Halaman Hapus Kelas



**Gambar 4.** Halaman Hapus Kelas

Pada **gambar 4.** Diatas merupakan halaman hapus kelas, dimana untuk konfirmasi dalam melakukan hapus data, serta terdapat tombol batal dan simpan.

1. Halaman Mata Pelajaran Admin



**Gambar 4.** Halaman Mata Pelajaran Admin

Pada **gambar 4.** Diatas merupakan halaman mata pelajaran, dimana terdapat beberapa fitur yang tersedia yaitu pencarian, kelola data mata pelajaran, dan tabel daftar mata pelajaran.

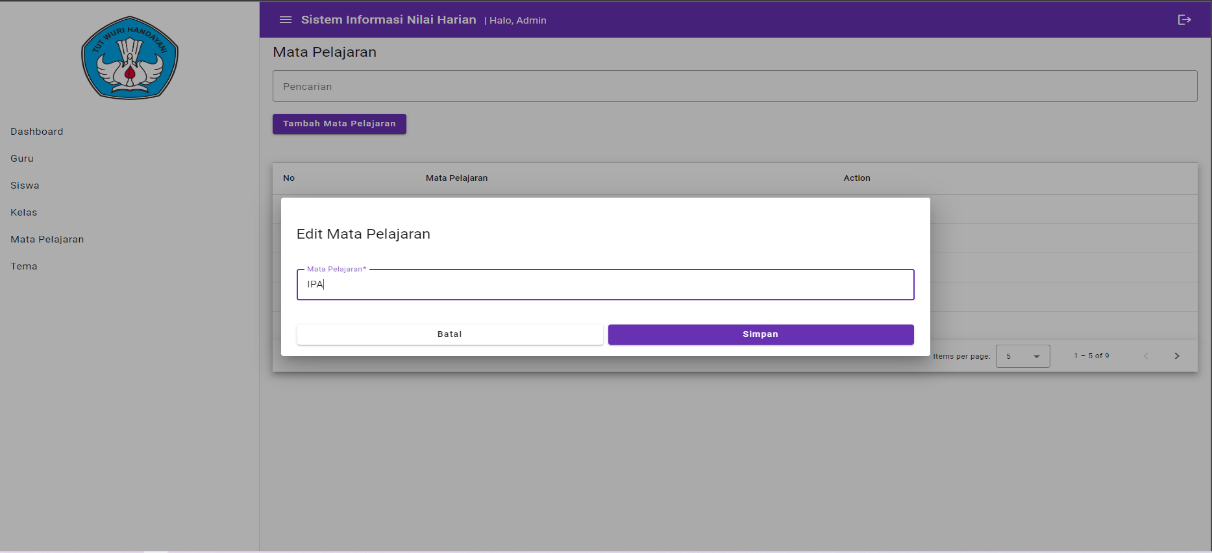
1. Halaman Tambah Mata Pelajaran



**Gambar 4.** Halaman Tambah Mata Pelajaran

Pada **gambar 4.** Diatas merupakan halaman tambah mata pelajaran, dimana terdapat beberapa field yang harus disini terlebih dahulu mengenai data kelas dan terdapat tombol batal dan simpan.

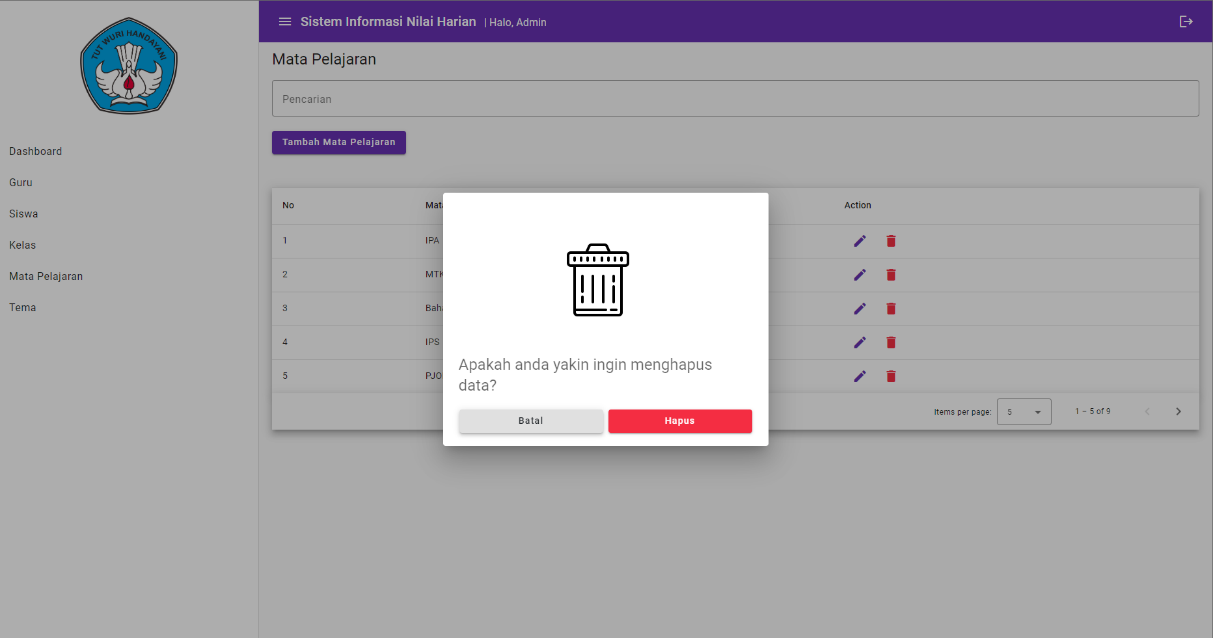
1. Halaman Edit Mata Pelajaran



**Gambar 4.** Halaman Edit Mata Pelajaran

Pada **gambar 4.** Diatas merupakan halaman edit mata pelajaran, dimana terdapat data guru yang ingin diubah atau diupdate, serta terdapat tombol batal dan simpan.

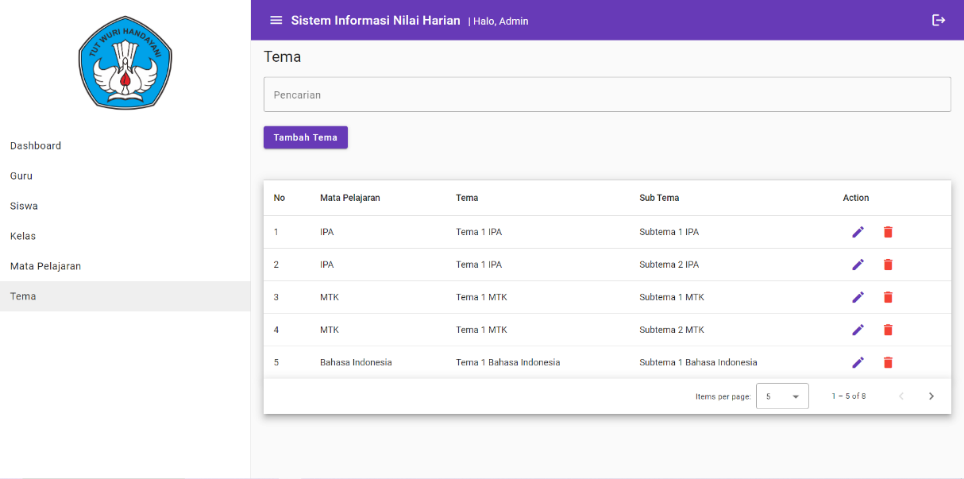
1. Halaman Hapus Mata Pelajaran



**Gambar 4.** Halaman Hapus Mata Pelajaran

Pada **gambar 4.** Diatas merupakan halaman hapus mata pelajaran, dimana untuk konfirmasi dalam melakukan hapus data, serta terdapat tombol batal dan simpan.

1. Halaman Tema Admin



**Gambar 4.** Halaman Tema Admin

Pada **gambar 4.** Diatas merupakan halaman tema, dimana terdapat beberapa fitur yang tersedia yaitu pencarian, kelola data tema, dan tabel daftar tema.

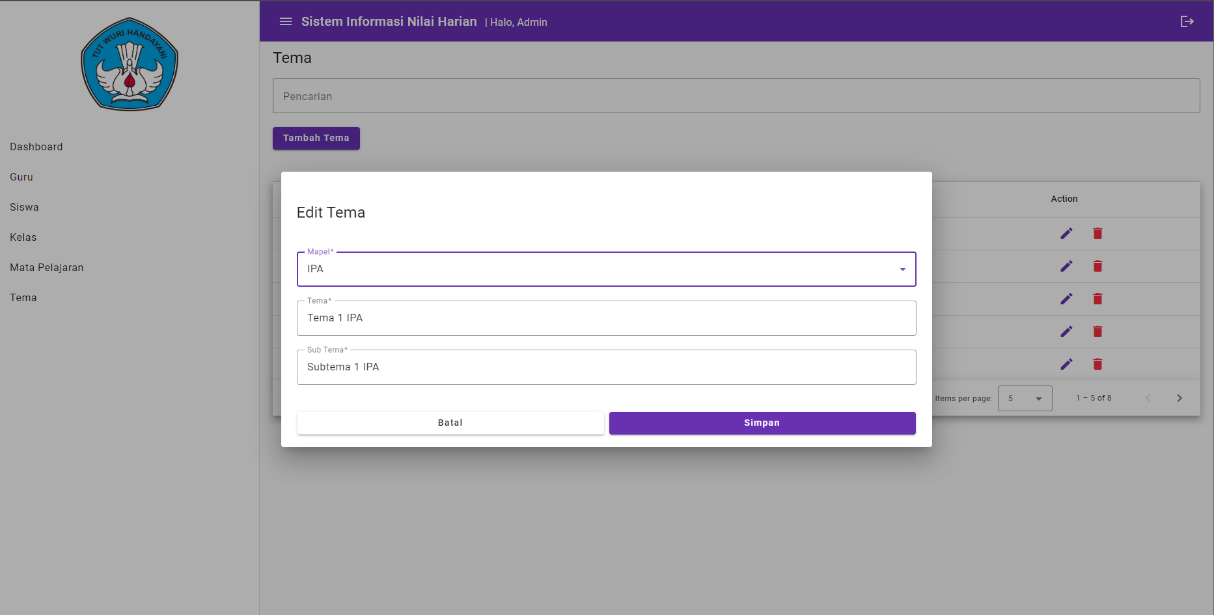
1. Halaman Tambah Tema



**Gambar 4.** Halaman Tambah Tema

Pada **gambar 4.** Diatas merupakan halaman tambah tema, dimana terdapat beberapa field yang harus disini terlebih dahulu mengenai data tema dan terdapat tombol batal dan simpan.

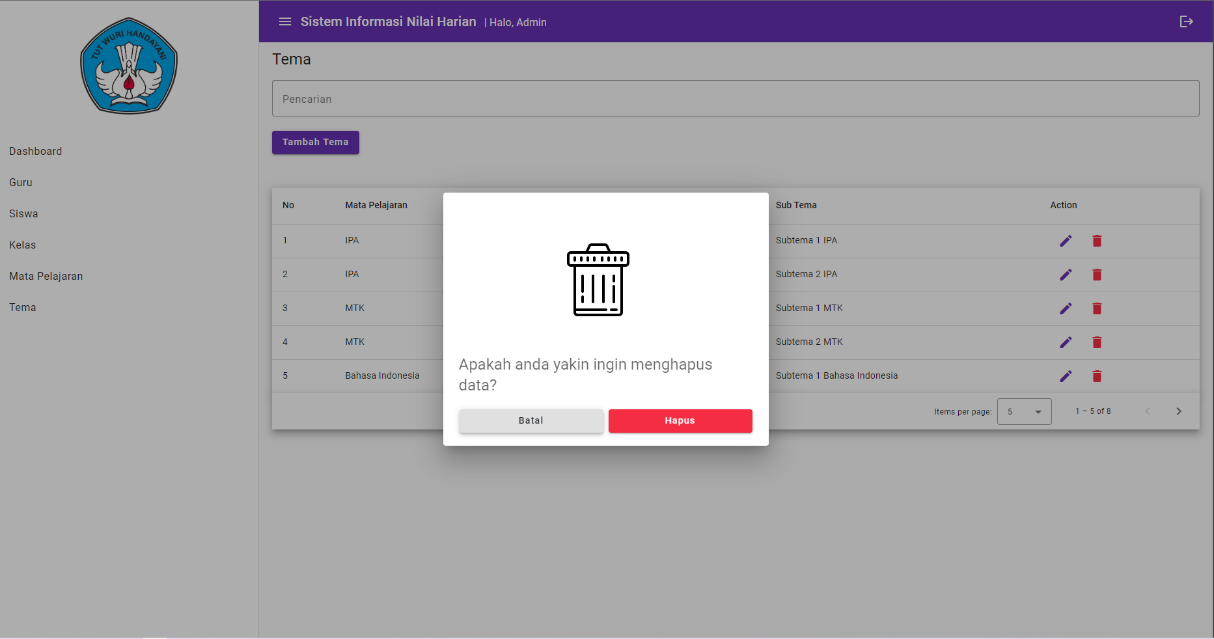
1. Halaman Edit Tema



**Gambar 4.** Halaman Edit Tema

Pada **gambar 4.** Diatas merupakan halaman edit tema, dimana terdapat data guru yang ingin diubah atau diupdate, serta terdapat tombol batal dan simpan.

1. Halaman Hapus Tema



**Gambar 4.** Halaman Hapus Tema

Pada **gambar 4.** Diatas merupakan halaman hapus tema, dimana untuk konfirmasi dalam melakukan hapus data, serta terdapat tombol batal dan simpan.

## **4.11 Black Box Testing**

Pada penelitian ini akan menggunakan salah satu dari teknik blackbox testing yaitu Equivalence Partitioning. Pengujian blackbox testing berbasis Equivalence Partitioning merupakan salah satu teknik pengujian fungsional yang digunakan untuk menemukan kesalahan atau cacat pada sebuah sistem atau aplikasi. Teknik ini bertujuan untuk membagi input atau masukan menjadi beberapa kelompok yang setara atau setara dengan prinsip kesamaan fungsional, sehingga setiap kelompok dapat dianggap sebagai representasi dari satu set kondisi input yang serupa. Tidak hanya itu dengan menggunakan teknik Equivalence Partitioning kita dapat menguji sebuah aplikasi menjadi lebih efisien, cakupan pengujian lebih baik, mudah dikelola dan dipahami, mengidentifikasi kesalahan pada sistem lebih cepat, dan mengurangi redundansi dalam melakukan pengujian.

Dalam tahap pengujian menggunakan teknik Equivalence Partitioning, langkah awal yang harus dilakukan adalah merancang test case berdasarkan fungsi yang ingin diuji pada perangkat lunak. Selanjutnya, kita perlu menentukan batasan-batasan untuk setiap partition dalam Equivalence Partitioning. Setelah menentukan batasan-batasan tersebut, langkah selanjutnya adalah membuat model pengujian berdasarkan skenario pengujian dan hasil yang diharapkan. Tahap terakhir adalah melakukan pengujian berdasarkan model yang telah dirancang sebelumnya untuk mendapatkan data berupa dokumentasi pengujian menggunakan metode Equivalence Partitions. Dari dokumentasi tersebut, dapat dihitung tingkat efektivitas pengujian yang dilakukan dengan menggunakan teknik Equivalence Partitions.

### **4.10.1 Guru Test Case**

1. Login

**Tabel 4.** Rancangan Test Case Form login

|  |  |  |
| --- | --- | --- |
| **Id** | **Deskripsi Pengujian** | **Hasil Yang Diharapkan** |
| A001 | Mengisi field username dan password yang benar, lalu menekan tombol Login | Sistem menerima dan menampilkan Pemberitahuan “Login Berhasil, Login Sebagai Guru” Kemudian tampil halaman dashboard |
| A002 | Mengisi field username dan password yang salah lalu menekan tombol Login | Sistem menolak dan menampilkan Pemberitahuan “Login Gagal, Mohon Cek Kembali Username dan Password Anda” |
| A003 | Mengosongkan salah satu field username dan password dan menekan tombol Login | Sistem menolak dan menampilkan Pemberitahuan “Login Gagal, Mohon Cek Kembali Username dan Password Anda”. Serta pada field menampilkan pemberitahuan “Tolong masukkan Username/Password terlebih dahulu.” |
| A004 | Mengosongkan field username dan password dan menekan tombol Login | Sistem menolak dan menampilkan Pemberitahuan “Login Gagal, Mohon Cek Kembali Username dan Password Anda”. Serta pada field menampilkan pemberitahuan “Tolong masukkan Username/Password terlebih dahulu.” |

1. Profil Guru

**Tabel 4.** Rancangan Test Case Profil Guru

|  |  |  |
| --- | --- | --- |
| **Id** | **Deskripsi Pengujian** | **Hasil Yang Diharapkan** |
| B001 | Melakukan pengecekan apakah data yang ditampilkan sesuai dengan data guru yang login ke dalam aplikasi | Sistem menampilkan data guru sesuai dengan guru yang mengakses aplikasi saat itu. |

1. Siswa

**Tabel 4.** Rancangan Test Case Siswa

|  |  |  |
| --- | --- | --- |
| **Id** | **Deskripsi Pengujian** | **Hasil Yang Diharapkan** |
| C001 | Mengisi field pencarian dengan NIS | Sistem menampilkan siswa berdasarkan NIS yang diinputkan. |
| C002 | Mengisi field pencarian dengan Nama | Sistem menampilkan siswa berdasarkan Nama yang diinputkan. |
| C003 | Mengisi field pencarian dengan Kelas | Sistem menampilkan siswa berdasarkan Kelas yang diinputkan. |
| C004 | Mengisi field pencarian dengan selain NIS/Nama/Kelas | Sistem menampilkan tabel kosong. |

1. Nilai

**Tabel 4.** Rancangan Test Case Nilai

|  |  |  |
| --- | --- | --- |
| **Id** | **Deskripsi Pengujian** | **Hasil Yang Diharapkan** |
| D001 | Mengisi field pencarian dengan NIS | Sistem menampilkan siswa berdasarkan NIS yang diinputkan. |
| D002 | Mengisi field pencarian dengan Nama | Sistem menampilkan siswa berdasarkan Nama yang diinputkan. |
| D003 | Mengisi field pencarian dengan Mata Pelajaran | Sistem menampilkan siswa berdasarkan Mata Pelajaran yang diinputkan. |
| D004 | Mengisi field pencarian dengan selain NIS/Nama/Mata Pelajaran | Sistem menampilkan tabel kosong. |
| D005 | Menekan tombol Tambah Nilai | Sistem menampilkan dialog Tambah Nilai yang berisi beberapa field. |
| D006 | Menekan tombol Batal pada dialog Tambah Nilai | Sistem menutup dialog Tambah Nilai |
| D007 | Mengisi field NIS "1000", Nama "Budi", Nilai "80", Mata Pelajaran "IPA", Tema "Tema 1 IPA", Subtema "Subtema 1 IPA" dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Nilai, Berhasil Menambah Nilai” Kemudian tampil halaman Nilai |
| D008 | Mengisi field NIS "1000", Nama "Budi", Nilai "80", Mata Pelajaran "IPA", Tema "Tema 1 IPA", Subtema " ", kemudian mengosongkan salah satu field, dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Nilai, Gagal Menambah Nilai”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. |
| D009 | Mengosongkan semua field dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Nilai, Gagal Menambah Nilai”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. |
| D010 | Mengisi field Nis dengan string ‘’Seribu” dan menekan Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Nilai, Gagal Menambah Nilai”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. |
| D0011 | Mengisi field Nilai 60 dimana kurang dari 65 (KKM) dan menekan Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Nilai, Berhasil Menambah Nilai”, tetapi muncul pemberitahuan “Nilai yang dimasukkan (60) kurang dari KKM (65)`, 'Masuk Kategori Remidial” dan kemudian tampil halaman Nilai, pada halaman nilai akan berwarna merah. |
| D012 | Mengisi field Nilai 80 dimana lebih dari 65 (KKM) dan menekan Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Nilai, Berhasil Menambah Nilai” Kemudian tampil halaman Nilai |
| D013 | Mengisi field Nilai dengan string ‘’Delapan Puluh” dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Nilai, Gagal Menambah Nilai”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. |
| D014 | Mengisi field nilai kurang dari 0 (-50) dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Nilai, Gagal Menambah Nilai”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. |
| D015 | Mengisi field nilai lebih dari 100 (101) dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Nilai, Gagal Menambah Nilai”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. |
| D016 | Menekan tombol Edit | Sistem menampilkan dialog Edit Nilai yang berisi beberapa field. Isi dari field sesuai dengan data yang ingin di edit. |
| D017 | Mengisi field Mata Pelajaran, Tema, Sub Tema sama dengan data sudah terdaftar dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Nilai, Data sudah terdaftar, Gagal Menambah Nilai”. |
| D018 | Menekan tombol Batal pada dialog Edit Nilai | Sistem menutup dialog Edit Nilai |
| D019 | Mengubah isi dari field dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Nilai, Berhasil Mengupdate Nilai” Kemudian tampil halaman Nilai |
| D020 | Mengosongkan isi field dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Nilai, Gagal Mengupdate Nilai”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. |
| D021 | Mengubah NIS “1000”, menjadi “Seribu” dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Nilai, Gagal Mengupdate Nilai”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. |
| D022 | Mengubah Nilai “75”, menjadi “50” dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Nilai, Berhasil Menambah Nilai”, tetapi muncul pemberitahuan “Nilai yang dimasukkan (50) kurang dari KKM (65)`, 'Masuk Kategori Remidial” dan kemudian tampil halaman Nilai, pada halaman nilai akan berwarna merah. |
| D023 | Mengubah Nilai “75”, menjadi “-75” dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Nilai, Gagal Mengupdate Nilai”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. |
| D024 | Mengubah Nilai “75”, menjadi “101” dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Nilai, Gagal Mengupdate Nilai”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. |
| D025 | Mengubah Nilai “75”, menjadi “Tujuh Puluh Lima” dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Nilai, Gagal Mengupdate Nilai”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. |
| D026 | Mengubah isi field dan menekan tombol Batal | Sistem menutup dialog Edit Nilai, data tidak berubah, dan menampilkan halaman Nilai. |
| D027 | Menekan tombol Hapus | Sistem menampilkan dialog Hapus Data. |
| D028 | Menekan tombol Batal pada dialog Hapus Nilai | Sistem menutup dialog Hapus Nilai |
| D029 | Menekan tombol Hapus pada dialog Hapus Nilai | Sistem menghapus Data Nilai sesuai dengan yang dipilih, sistem menutup dialog Hapus Nilai, dan menampilkan halaman Nilai. |
| D030 | Menekan tombol Cetak ke PDF | Sistem secara otomatis akan mengkonversi data tabel ke dalam bentuk PDF dan langsung terunduh. |
| D031 | Menekan tombol Cetak ke Excel | Sistem secara otomatis akan mengkonversi data tabel ke dalam bentuk Excel dan langsung terunduh. |

1. KKM

**Tabel 4.** Rancangan Test Case Nilai

|  |  |  |
| --- | --- | --- |
| **Id** | **Deskripsi Pengujian** | **Hasil Yang Diharapkan** |
| E001 | Mengisi field pencarian dengan Kelas | Sistem menampilkan siswa berdasarkan Kelas yang diinputkan. |
| E002 | Mengisi field pencarian dengan Nama | Sistem menampilkan siswa berdasarkan Nama yang diinputkan. |
| E003 | Mengisi field pencarian dengan selain NIS/Nama/Mata Pelajaran | Sistem menampilkan tabel kosong. |
| E004 | Menekan tombol Tambah KKM | Sistem menampilkan dialog Tambah KKM yang berisi beberapa field. |
| E005 | Menekan tombol Batal pada dialog Tambah KKM | Sistem menutup dialog Tambah KKM |
| E006 | Mengisi field Kelas "5", Mata Pelajaran "IPA", Tema "Tema 1 IPA", Subtema "Subtema 1 IPA", KKM "65"dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data KKM, Berhasil Menambah KKM” kemudian tampil halaman KKM | |
| E007 | Mengisi field Kelas "5", Mata Pelajaran "IPA", Tema "Tema 1 IPA", Subtema "Subtema 1 IPA", KKM " "kemudian mengosongkan salah satu field, dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data KKM, Gagal Menambah KKM”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | |
| E008 | Mengisi field KKM dengan string ‘’Delapan Puluh” dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data KKM, Gagal Menambah KKM”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. |
| E009 | Mengisi field nilai kurang dari 0 (-1) dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data KKM, Gagal Menambah KKM”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. |
| E010 | Mengisi field nilai lebih dari 100 (101) dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data KKM, Gagal Menambah KKM”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. |
| E011 | Menekan tombol Edit KKM | Sistem menampilkan dialog Edit KKM yang berisi beberapa field. Isi dari field sesuai dengan data yang ingin di edit. |
| E012 | Mengisi field Mata Pelajaran, Tema, Sub Tema sama dengan data sudah terdaftar dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data KKM, Data sudah terdaftar, Gagal Menambah KKM”. |
| E013 | Menekan tombol Batal pada dialog Edit KKM | Sistem menutup dialog Edit KKM |
| E014 | Mengubah isi dari field dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data KKM, Berhasil Mengupdate KKM” Kemudian tampil halaman KKM. |
| E015 | Mengosongkan isi field dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data KKM, Gagal Mengupdate KKM”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. |
| E016 | Mengubah isi field dan menekan tombol Batal | Sistem menutup dialog Edit Nilai, data tidak berubah, dan menampilkan halaman KKM. |
| E017 | Menekan tombol Hapus | Sistem menampilkan dialog Hapus Data. |
| E018 | Menekan tombol Batal pada dialog Hapus KKM | Sistem menutup dialog Hapus KKM |
| E019 | Menekan tombol Hapus pada dialog Hapus KKM | Sistem menghapus Data KKM sesuai dengan yang dipilih, sistem menutup dialog Hapus KKM, dan menampilkan halaman KKM. |

1. Laporan

**Tabel 4.** Rancangan Test Case Nilai

|  |  |  |
| --- | --- | --- |
| **Id** | **Deskripsi Pengujian** | **Hasil Yang Diharapkan** |
| F001 | Mengisi field pencarian dengan NIS | Sistem menampilkan siswa berdasarkan NIS yang diinputkan. |
| F002 | Mengisi field pencarian dengan Nama | Sistem menampilkan siswa berdasarkan Nama yang diinputkan. |
| F003 | Melakukan cek pada tabel Laporan | Terdapat data nilai siswa yang sudah dalam bentuk nilai rata-rata per tema(T = Tema) dan per mata pelajaran. |
| F004 | Melakukan cek pada data nilai | Nilai sudah terbagi per mata pelajaran dan per tema, dimana dalam satu tema terdapat 4 sub tema. Jadi nilai per tema tersebut adalah total nilai 4 sub tema dibagi jumlah sub tema. |
| F005 | Melakukan cek pada nilai rata-rata | Nilai yang sudah berada diatas KKM akan berwarna hitam dan jika dibawah KKM atau nilai belum diinputkan maka akan berwarna merah. |
| F006 | Menekan tombol Cetak ke PDF | Sistem secara otomatis akan mengkonversi data tabel ke dalam bentuk PDF dan langsung terunduh. |
| F007 | Menekan tombol Cetak ke Excel | Sistem secara otomatis akan mengkonversi data tabel ke dalam bentuk Excel dan langsung terunduh. |

### **4.10.1 Admin Test Case**

1. Login

**Tabel 4.** Rancangan Test Case Login

|  |  |  |
| --- | --- | --- |
| **Id** | **Deskripsi Pengujian** | **Hasil Yang Diharapkan** |
| A001 | Mengisi field username dan password yang benar, lalu menekan tombol Login | Sistem menerima dan menampilkan Pemberitahuan “Login Berhasil, Login Sebagai Guru” Kemudian tampil halaman dashboard |
| A002 | Mengisi field username dan password yang salah, lalu menekan tombol Login | Sistem menolak dan menampilkan Pemberitahuan “Login Gagal, Mohon Cek Kembali Username dan Password Anda” |
| A003 | Mengosongkan salah satu field username dan password dan menekan tombol Login | Sistem menolak dan menampilkan Pemberitahuan “Login Gagal, Mohon Cek Kembali Username dan Password Anda”. Serta pada field menampilkan pemberitahuan “Tolong masukkan Username/Password terlebih dahulu.” |
| A004 | Mengosongkan field username dan password dan menekan tombol Login | Sistem menolak dan menampilkan Pemberitahuan “Login Gagal, Mohon Cek Kembali Username dan Password Anda”. Serta pada field menampilkan pemberitahuan “Tolong masukkan Username/Password terlebih dahulu.” |

1. Guru

**Tabel 4.** Rancangan Test Case Guru

|  |  |  |
| --- | --- | --- |
| **Id** | **Deskripsi Pengujian** | **Hasil Yang Diharapkan** |
| B001 | Mengisi field pencarian dengan NIP | Sistem menampilkan siswa berdasarkan NIP yang diinputkan. |
| B002 | Mengisi field pencarian dengan Nama | Sistem menampilkan siswa berdasarkan Nama yang diinputkan. |
| B003 | Mengisi field pencarian dengan selain NIS/Nama | Sistem menampilkan tabel kosong. |
| B004 | Menekan tombol Tambah Guru | Sistem menampilkan dialog Tambah Guru yang berisi beberapa field. |
| B005 | Menekan tombol Batal pada dialog Tambah Guru | Sistem menutup dialog Tambah Guru |
| B006 | Mengisi field NIP "123", Nama "Samsudin", Username"sms007", Password "sms777", Alamat "Yogyakarta", Jabatan "Guru", No. Hp "123", Status "Aktif" dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Guru, Berhasil Menambah Guru” kemudian tampil halaman Guru |
| B007 | Mengisi field NIP " ", Nama "Samsudin", Username"sms007", Password "sms777", Alamat "Yogyakarta", Jabatan "Guru", No. Hp "123", Status "Aktif" kemudian mengosongkan salah satu field, dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Guru, Gagal Menambah Guru”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. |
| B008 | Mengisi field NIP dengan string ‘’satu dua tiga” dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Guru, Gagal Menambah Guru”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. |
| B009 | Mengisi field No. Hp dengan string ‘’satu dua tiga” dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Guru, Gagal Menambah Guru”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. |
| B010 | Mengisi field NIP sama dengan NIP sudah terdaftar dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Guru, NIP sudah teraftar, Gagal Menambah Guru”. |
| B011 | Menekan tombol Edit Guru | Sistem menampilkan dialog Edit Guru yang berisi beberapa field. Isi dari field sesuai dengan data yang ingin di edit. |
| B012 | Menekan tombol Batal pada dialog Edit Guru | Sistem menutup dialog Edit Guru |
| B013 | Mengubah isi dari field dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Guru, Berhasil Mengupdate Guru” Kemudian tampil halaman Guru. |
| B014 | Mengosongkan isi field dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Guru, Gagal Mengupdate Guru”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. |
| B015 | Mengubah isi field dan menekan tombol Batal | Sistem menutup dialog Edit Guru, data tidak berubah, dan menampilkan halaman Guru. |
| B016 | Menekan tombol Hapus | Sistem menampilkan dialog Hapus Data. |
| B017 | Menekan tombol Batal pada dialog Hapus Nilai | Sistem menutup dialog Hapus Guru |
| B018 | Menekan tombol Hapus pada dialog Hapus Nilai | Sistem menghapus Data Guru sesuai dengan yang dipilih, sistem menutup dialog Hapus Guru, dan menampilkan halaman Guru. |

1. Siswa

**Tabel 4.** Rancangan Test Case Guru

|  |  |  |
| --- | --- | --- |
| **Id** | **Deskripsi Pengujian** | **Hasil Yang Diharapkan** |
| C001 | Mengisi field pencarian dengan NIS | Sistem menampilkan siswa berdasarkan NIS yang diinputkan. |
| C002 | Mengisi field pencarian dengan Nama | Sistem menampilkan siswa berdasarkan Nama yang diinputkan. |
| C003 | Mengisi field pencarian dengan Kelas | Sistem menampilkan siswa berdasarkan Kelas yang diinputkan. |
| C004 | Mengisi field pencarian dengan selain NIS/Nama/Kelas | Sistem menampilkan tabel kosong. |
| C005 | Menekan tombol Tambah Siswa | Sistem menampilkan dialog Tambah Siswa yang berisi beberapa field. |
| C006 | Menekan tombol Batal pada dialog Tambah Siswa | Sistem menutup dialog Tambah Siswa |
| C007 | Mengisi field NIS "123", Nama "Ali", Alamat "Yogyakarta", No. Hp "123", Kelas "5", Tahun Ajaran "2022/2023" dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Siswa, Berhasil Menambah Siswa” kemudian tampil halaman Siswa |
| C008 | Mengisi field NIS " ", Nama "Ali", Alamat "Yogyakarta", No. Hp "123", Kelas "5", Tahun Ajaran "2022/2023" kemudian mengosongkan salah satu field, dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Siswa, Gagal Menambah Siswa”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. |
| C009 | Mengisi field NIS dengan string ‘’satu dua tiga” dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Siswa, Gagal Menambah Siswa”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. |
| C010 | Mengisi field No. Hp dengan string ‘’satu dua tiga” dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Siswa, Gagal Menambah Siswa”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. |
| C011 | Mengisi field NIS sama dengan NIS sudah terdaftar dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Siswa, NIS sudah teraftar, Gagal Menambah Siswa”. |
| C012 | Menekan tombol Edit Siswa | Sistem menampilkan dialog Edit Siswa yang berisi beberapa field. Isi dari field sesuai dengan data yang ingin di edit. |
| C013 | Menekan tombol Batal pada dialog Edit Siswa | Sistem menutup dialog Edit Siswa |
| C014 | Mengubah isi dari field dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Siswa, Berhasil Mengupdate Siswa” Kemudian tampil halaman Siswa. |
| C015 | Mengosongkan isi field dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Siswa, Gagal Mengupdate Siswa”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. |
| C016 | Mengubah isi field dan menekan tombol Batal | Sistem menutup dialog Edit Siswa, data tidak berubah, dan menampilkan halaman Siswa. |
| C017 | Menekan tombol Hapus | Sistem menampilkan dialog Hapus Data. |
| C018 | Menekan tombol Batal pada dialog Hapus Siswa | Sistem menutup dialog Hapus Siswa | |
| C019 | Menekan tombol Hapus pada dialog Hapus Siswa | Sistem menghapus Data Siswa sesuai dengan yang dipilih, sistem menutup dialog Hapus Siswa, dan menampilkan halaman Siswa. | |

1. Kelas

**Tabel 4.** Rancangan Test Case Kelas

|  |  |  |
| --- | --- | --- |
| **Id** | **Deskripsi Pengujian** | **Hasil Yang Diharapkan** |
| D001 | Mengisi field pencarian dengan Kelas | Sistem menampilkan siswa berdasarkan Kelas yang diinputkan. |
| D002 | Mengisi field pencarian dengan Wali Kelas | Sistem menampilkan siswa berdasarkan Wali Kelas yang diinputkan. |
| D003 | Mengisi field pencarian dengan selain Kelas/Wali Kelas | Sistem menampilkan tabel kosong. |
| D004 | Menekan tombol Tambah Kelas | Sistem menampilkan dialog Tambah Kelas yang berisi beberapa field. |
| D005 | Menekan tombol Batal pada dialog Tambah Kelas | Sistem menutup dialog Tambah Kelas |
| D006 | Mengisi field Kelas "5", Jumlah Siswa "30", Wali Kelas "Dodoro", Tahun Ajaran "2022/2023" dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Kelas, Berhasil Menambah Kelas” kemudian tampil halaman Kelas |
| D007 | Mengisi field Kelas "5", Jumlah Siswa " ", Wali Kelas "Dodoro", Tahun Ajaran "2022/2023" kemudian mengosongkan salah satu field, dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Kelas, Gagal Menambah Kelas”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. |
| D008 | Mengisi field Jumlah Kelas dengan string ‘’satu dua tiga” dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Kelas, Gagal Menambah Kelas”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. |
| D009 | Mengisi field data Kelas yang sudah ada dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Kelas, Gagal Menambah Data!!! Kelas sudah terdaftar”. |
| D010 | Mengisi field Kelas sama dengan Kelas sudah terdaftar dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Kelas, Kelas sudah terdaftar, Gagal Menambah Kelas”. |
| D011 | Menekan tombol Edit Kelas | Sistem menampilkan dialog Edit Kelas yang berisi beberapa field. Isi dari field sesuai dengan data yang ingin di edit. |
| D012 | Menekan tombol Batal pada dialog Edit Kelas | Sistem menutup dialog Edit Kelas |
| D013 | Mengubah isi dari field dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Kelas, Berhasil Mengupdate Kelas” Kemudian tampil halaman Kelas. |
| D014 | Mengosongkan isi field dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Kelas, Gagal Mengupdate Kelas”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. |
| D015 | Mengubah isi field dan menekan tombol Batal | Sistem menutup dialog Edit Kelas, data tidak berubah, dan menampilkan halaman Kelas. |
| D016 | Menekan tombol Hapus | Sistem menampilkan dialog Hapus Data. |
| D017 | Menekan tombol Batal pada dialog Hapus Kelas | Sistem menutup dialog Hapus Kelas |
| D018 | Menekan tombol Hapus pada dialog Hapus Kelas | Sistem menghapus Data Kelas sesuai dengan yang dipilih, sistem menutup dialog Hapus Kelas, dan menampilkan halaman Kelas. |

1. Mata Pelajaran

**Tabel 4.** Rancangan Test Case Mata Pelajaran

|  |  |  |
| --- | --- | --- |
| **Id** | **Deskripsi Pengujian** | **Hasil Yang Diharapkan** |
| E001 | Mengisi field pencarian dengan Mata Pelajaran | Sistem menampilkan Mata Pelajaran berdasarkan Mata Pelajaran yang diinputkan. |
| E002 | Mengisi field pencarian dengan selain Mata Pelajaran | Sistem menampilkan tabel kosong. |
| E003 | Menekan tombol Tambah Mata Pelajaran | Sistem menampilkan dialog Tambah Mata Pelajaran yang berisi beberapa field. |
| E004 | Menekan tombol Batal pada dialog Tambah Mata Pelajaran | Sistem menutup dialog Tambah Mata Pelajaran |
| E005 | Mengisi field Mata Pelajaran “IPA ”dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Mata Pelajaran, Berhasil Menambah Mata Pelajaran” kemudian tampil halaman Mata Pelajaran |
| E006 | Mengosongkan field Mata Pelajaran dan menekan tombol simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Mata Pelajaran, Gagal Menambah Mata Pelajaran”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. |

|  |  |  |
| --- | --- | --- |
| E007 | Mengisi field Mata Pelajaran sama dengan Mata Pelajaran sudah terdaftar dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Mata Pelajaran, Kelas sudah terdaftar, Gagal Menambah Mata Pelajaran”. |
| E008 | Menekan tombol Edit Mata Pelajaran | Sistem menampilkan dialog Edit Mata Pelajaran yang berisi beberapa field. Isi dari field sesuai dengan data yang ingin di edit. |
| E009 | Menekan tombol Batal pada dialog Edit Mata Pelajaran | Sistem menutup dialog Edit Mata Pelajaran |
| E010 | Mengubah isi dari field dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Mata Pelajaran, Berhasil Mengupdate Mata Pelajaran” Kemudian tampil halaman Mata Pelajaran. |
| E011 | Mengosongkan isi field dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Mata Pelajaran, Gagal Mengupdate Mata Pelajaran”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. |
| E012 | Mengubah isi field dan menekan tombol Batal | Sistem menutup dialog Edit Mata Pelajaran, data tidak berubah, dan menampilkan halaman Mata Pelajaran. |
| E013 | Menekan tombol Hapus | Sistem menampilkan dialog Hapus Data. |
| E014 | Menekan tombol Batal pada dialog Hapus Mata Pelajaran | Sistem menutup dialog Hapus Mata Pelajaran |
| E015 | Menekan tombol Hapus pada dialog Hapus Mata Pelajaran | Sistem menghapus Data Mata Pelajaran sesuai dengan yang dipilih, sistem menutup dialog Hapus Mata Pelajaran, dan menampilkan halaman Mata Pelajaran. |

1. Tema

**Tabel 4.** Rancangan Test Case Tema

|  |  |  |
| --- | --- | --- |
| **Id** | **Deskripsi Pengujian** | **Hasil Yang Diharapkan** |
| F001 | Mengisi field pencarian dengan Mata Pelajaran | Sistem menampilkan Tema berdasarkan Mata Pelajaran yang diinputkan. |
| F002 | Mengisi field pencarian dengan Tema | Sistem menampilkan Tema berdasarkan Tema yang diinputkan. |
| F003 | Mengisi field pencarian dengan selain Mata Pelajaran/Tema | Sistem menampilkan tabel kosong. |
| F004 | Menekan tombol Tambah Tema | Sistem menampilkan dialog Tambah Tema yang berisi beberapa field. |
| F005 | Menekan tombol Batal pada dialog Tambah Tema | Sistem menutup dialog Tambah Tema |
| F006 | Mengisi field Mata Pelajaran "IPA", Tema "Tema 1 IPA", Sub Tema "Sub Tema 1 IPA" dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Tema, Berhasil Menambah Tema” kemudian tampil halaman Tema |
| F007 | Mengisi field Mata Pelajaran "IPA", Tema "Tema 1 IPA", Sub Tema "Sub Tema 1 IPA" kemudian mengosongkan salah satu field, dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Tema, Gagal Menambah Tema”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. |
| F008 | Mengisi field Tema sama dengan Tema sudah terdaftar dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Tema, Tema sudah terdaftar, Gagal Menambah Tema”. |
| F009 | Mengisi field Sub Tema sama dengan Sub Tema sudah terdaftar dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Sub Tema, Sub Tema sudah terdaftar, Gagal Menambah Sub Tema”. |
| F010 | Menekan tombol Edit Tema | Sistem menampilkan dialog Edit Tema yang berisi beberapa field. Isi dari field sesuai dengan data yang ingin di edit. |
| F011 | Menekan tombol Batal pada dialog Edit Tema | Sistem menutup dialog Edit Tema |
| F012 | Mengubah isi dari field dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Tema, Berhasil Mengupdate Tema” Kemudian tampil halaman Tema. |
| F013 | Mengosongkan isi field dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Tema, Gagal Mengupdate Tema”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. |
| F014 | Mengubah isi field dan menekan tombol Batal | Sistem menutup dialog Edit Tema, data tidak berubah, dan menampilkan halaman Tema. |
| F015 | Menekan tombol Hapus | Sistem menampilkan dialog Hapus Data. |
| F016 | Menekan tombol Batal pada dialog Hapus Tema | Sistem menutup dialog Hapus Tema |
| F017 | Menekan tombol Hapus pada dialog Hapus Tema | Sistem menghapus Data Tema sesuai dengan yang dipilih, sistem menutup dialog Hapus Tema, dan menampilkan halaman Tema. |

## **4.12 Hasil Pengujian Equivalence Partitioning**

Tujuan dari pengujian pada Sistem Informasi Pengolahan Nilai Harian SDN 1 Warga Mulya Berbasis Web adalah untuk menemukan kekurangan dan kelemahan pada sistem sebelum digunakan oleh pengguna. Jika ada kelemahan yang ditemukan, maka pengguna bisa mengembangkan sistem pada bagian yang dianggap lemah. Dalam pengujian, digunakan tabel test case yang berfungsi untuk menyimpulkan apakah sistem berhasil atau tidak. Metode yang digunakan dalam pengujian adalah Black Box berbasis Equivalence Partitioning. Metode ini digunakan untuk memastikan bahwa setiap bagian atau kelompok pada sistem telah diuji dengan benar dan sama-sama mendapatkan perlakuan yang adil dalam pengujian. Berikut adalah Hasil Pengujian Equivalence Partitioning.

### **4.11.1 Guru**

**Tabel 4.** Hasil Pengujian Equivalence Partitioning Guru

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **Deskripsi Pengujian** | **Hasil Yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| A001 | Mengisi field username dan password yang benar, lalu menekan tombol Login | Sistem menerima dan menampilkan Pemberitahuan “Login Berhasil, Login Sebagai Guru” Kemudian tampil halaman dashboard | Sistem menampilkan  pemberitahuan  “Login Berhasil, Login Sebagai Guru”  Kemudian tampil  halaman dashboard | Berhasil |
| A002 | Mengisi field username dan password yang salah, lalu menekan tombol Login | Sistem menolak dan menampilkan Pemberitahuan “Login Gagal, Mohon Cek Kembali Username dan Password Anda” | Sistem menampilkan  pemberitahuan  “Login Gagal, Mohon Cek Kembali Username dan Password Anda” | Berhasil |
| A003 | Mengosongkan salah satu field username dan password dan menekan tombol Login | Sistem menolak dan menampilkan Pemberitahuan “Login Gagal, Mohon Cek Kembali Username dan Password Anda”. Serta pada field menampilkan pemberitahuan “Tolong masukkan Username/Password terlebih dahulu.” | Sistem menampilkan  pemberitahuan  “Login Gagal, Mohon Cek Kembali Username dan Password Anda”. Field menampilkan pemberitahuan “Tolong masukkan Username/Password terlebih dahulu.” | Berhasil |
| A004 | Mengosongkan field username dan password dan menekan tombol Login | Sistem menolak dan menampilkan Pemberitahuan “Login Gagal, Mohon Cek Kembali Username dan Password Anda”. Serta pada field menampilkan pemberitahuan “Tolong masukkan Username/Password terlebih dahulu.” | Sistem menampilkan  pemberitahuan  “Login Gagal, Mohon Cek Kembali Username dan Password Anda”. Field menampilkan pemberitahuan “Tolong masukkan Username/Password terlebih dahulu.” | Berhasil |
| B001 | Melakukan pengecekan apakah data yang ditampilkan sesuai dengan data guru yang login ke dalam aplikasi | Sistem menampilkan data guru sesuai dengan guru yang mengakses aplikasi saat itu. | Sistem menampilkan data guru sesuai dengan guru yang login ke dalam aplikasi | Berhasil |
| C001 | Mengisi field pencarian dengan NIS | Sistem menampilkan siswa berdasarkan NIS yang diinputkan. | Sistem menampilkan siswa berdasarkan NIS. | Berhasil |
| C002 | Mengisi field pencarian dengan Nama | Sistem menampilkan siswa berdasarkan Nama yang diinputkan. | Sistem menampilkan siswa berdasarkan Nama. | Berhasil |
| C003 | Mengisi field pencarian dengan Kelas | Sistem menampilkan siswa berdasarkan Kelas yang diinputkan. | Sistem menampilkan siswa berdasarkan Kelas. | Berhasil |
| C004 | Mengisi field pencarian dengan selain NIS/Nama/Kelas | Sistem menampilkan tabel kosong. | Sistem menampilkan tabel kosong. | Berhasil |
| D001 | Mengisi field pencarian dengan NIS | Sistem menampilkan nilai berdasarkan NIS yang diinputkan. | Sistem menampilkan nilai berdasarkan NIS. | Berhasil |
| D002 | Mengisi field pencarian dengan Nama | Sistem menampilkan nilai berdasarkan Nama yang diinputkan. | Sistem menampilkan nilai berdasarkan Nama. | Berhasil |
| D003 | Mengisi field pencarian dengan Mata Pelajaran | Sistem menampilkan nilai berdasarkan Mata Pelajaran yang diinputkan. | Sistem menampilkan nilai berdasarkan Mata Pelajaran. | Berhasil |
| D004 | Mengisi field pencarian dengan selain NIS/Nama/Mata Pelajaran | Sistem menampilkan tabel kosong. | Sistem menampilkan tabel kosong. | Berhasil |
| D005 | Menekan tombol Tambah Nilai | Sistem menampilkan dialog Tambah Nilai yang berisi beberapa field. | Sistem menampilkan dialog Tambah Nilai yang berisi beberapa field. | Berhasil |
| D006 | Menekan tombol Batal pada dialog Tambah Nilai | Sistem menutup dialog Tambah Nilai | Sistem menutup dialog Tambah Nilai | Berhasil |
| D007 | Mengisi field NIS "1000", Nama "Budi", Nilai "80", Mata Pelajaran "IPA", Tema "Tema 1 IPA", Subtema "Subtema 1 IPA" dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Nilai, Berhasil Menambah Nilai” Kemudian tampil halaman Nilai | Sistem menampilkan pemberitahuan “Data Nilai, Berhasil Menambah Nilai”, Kemudian tampil halaman Nilai | Berhasil |
| D008 | Mengisi field NIS "1000", Nama "Budi", Nilai "80", Mata Pelajaran "IPA", Tema "Tema 1 IPA", Subtema " ", kemudian mengosongkan salah satu field, dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Nilai, Gagal Menambah Nilai”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan pemberitahuan “Data Nilai, Gagal Menambah Nilai”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Berhasil |
| D009 | Mengosongkan semua field dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Nilai, Gagal Menambah Nilai”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan pemberitahuan “Data Nilai, Gagal Menambah Nilai”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Berhasil |
| D010 | Mengisi field Nis dengan string ‘’Seribu” dan menekan Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Nilai, Gagal Menambah Nilai”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan pemberitahuan “Data Nilai, Gagal Menambah Nilai”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Berhasil |
| D0011 | Mengisi field Nilai 60 dimana kurang dari 65 (KKM) dan menekan Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Nilai, Berhasil Menambah Nilai”, tetapi muncul pemberitahuan “Nilai yang dimasukkan (60) kurang dari KKM (65)`, 'Masuk Kategori Remidial” dan kemudian tampil halaman Nilai, pada halaman nilai akan berwarna merah. | Sistem menampilkan pemberitahuan “Data Nilai, Berhasil Menambah Nilai”, “Nilai yang dimasukkan (60) kurang dari KKM (65)`, 'Masuk Kategori Remidial” dan pada halaman nilai akan berwarna merah. | Berhasil |
| D012 | Mengisi field Nilai 80 dimana lebih dari 65 (KKM) dan menekan Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Nilai, Berhasil Menambah Nilai” Kemudian tampil halaman Nilai | Sistem menampilkan pemberitahuan “Data Nilai, Berhasil Menambah Nilai”, Kemudian tampil halaman Nilai | Berhasil |
| D013 | Mengisi field Nilai dengan string ‘’Delapan Puluh” dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Nilai, Gagal Menambah Nilai”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan pemberitahuan “Data Nilai, Gagal Menambah Nilai”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Berhasil |
| D014 | Mengisi field nilai kurang dari 0 (-50) dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Nilai, Gagal Menambah Nilai”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan pemberitahuan “Data Nilai, Gagal Menambah Nilai”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Berhasil |
| D015 | Mengisi field nilai lebih dari 100 (101) dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Nilai, Gagal Menambah Nilai”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan pemberitahuan “Data Nilai, Gagal Menambah Nilai”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Berhasil |
| D016 | Menekan tombol Edit | Sistem menampilkan dialog Edit Nilai yang berisi beberapa field. Isi dari field sesuai dengan data yang ingin di edit. | Sistem menampilkan dialog edit. | Berhasil |
| D017 | Mengisi field Mata Pelajaran, Tema, Sub Tema sama dengan data sudah terdaftar dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Nilai, Data sudah terdaftar, Gagal Menambah Nilai”. | Sistem menampilkan pemberitahuan “Data Nilai, Berhasil Menambah Nilai”, Kemudian tampil halaman Nilai | Berhasil |
| D018 | Menekan tombol Batal pada dialog Edit Nilai | Sistem menutup dialog Edit Nilai | Sistem menutup dialog Edit Nilai | Berhasil |
| D019 | Mengubah isi dari field dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Nilai, Berhasil Mengupdate Nilai” Kemudian tampil halaman Nilai | Sistem menampilkan pemberitahuan “Data Nilai, Berhasil Menambah Nilai”, Kemudian tampil halaman Nilai | Berhasil |
| D020 | Mengosongkan isi field dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Nilai, Gagal Mengupdate Nilai”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan pemberitahuan “Data Nilai, Gagal Menambah Nilai”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Berhasil |
| D021 | Mengubah NIS “1000”, menjadi “Seribu” dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Nilai, Gagal Mengupdate Nilai”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan pemberitahuan “Data Nilai, Gagal Menambah Nilai”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Berhasil |
| D022 | Mengubah Nilai “75”, menjadi “50” dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Nilai, Berhasil Menambah Nilai”, tetapi muncul pemberitahuan “Nilai yang dimasukkan (50) kurang dari KKM (65)`, 'Masuk Kategori Remidial” dan kemudian tampil halaman Nilai, pada halaman nilai akan berwarna merah. | Sistem menampilkan pemberitahuan “Data Nilai, Gagal Menambah Nilai”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Berhasil |
| D023 | Mengubah Nilai “75”, menjadi “-50” dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Nilai, Gagal Mengupdate Nilai”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan pemberitahuan “Data Nilai, Gagal Menambah Nilai”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Berhasil |
| D024 | Mengubah Nilai “75”, menjadi “101” dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Nilai, Gagal Mengupdate Nilai”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan pemberitahuan “Data Nilai, Gagal Menambah Nilai”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Berhasil |
| D025 | Mengubah Nilai “75”, menjadi “Tujuh Puluh Lima” dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Nilai, Gagal Mengupdate Nilai”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan pemberitahuan “Data Nilai, Gagal Menambah Nilai”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Berhasil |
| D026 | Mengubah isi field dan menekan tombol Batal | Sistem menutup dialog Edit Nilai, data tidak berubah, dan menampilkan halaman Nilai. | Sistem menutup dialog Edit Nilai dan menampilkan halaman Nilai. | Berhasil |
| D027 | Menekan tombol Hapus | Sistem menampilkan dialog Hapus Data. | Sistem menampilkan dialog Hapus Data | Berhasil |
| D028 | Menekan tombol Batal pada dialog Hapus Nilai | Sistem menutup dialog Hapus Nilai | Sistem menutup dialog Hapus Data dan menampilkan halaman nilai. | Berhasil |
| D029 | Menekan tombol Hapus pada dialog Hapus Nilai | Sistem menghapus Data Nilai sesuai dengan yang dipilih, sistem menutup dialog Hapus Nilai, dan menampilkan halaman Nilai. | Sistem menampilkan pemberitahuan “Data Nilai, Berhasil Menghapus Nilai”, Kemudian tampil halaman Nilai | Berhasil |
| D030 | Menekan tombol Cetak ke PDF | Sistem secara otomatis akan mengkonversi data tabel ke dalam bentuk PDF dan langsung terunduh. | Sistem otomatis mengunduh data PDF | Berhasil |
| D031 | Menekan tombol Cetak ke Excel | Sistem secara otomatis akan mengkonversi data tabel ke dalam bentuk Excel dan langsung terunduh. | Sistem otomatis mengunduh data Excel | Berhasil |
| E001 | Mengisi field pencarian dengan Kelas | Sistem menampilkan siswa berdasarkan Kelas yang diinputkan. | Sistem menampilkan siswa berdasarkan Kelas. | Berhasil |
| E002 | Mengisi field pencarian dengan Nama | Sistem menampilkan siswa berdasarkan Nama yang diinputkan. | Sistem menampilkan siswa berdasarkan Nama. | Berhasil |
| E003 | Mengisi field pencarian dengan selain NIS/Nama/Mata Pelajaran | Sistem menampilkan tabel kosong. | Sistem menampilkan tabel kosong. | Berhasil |
| E004 | Menekan tombol Tambah KKM | Sistem menampilkan dialog Tambah KKM yang berisi beberapa field. | Sistem menampilkan dialog Tambah KKM | Berhasil |
| E005 | Menekan tombol Batal pada dialog Tambah KKM | Sistem menutup dialog Tambah KKM | Sistem menutup dialog Tambah KKM dan menampilkan halaman KKM. | Berhasil |
| E006 | Mengisi field Kelas "5", Mata Pelajaran "IPA", Tema "Tema 1 IPA", Subtema "Subtema 1 IPA", KKM "65"dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data KKM, Berhasil Menambah KKM” kemudian tampil halaman KKM | Sistem menampilkan pemberitahuan “Data KKM, Berhasil Menambah KKM” kemudian tampil halaman KKM | Berhasil |
| E007 | Mengisi field Kelas "5", Mata Pelajaran "IPA", Tema "Tema 1 IPA", Subtema "Subtema 1 IPA", KKM " "kemudian mengosongkan salah satu field, dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data KKM, Gagal Menambah KKM”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan Pemberitahuan “Data KKM, Gagal Menambah KKM”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Berhasil |
| E008 | Mengisi field KKM dengan string ‘’Delapan Puluh” dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data KKM, Gagal Menambah KKM”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan Pemberitahuan “Data KKM, Gagal Menambah KKM”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Berhasil |
| E009 | Mengisi field nilai kurang dari 0 (-50) dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data KKM, Gagal Menambah KKM”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan Pemberitahuan “Data KKM, Gagal Menambah KKM”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Berhasil |
| E010 | Mengisi field nilai lebih dari 100 (101) dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data KKM, Gagal Menambah KKM”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan Pemberitahuan “Data KKM, Gagal Menambah KKM”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Berhasil |
| E011 | Menekan tombol Edit KKM | Sistem menampilkan dialog Edit KKM yang berisi beberapa field. Isi dari field sesuai dengan data yang ingin di edit. | Sistem menampilkan dialog Edit KKM | Berhasil |
| E012 | Mengisi field Mata Pelajaran, Tema, Sub Tema sama dengan data sudah terdaftar dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data KKM, Data sudah terdaftar, Gagal Menambah KKM”. | Sistem menampilkan Pemberitahuan “Data KKM, Data sudah terdaftar, Gagal Menambah KKM”. | Berhasil |
| E013 | Menekan tombol Batal pada dialog Edit KKM | Sistem menutup dialog Edit KKM | Sistem menutup dialog Edit KKM | Berhasil |
| E014 | Mengubah isi dari field dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data KKM, Berhasil Mengupdate KKM” Kemudian tampil halaman KKM. | Sistem menampilkan Pemberitahuan “Data KKM, Berhasil Mengupdate KKM” Kemudian tampil halaman KKM. | Berhasil |
| E015 | Mengosongkan isi field dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data KKM, Gagal Mengupdate KKM”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan Pemberitahuan “Data KKM, Gagal Mengupdate KKM”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Berhasil |
| E016 | Mengubah isi field dan menekan tombol Batal | Sistem menutup dialog Edit Nilai, data tidak berubah, dan menampilkan halaman Nilai. | Sistem menutup dialog Edit KKM | Berhasil |
| E017 | Menekan tombol Hapus | Sistem menampilkan dialog Hapus Data. | Sistem menampilkan dialog Hapus Data. | Berhasil |
| E018 | Menekan tombol Batal pada dialog Hapus Nilai | Sistem menutup dialog Hapus Nilai | Sistem menutup dialog Hapus Nilai | Berhasil |
| E019 | Menekan tombol Hapus pada dialog Hapus Nilai | Sistem menghapus Data Nilai sesuai dengan yang dipilih, sistem menutup dialog Hapus Nilai, dan menampilkan halaman Nilai. | Sistem menampilkan Pemberitahuan “Data KKM, Berhasil Menghapus KKM”. | Berhasil |
| F001 | Mengisi field pencarian dengan NIS | Sistem menampilkan siswa berdasarkan NIS yang diinputkan. | Sistem menampilkan siswa berdasarkan NIS. | Berhasil |
| F002 | Mengisi field pencarian dengan Nama | Sistem menampilkan siswa berdasarkan Nama yang diinputkan. | Sistem menampilkan siswa berdasarkan Nama. | Berhasil |
| F003 | Melakukan cek pada tabel Laporan | Terdapat data nilai siswa yang sudah dalam bentuk nilai rata-rata per tema(T = Tema) dan per mata pelajaran. | Sistem menampilkan nilai per mata pelajaran dan per tema pada siswa | Berhasil |
| F004 | Melakukan cek pada data nilai | Nilai sudah terbagi per mata pelajaran dan per tema, dimana dalam satu tema terdapat 4 sub tema. Jadi nilai per tema tersebut adalah total nilai 4 sub tema dibagi jumlah sub tema. | Sistem sudah menjumlahkan nilai rata-rata per tema(T = Tema) dan per mata pelajaran. | Berhasil |
| F005 | Melakukan cek pada nilai rata-rata | Nilai yang sudah berada diatas KKM akan berwarna hitam dan jika dibawah KKM atau nilai belum diinputkan maka akan berwarna merah. | Sistem otomatis mengubah warna nilai jika kurang dari kkm | Berhasil |
| F006 | Menekan tombol Cetak ke PDF | Sistem secara otomatis akan mengkonversi data tabel ke dalam bentuk PDF dan langsung terunduh. | Sistem otomatis mengunduh data PDF | Berhasil |
| F007 | Menekan tombol Cetak ke Excel | Sistem secara otomatis akan mengkonversi data tabel ke dalam bentuk Excel dan langsung terunduh. | Sistem otomatis mengunduh data PDF | Berhasil |
| Totak task: **66** | | | | |

### **4.11.2. Admin**

**Tabel 4.** Hasil Pengujian Equivalence Partitioning Admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **Deskripsi Pengujian** | **Hasil Yang Diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| A001 | Mengisi field username dan password yang benar, lalu menekan tombol Login | Sistem menerima dan menampilkan Pemberitahuan “Login Berhasil, Login Sebagai Admin” Kemudian tampil halaman dashboard | Sistem menampilkan  pemberitahuan  “Login Berhasil, Login Sebagai Admin”  Kemudian tampil  halaman dashboard | Berhasil |
| A002 | Mengisi field username dan password yang salah, lalu menekan tombol Login | Sistem menolak dan menampilkan Pemberitahuan “Login Gagal, Mohon Cek Kembali Username dan Password Anda” | Sistem menampilkan  pemberitahuan  “Login Gagal, Mohon Cek Kembali Username dan Password Anda” | Berhasil |
| A003 | Mengosongkan salah satu field username dan password dan menekan tombol Login | Sistem menolak dan menampilkan Pemberitahuan “Login Gagal, Mohon Cek Kembali Username dan Password Anda”. Serta pada field menampilkan pemberitahuan “Tolong masukkan Username/Password terlebih dahulu.” | Sistem menampilkan  pemberitahuan  “Login Gagal, Mohon Cek Kembali Username dan Password Anda”. Field menampilkan pemberitahuan “Tolong masukkan Username/Password terlebih dahulu.” | Berhasil |
| A004 | Mengosongkan field username dan password dan menekan tombol Login | Sistem menolak dan menampilkan Pemberitahuan “Login Gagal, Mohon Cek Kembali Username dan Password Anda”. Serta pada field menampilkan pemberitahuan “Tolong masukkan Username/Password terlebih dahulu.” | Sistem menampilkan  pemberitahuan  “Login Gagal, Mohon Cek Kembali Username dan Password Anda”. Field menampilkan pemberitahuan “Tolong masukkan Username/Password terlebih dahulu.” | Berhasil |
| B001 | Mengisi field pencarian dengan NIP | Sistem menampilkan guru berdasarkan NIP yang diinputkan. | Sistem menampilkan guru berdasarkan NIP. | Berhasil |
| B002 | Mengisi field pencarian dengan Nama | Sistem menampilkan guru berdasarkan Nama yang diinputkan. | Sistem menampilkan guru berdasarkan NIP. | Berhasil |
| B003 | Mengisi field pencarian dengan selain NIS/Nama | Sistem menampilkan tabel kosong. | Sistem menampilkan tabel kosong. | Berhasil |
| B004 | Menekan tombol Tambah Guru | Sistem menampilkan dialog Tambah Guru yang berisi beberapa field. | Sistem menampilkan dialog Tambah Guru | Berhasil |
| B005 | Menekan tombol Batal pada dialog Tambah Guru | Sistem menutup dialog Tambah Guru | Sistem menutup dialog Tambah Guru | Berhasil |
| B006 | Mengisi field NIP "123", Nama "Samsudin", Username"sms007", Password "sms777", Alamat "Yogyakarta", Jabatan "Guru", No. Hp "123", Status "Aktif" dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Guru, Berhasil Menambah Guru” kemudian tampil halaman Guru | Sistem menampilkan Pemberitahuan “Data Guru, Berhasil Menambah Guru” kemudian tampil halaman Guru | Berhasil |
| B007 | Mengisi field NIP " ", Nama "Samsudin", Username"sms007", Password "sms777", Alamat "Yogyakarta", Jabatan "Guru", No. Hp "123", Status "Aktif" kemudian mengosongkan salah satu field, dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Guru, Gagal Menambah Guru”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan Pemberitahuan “Data Guru, Gagal Menambah Guru”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Berhasil |
| B008 | Mengisi field NIP dengan string ‘’satu dua tiga” dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Guru, Gagal Menambah Guru”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan Pemberitahuan “Data Guru, Gagal Menambah Guru”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Berhasil |
| B009 | Mengisi field No. Hp dengan string ‘’satu dua tiga” dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Guru, Gagal Menambah Guru”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan Pemberitahuan “Data Guru, Gagal Menambah Guru”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Berhasil |
| B010 | Mengisi field NIP sama dengan NIP sudah terdaftar dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Guru, NIP sudah teraftar, Gagal Menambah Guru”. | Sistem menampilkan Pemberitahuan “Data Guru, NIP sudah teraftar, Gagal Menambah Guru”. | Berhasil |
| B011 | Menekan tombol Edit Guru | Sistem menampilkan dialog Edit Guru yang berisi beberapa field. Isi dari field sesuai dengan data yang ingin di edit. | Sistem menampilkan dialog Edit Guru | Berhasil |
| B012 | Menekan tombol Batal pada dialog Edit Guru | Sistem menutup dialog Edit Guru | Sistem menutup dialog Edit Guru | Berhasil |
| B013 | Mengubah isi dari field dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Guru, Berhasil Mengupdate Guru” Kemudian tampil halaman Guru. | Sistem menampilkan Pemberitahuan “Data Guru, Berhasil Mengupdate Guru” Kemudian tampil halaman Guru. | Berhasil |
| B014 | Mengosongkan isi field dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Guru, Gagal Mengupdate Guru”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan Pemberitahuan “Data Guru, Gagal Mengupdate Guru”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Berhasil |
| B015 | Mengubah isi field dan menekan tombol Batal | Sistem menutup dialog Edit Guru, data tidak berubah, dan menampilkan halaman Guru. | Sistem menutup dialog Edit Guru. | Berhasil |
| B016 | Menekan tombol Hapus | Sistem menampilkan dialog Hapus Data. | Sistem menampilkan dialog Hapus Data. | Berhasil |
| B017 | Menekan tombol Batal pada dialog Hapus Nilai | Sistem menutup dialog Hapus Guru | Sistem menampilkan dialog Hapus Data. | Berhasil |
| B018 | Menekan tombol Hapus pada dialog Hapus Nilai | Sistem menghapus Data Guru sesuai dengan yang dipilih, sistem menutup dialog Hapus Guru, menampilkan pemberitahuan “Data Guru”,”Berhasil Menghapus Data Guru!!!”, dan menampilkan halaman Guru. | Sistem menghapus Data Guru sesuai dengan yang dipilih, sistem menutup dialog Hapus Guru, menampilkan pemberitahuan “Data Guru”,”Berhasil Menghapus Data Guru!!!”, dan menampilkan halaman Guru. | Berhasil |
| C001 | Mengisi field pencarian dengan NIS | Sistem menampilkan siswa berdasarkan NIS yang diinputkan. | Sistem menampilkan siswa berdasarkan NIS | Berhasil |
| C002 | Mengisi field pencarian dengan Nama | Sistem menampilkan siswa berdasarkan Nama yang diinputkan. | Sistem menampilkan siswa berdasarkan Nama | Berhasil |
| C003 | Mengisi field pencarian dengan Kelas | Sistem menampilkan siswa berdasarkan Kelas yang diinputkan. | Sistem menampilkan siswa berdasarkan Kelas | Berhasil |
| C004 | Mengisi field pencarian dengan selain NIS/Nama/Kelas | Sistem menampilkan tabel kosong. | Sistem menampilkan tabel kosong. | Berhasil |
| C005 | Menekan tombol Tambah Siswa | Sistem menampilkan dialog Tambah Siswa yang berisi beberapa field. | Sistem menampilkan dialog Tambah Siswa | Berhasil |
| C006 | Menekan tombol Batal pada dialog Tambah Siswa | Sistem menutup dialog Tambah Siswa | Sistem menutup dialog Tambah Siswa | Berhasil |
| C007 | Mengisi field NIS "123", Nama "Ali", Alamat "Yogyakarta", No. Hp "123", Kelas "5", Tahun Ajaran "2022/2023" dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Siswa, Berhasil Menambah Siswa” kemudian tampil halaman Siswa | Sistem menerima dan menampilkan Pemberitahuan “Data Siswa, Berhasil Menambah Siswa” kemudian tampil halaman Siswa | Berhasil |
| C008 | Mengisi field NIS " ", Nama "Ali", Alamat "Yogyakarta", No. Hp "123", Kelas "5", Tahun Ajaran "2022/2023" kemudian mengosongkan salah satu field, dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Siswa, Gagal Menambah Siswa”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan Pemberitahuan “Data Siswa, Gagal Menambah Siswa”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Berhasil |
| C009 | Mengisi field NIS dengan string ‘’satu dua tiga” dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Siswa, Gagal Menambah Siswa”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan Pemberitahuan “Data Siswa, Gagal Menambah Siswa”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Berhasil |
| C010 | Mengisi field No. Hp dengan string ‘’satu dua tiga” dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Siswa, Gagal Menambah Siswa”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan Pemberitahuan “Data Siswa, Gagal Menambah Siswa”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Berhasil |
| C011 | Mengisi field NIS sama dengan NIS sudah terdaftar dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Siswa, NIS sudah teraftar, Gagal Menambah Siswa”. | Sistem menolak dan menampilkan Pemberitahuan “Data Siswa, NIS sudah teraftar, Gagal Menambah Siswa”. | Berhasil |
| C012 | Menekan tombol Edit Siswa | Sistem menampilkan dialog Edit Siswa yang berisi beberapa field. Isi dari field sesuai dengan data yang ingin di edit. | Sistem menampilkan dialog Edit Siswa | Berhasil |
| C013 | Menekan tombol Batal pada dialog Edit Siswa | Sistem menutup dialog Edit Siswa | Sistem menutup dialog Edit Siswa | Berhasil |
| C014 | Mengubah isi dari field dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Siswa, Berhasil Mengupdate Siswa” Kemudian tampil halaman Siswa. | Sistem menampilkan Pemberitahuan “Data Siswa, Berhasil Mengupdate Siswa” Kemudian tampil halaman Siswa. | Berhasil |
| C015 | Mengosongkan isi field dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Siswa, Gagal Mengupdate Siswa”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan Pemberitahuan “Data Siswa, Gagal Mengupdate Siswa”. Muncul pemberitahuan bahwa field harus diisi terlebih dahulu. | Berhasil |
| C016 | Mengubah isi field dan menekan tombol Batal | Sistem menutup dialog Edit Siswa, data tidak berubah, dan menampilkan halaman Siswa. | Sistem menutup dialog Edit Siswa | Berhasil |
| C017 | Menekan tombol Hapus | Sistem menampilkan dialog Hapus Data. |  |  |
| C018 | Menekan tombol Batal pada dialog Hapus Siswa | Sistem menutup dialog Hapus Siswa | Sistem menutup dialog Hapus Siswa | Berhasil |
| C019 | Menekan tombol Hapus pada dialog Hapus Siswa | Sistem menghapus Data Siswa sesuai dengan yang dipilih, sistem menutup dialog Hapus Siswa, dan menampilkan halaman Siswa. | Sistem menghapus Data Siswa sesuai dengan yang dipilih, sistem menutup dialog Hapus Siswa, dan menampilkan halaman Siswa. | Berhasil |
| D001 | Mengisi field pencarian dengan Kelas | Sistem menampilkan siswa berdasarkan Kelas yang diinputkan. | Sistem menampilkan siswa berdasarkan Kelas | Berhasil |
| D002 | Mengisi field pencarian dengan Wali Kelas | Sistem menampilkan siswa berdasarkan Wali Kelas yang diinputkan. | Sistem menampilkan siswa berdasarkan Wali Kelas | Berhasil |
| D003 | Mengisi field pencarian dengan selain Kelas/Wali Kelas | Sistem menampilkan tabel kosong. | Sistem menampilkan tabel kosong. | Berhasil |
| D004 | Menekan tombol Tambah Kelas | Sistem menampilkan dialog Tambah Kelas yang berisi beberapa field. | Sistem menampilkan dialog Tambah Kelas | Berhasil |
| D005 | Menekan tombol Batal pada dialog Tambah Kelas | Sistem menutup dialog Tambah Kelas | Sistem menutup dialog Tambah Kelas | Berhasil |
| D006 | Mengisi field Kelas "5", Jumlah Siswa "30", Wali Kelas "Dodoro", Tahun Ajaran "2022/2023" dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Kelas, Berhasil Menambah Kelas” kemudian tampil halaman Kelas | Sistem menampilkan Pemberitahuan “Data Kelas, Berhasil Menambah Kelas” kemudian tampil halaman Kelas | Berhasil |
| D007 | Mengisi field Kelas "5", Jumlah Siswa " ", Wali Kelas "Dodoro", Tahun Ajaran "2022/2023" kemudian mengosongkan salah satu field, dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Kelas, Gagal Menambah Kelas”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan Pemberitahuan “Data Kelas, Gagal Menambah Kelas”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Berhasil |
| D008 | Mengisi field Jumlah Kelas dengan string ‘’satu dua tiga” dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Kelas, Gagal Menambah Kelas”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan Pemberitahuan “Data Kelas, Gagal Menambah Kelas”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Berhasil |
| D009 | Mengisi field data Kelas yang sudah ada dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Kelas, Gagal Menambah Data!!! Kelas sudah terdaftar”. | Sistem menampilkan Pemberitahuan “Data Kelas, Gagal Menambah Data!!! Kelas sudah terdaftar”. | Berhasil |
| D010 | Menekan tombol Edit Kelas | Sistem menampilkan dialog Edit Kelas yang berisi beberapa field. Isi dari field sesuai dengan data yang ingin di edit. | Sistem menampilkan dialog Edit Kelas | Berhasil |
| D011 | Menekan tombol Batal pada dialog Edit Kelas | Sistem menutup dialog Edit Kelas | Sistem menutup dialog Edit Kelas | Berhasil |
| D012 | Mengubah isi dari field dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Kelas, Berhasil Mengupdate Kelas” Kemudian tampil halaman Kelas. | Sistem menampilkan Pemberitahuan “Data Kelas, Berhasil Mengupdate Kelas” Kemudian tampil halaman Kelas. | Berhasil |
| D013 | Mengosongkan isi field dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Kelas, Gagal Mengupdate Kelas”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan Pemberitahuan “Data Kelas, Gagal Mengupdate Kelas”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Berhasil |
| D014 | Mengubah isi field dan menekan tombol Batal | Sistem menutup dialog Edit Kelas, data tidak berubah, dan menampilkan halaman Kelas. | Sistem menutup dialog Edit Kelas | Berhasil |
| D015 | Menekan tombol Hapus | Sistem menampilkan dialog Hapus Data. | Sistem menampilkan dialog Hapus Data. | Berhasil |
| D016 | Menekan tombol Batal pada dialog Hapus Kelas | Sistem menutup dialog Hapus Kelas | Sistem menutup dialog Hapus Kelas | Berhasil |
| D017 | Menekan tombol Hapus pada dialog Hapus Kelas | Sistem menghapus Data Kelas sesuai dengan yang dipilih, sistem menutup dialog Hapus Kelas, dan menampilkan halaman Kelas. | Sistem menghapus Data Kelas sesuai dengan yang dipilih, muncul pemberitahuan “Data Kelas, Berhasil Menghapus Kelas”, sistem menutup dialog Hapus Kelas, dan menampilkan halaman Kelas. | Berhasil |
| E001 | Mengisi field pencarian dengan Mata Pelajaran | Sistem menampilkan Mata Pelajaran berdasarkan Mata Pelajaran yang diinputkan. | Sistem menampilkan Mata Pelajaran berdasarkan Mata Pelajaran | Berhasil |
| E002 | Mengisi field pencarian dengan selain Mata Pelajaran | Sistem menampilkan tabel kosong. | Sistem menampilkan tabel kosong. | Berhasil |
| E003 | Menekan tombol Tambah Mata Pelajaran | Sistem menampilkan dialog Tambah Mata Pelajaran yang berisi beberapa field. | Sistem menampilkan dialog Tambah Mata Pelajaran yang berisi beberapa field. | Berhasil |
| E004 | Menekan tombol Batal pada dialog Tambah Mata Pelajaran | Sistem menutup dialog Tambah Mata Pelajaran | Sistem menutup dialog Tambah Mata Pelajaran | Berhasil |
| E005 | Mengisi field Mata Pelajaran “IPA ”dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Mata Pelajaran, Berhasil Menambah Mata Pelajaran” kemudian tampil halaman Mata Pelajaran | Sistem menerima dan menampilkan Pemberitahuan “Data Mata Pelajaran, Berhasil Menambah Mata Pelajaran” kemudian tampil halaman Mata Pelajaran | Berhasil |
| E006 | Mengosongkan field Mata Pelajaran dan menekan tombol simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Mata Pelajaran, Gagal Menambah Mata Pelajaran”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan Pemberitahuan “Data Mata Pelajaran, Gagal Menambah Mata Pelajaran”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Berhasil |
| E007 | Mengisi field Mata Pelajaran sama dengan Mata Pelajaran sudah terdaftar dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Mata Pelajaran, Kelas sudah terdaftar, Gagal Menambah Mata Pelajaran”. | Sistem menampilkan Pemberitahuan “Data Mata Pelajaran, Kelas sudah terdaftar, Gagal Menambah Mata Pelajaran”. | Berhasil |
| E008 | Menekan tombol Edit Mata Pelajaran | Sistem menampilkan dialog Edit Mata Pelajaran yang berisi beberapa field. Isi dari field sesuai dengan data yang ingin di edit. | Sistem menampilkan dialog Edit Mata Pelajaran. | Berhasil |
| E009 | Menekan tombol Batal pada dialog Edit Mata Pelajaran | Sistem menutup dialog Edit Mata Pelajaran | Sistem menutup dialog Edit Mata Pelajaran | Berhasil |
| E010 | Mengubah isi dari field dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Mata Pelajaran, Berhasil Mengupdate Mata Pelajaran” Kemudian tampil halaman Mata Pelajaran. | Sistem menampilkan Pemberitahuan “Data Mata Pelajaran, Berhasil Mengupdate Mata Pelajaran” Kemudian tampil halaman Mata Pelajaran. | Berhasil |
| E011 | Mengosongkan isi field dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Mata Pelajaran, Gagal Mengupdate Mata Pelajaran”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan Pemberitahuan “Data Mata Pelajaran, Gagal Mengupdate Mata Pelajaran”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Berhasil |
| E012 | Mengubah isi field dan menekan tombol Batal | Sistem menutup dialog Edit Mata Pelajaran, data tidak berubah, dan menampilkan halaman Mata Pelajaran. | Sistem menutup dialog Edit Mata Pelajaran, data tidak berubah, dan menampilkan halaman Mata Pelajaran. | Berhasil |
| E013 | Menekan tombol Hapus | Sistem menampilkan dialog Hapus Data. | Sistem menampilkan dialog Hapus Data. | Berhasil |
| E014 | Menekan tombol Batal pada dialog Hapus Mata Pelajaran | Sistem menutup dialog Hapus Mata Pelajaran | Sistem menutup dialog Hapus Mata Pelajaran | Berhasil |
| E015 | Menekan tombol Hapus pada dialog Hapus Mata Pelajaran | Sistem menghapus Data Mata Pelajaran sesuai dengan yang dipilih, sistem menutup dialog Hapus Mata Pelajaran, dan menampilkan halaman Mata Pelajaran. | Sistem menghapus Data Mata Pelajaran sesuai dengan yang dipilih, muncul pemberitahuan “Data Mata Pelajaran, Berhasil Menghapus Mata Pelajaran”. sistem menutup dialog Hapus Mata Pelajaran, dan menampilkan halaman Mata Pelajaran. | Berhasil |
| F001 | Mengisi field pencarian dengan Mata Pelajaran | Sistem menampilkan Tema berdasarkan Mata Pelajaran yang diinputkan. | Sistem menampilkan Tema berdasarkan Mata Pelajaran. | Berhasil |
| F002 | Mengisi field pencarian dengan Tema | Sistem menampilkan Tema berdasarkan Tema yang diinputkan. | Sistem menampilkan Tema berdasarkan Tema. | Berhasil |
| F003 | Mengisi field pencarian dengan selain Mata Pelajaran/Tema | Sistem menampilkan tabel kosong. | Sistem menampilkan tabel kosong. | Berhasil |
| F004 | Menekan tombol Tambah Tema | Sistem menampilkan dialog Tambah Tema yang berisi beberapa field. | Sistem menampilkan dialog Tambah Tema. | Berhasil |
| F005 | Menekan tombol Batal pada dialog Tambah Tema | Sistem menutup dialog Tambah Tema | Sistem menutup dialog Tambah Tema | Berhasil |
| F006 | Mengisi field Mata Pelajaran "IPA", Tema "Tema 1 IPA", Sub Tema "Sub Tema 1 IPA" dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Tema, Berhasil Menambah Tema” kemudian tampil halaman Tema | Sistem menampilkan Pemberitahuan “Data Tema, Berhasil Menambah Tema” kemudian tampil halaman Tema | Berhasil |
| F007 | Mengisi field Mata Pelajaran "IPA", Tema "Tema 1 IPA", Sub Tema "Sub Tema 1 IPA" kemudian mengosongkan salah satu field, dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Tema, Gagal Menambah Tema”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan Pemberitahuan “Data Tema, Gagal Menambah Tema”. Muncul pemberitahuan pada field yang kosong, bahwa field harus diisi terlebih dahulu. | Berhasil |
| F008 | Mengisi field Tema sama dengan Tema sudah terdaftar dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Tema, Tema sudah terdaftar, Gagal Menambah Tema”. | Sistem menampilkan Pemberitahuan “Data Tema, Tema sudah terdaftar, Gagal Menambah Tema”. | Berhasil |
| F009 | Menekan tombol Edit Tema | Sistem menampilkan dialog Edit Tema yang berisi beberapa field. Isi dari field sesuai dengan data yang ingin di edit. | Sistem menampilkan dialog Edit Tema. | Berhasil |
| F010 | Menekan tombol Batal pada dialog Edit Tema | Sistem menutup dialog Edit Tema | Sistem menutup dialog Edit Tema | Berhasil |
| F011 | Mengubah isi dari field dan menekan tombol Simpan | Sistem menerima dan menampilkan Pemberitahuan “Data Tema, Berhasil Mengupdate Tema” Kemudian tampil halaman Tema. | Sistem menampilkan Pemberitahuan “Data Tema, Berhasil Mengupdate Tema” Kemudian tampil halaman Tema. | Berhasil |
| F012 | Mengosongkan isi field dan menekan tombol Simpan | Sistem menolak dan menampilkan Pemberitahuan “Data Tema, Gagal Mengupdate Tema”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Sistem menampilkan Pemberitahuan “Data Tema, Gagal Mengupdate Tema”. Muncul pemberitahuan pada field yang error data input, bahwa field harus diisi terlebih dahulu. | Berhasil |
| F013 | Mengubah isi field dan menekan tombol Batal | Sistem menutup dialog Edit Tema, data tidak berubah, dan menampilkan halaman Tema. | Sistem menutup dialog Edit Tema, data tidak berubah, dan menampilkan halaman Tema. | Berhasil |
| F014 | Menekan tombol Hapus | Sistem menampilkan dialog Hapus Data. | Sistem menampilkan dialog Hapus Data. | Berhasil |
| F015 | Menekan tombol Batal pada dialog Hapus Tema | Sistem menutup dialog Hapus Tema | Sistem menutup dialog Hapus Tema | Berhasil |
| F016 | Menekan tombol Hapus pada dialog Hapus Tema | Sistem menghapus Data Tema sesuai dengan yang dipilih, sistem menutup dialog Hapus Tema, dan menampilkan halaman Tema. | Sistem menghapus Data Tema sesuai dengan yang dipilih, muncul pemberitahuan “Data Tema, Berhasil Menghapus Data Tema”, sistem menutup dialog Hapus Tema, dan menampilkan halaman Tema. | Berhasil |
| Total task: **89** | | | | |

Untuk melakukan pengujian perangkat lunak, test case terbagi menjadi 2 bagian yaitu untuk Guru dan Admin. Pada pengujian untuk Guru, terdapat 66 test case yang diujikan. Setelah pengujian dilakukan, ditemukan hasil bahwa 64 test case berhasil dan 2 test case gagal. Sedangkan pada pengujian untuk Admin, terdapat 89 test case yang diujikan. Setelah pengujian dilakukan, ditemukan hasil bahwa 85 test case berhasil dan 4 test case gagal.

**Tabel 4.** Presentase Kelayakan Aplikasi Guru

|  |  |
| --- | --- |
| Jumlah test case | 66 |
| Jumlah test case berhasil | 66 |
| Jumlah test case gagal | 0 |
| Presentase Kelayakan | **(66/66) x 100 = 100%** |

**Tabel 4.** Presentase Kelayakan Aplikasi Admin

|  |  |
| --- | --- |
| Jumlah test case | 89 |
| Jumlah test case berhasil | 89 |
| Jumlah test case gagal | 0 |
| Presentase Kelayakan | **(89/89) x 100 = 100%** |

Berdasarkan hasil pengujian dengan menggunakan beberapa form sampel, didapatkan bahwa aplikasi ini memiliki skor kelayakan sebesar **100%** untuk bagian Guru dan **100%** untuk bagian Admin. Meskipun demikian, pengujian tersebut belum dapat dikatakan sempurna karena masih terbatas dalam jumlah sampel yang digunakan. Oleh karena itu, masih diperlukan pengujian lebih lanjut untuk memastikan bahwa aplikasi dapat berjalan dengan baik dan sesuai kebutuhan pengguna.

BAB 5

Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan hasil perancangan aplikasi berbasis web framework Angular dengan metode Rapid Application Development (RAD) untuk Sistem Informasi Pengolahan Nilai Harian SDN 1 Warga Mulya Berbasis Web, berikut adalah simpulan yang dapat diambil:

1. Aplikasi ini berhasil dikembangkan menggunakan Angular dan metode RAD untuk mempercepat proses pengembangan.
2. Aplikasi ini berhasil dikembangkan sesuai dengan kebutuhan pengguna pada fase perencanaan kebutuhan (*Requirement Planning Phase*).
3. Beberapa tools dalam pengembangan ini memberikan efisiensi dan efektivitas, termasuk waktu pengembangan yang lebih singkat, tahapan yang jelas, dan hasil yang memuaskan.
4. Hasil perancangan ini dapat menjadi dasar untuk mengembangkan lebih lanjut aplikasi Sistem Informasi Pengolahan Nilai Harian SDN 1 Warga Mulya Berbasis Web.

5.2 Saran

Setelah melakukan pengembangan, diperlukan pengembangan lebih lanjut terutama pada bagian database. Database yang dapat mengelola data kompleks seperti MySQL, MongoDB, Firebase, dan lainnya dapat digunakan. Selain itu, perawatan terhadap aplikasi juga diperlukan, termasuk pembaharuan pada framework yang digunakan untuk meningkatkan tingkat keamanan dan keterbaharuan. Pengelolaan basis data juga perlu diperhatikan seiring dengan perkembangan Sistem Informasi Pengolahan Nilai Harian SDN 1 Warga Mulya Berbasis Web dan pertumbuhan jumlah data nilai di masa mendatang.

**Daftar Acuan**

Julio, E., Pakereng, M. A. I., & Artikel, I. (2021). Implementasi API Payment Gateway Menggunakan Arsitektur Microservice. *JURNAL INFORMATIKA*, *8*(2). http://ejournal.bsi.ac.id/ejurnal/index.php/ji

Pranata, B. A., Hijriani Dan, A., Junaidi, A., Komputer, J. I., Matematika, F., Pengetahuan, I., Universitas, A., Jl, L. S., Brodjonegoro, N., & Lampung, B. (2018). *PERANCANGAN APPLICATION PROGRAMMING INTERFACE (API) BERBASIS WEB MENGGUNAKAN GAYA ARSITEKTUR REPRESENTATIONAL STATE TRANSFER (REST) UNTUK PENGEMBANGAN SISTEM INFORMASI ADMINISTRASI PASIEN KLINIK PERAWATAN KULIT* (Vol. 6, Issue 1).

Putri, M. P., & Effendi, H. (2018). Implementasi Metode Rapid Application Development Pada Website Service Guide “Waterfall Tour South Sumatera.” In *Jurnal SISFOKOM* (Vol. 07).

Rulloh, A., Mahmudah, D. E., & Kabetta, H. (2017). *Implementasi REST API pada Aplikasi Panduan Kepaskibraan Berbasis Android*.

Sembara, S. (2020). *RANCANG BANGUN APPLICATION PROGRAMMING INTERFACE*.

Abushama, H., M., Alassam, H., A., & Elhaj, F., A. (2020). The effect of test Test-Driven Development and Behavior-Driven Development on Project Success Factors: A Systematic Literature Review Based Study. *2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE).* IEEE Xplore.

Beck, K. (2002). Test-Driven Development By Example. Addison Wesley. November.

Buani, D. C. P. (2020). Penerapan Metode Rad Dalam Perancangan (Siremedi) Pada Klinik Halo Fisio. *EVOLUSI : Jurnal Sains Dan Manajemen*, *8*(1), 56–62. https://doi.org/10.31294/evolusi.v8i1.7539

Hellmann, T. D., Hosseini-Khayat, A., & Maurer, F. (2010). Supporting test-driven development of graphical user interfaces using agile interaction design. *ICSTW 2010 - 3rd International Conference on Software Testing, Verification, and Validation Workshops*, 444–447. https://doi.org/10.1109/ICSTW.2010.35

Jain, P., Sharma, A., & Ahuja, L. (2018). The Impact of Agile Software Development Process on the Quality of Software Product. *2018 7th International Conference on Reliability, Infocom Technologies and Optimization: Trends and Future Directions, ICRITO 2018*, 812–815. https://doi.org/10.1109/ICRITO.2018.8748529

Khan, W., Ahmad, W., Luo, B., & Ahmed, E. (2019). SQL database with physical database tuning technique and NoSQL graph database comparisons. *Proceedings of 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2019*, *Itnec*, 110–116. https://doi.org/10.1109/ITNEC.2019.8729264

Komalasari, N., Murad, D. F., Agustine, D., Irsan, M., Budiman, J., & Fernando, E. (2018). Effect of education, performance, position and information technology competency of information systems to performance of information system. *2018 International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2018*, 221–226. https://doi.org/10.1109/ISRITI.2018.8864437

Kosasi, S., & Eka Yuliani, I. D. A. (2015). Penerapan Rapid Application Development Pada Sistem Penjualan Sepeda Online. *Simetris : Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, *6*(1), 27. https://doi.org/10.24176/simet.v6i1.234

Kyi, K. H., Lu, M., & Li, X. (2016). Development of a user-friendly web-based rainfall-runoff model. *Hydrological Research Letters*, *10*(1), 8–14. https://doi.org/10.3178/hrl.10.8

Manuaba, I. B. K. (2019). Combination of test-driven development and behavior-driven development for improving backend testing performance. *Procedia Computer Science*, *157*, 79–86. https://doi.org/10.1016/j.procs.2019.08.144

Multazam, M. (2020). Perancangan User Interface dan User Experience pada Placeplus menggunakan pendekatan User Centered Design. *Universitas Islam Indonesia*, *1*, 8.

Nilawati, L., Sulastri, D., & Yuningsih, Y. (2020). Penerapan Model Rapid Application Development Pada Perancangan Sistem Informasi Jasa Pengiriman Barang. *Paradigma - Jurnal Komputer Dan Informatika*, *22*(2), 197–204. https://doi.org/10.31294/p.v22i2.8314

Parahita, R., Widiartha, I. B. K., & Zubaidi, A. (2018). Sistem Informasi Perhotelan Berbasis Web Service, Studi Kasus Di Pulau Lombok. *Journal of Computer Science and Informatics Engineering (J-Cosine)*, *1*(1), 46. https://doi.org/10.29303/jcosine.v1i1.21

Purnia, D. S., Rifai, A., & Rahmatullah, S. (2019). Penerapan Metode Waterfall dalam Perancangan Sistem Informasi Aplikasi Bantuan Sosial Berbasis Android. *Seminar Nasional Sains Dan Teknologi 2019*, 1–7.

Ramirez, M. O. G., De-La-Torre, M., & Monsalve, C. (2019). Methodologies for the design of application frameworks: Systematic review. *2019 8th International Conference on Software Process Improvement, CIMPS 2019 - Applications in Software Engineering*. https://doi.org/10.1109/CIMPS49236.2019.9082427

Sneha, A., & Pushpanjali M., C. (2016). Angular Js. *International Journal of Scientific & Engineering Research,* *7*(2), 73–76. http://www.ijser.org/researchpaper/Angular-JS.pdf

Utomo, S. P., Alfiyah, N. H., Sani, Z. A., Hanafi, M., & Primadewi, A. (2020). Informasi Terintegrasi Menggunakan FrameWork CodeIgniter. *Seminar Nasional Dinamika Informatika*, 124–128.

"*JSON Server: A Simple Mock REST API for Rapid Application Development*" oleh Ruwan Geeganage, Muttukrishnan Rajarajan, dan Marimuthu Karuppiah

Beizer, B. (1990). *Software Testing Techniques, Second Edition by Boris Beizer*. The Coriolis Group.

Beizer, B. (1995). *Black-box Testing : Techniques for Functional Testing of Software and Systems*. John Wiley & Sons, Inc. (US).

Glenford J. Myers. (2004). The Art of Software Testing. In *John Wiley & Sons, Inc*.

Krismadi, A., Lestari, A. F., Pitriyah, A., Putra, I. W., & Mardangga, A. (2019). *Pengujian Black Box berbasis Equivalence Partitions pada Aplikasi Seleksi Promosi Kenaikan Jabatan*. *2*(4), 155–161.

Kurniawan, A., Maulana, A., Sukma, V. R., Keumala, W., & Saifudin, A. (2020). *Pengujian Black Box pada Aplikasi Penjualan Berbasis Web Menggunakan Metode Equivalents Partitions ( Studi Kasus : PT Arap Store )*. *3*(1), 50–56.

Nurudin, M., Jayanti, W., Saputro, R. D., & Saputra, M. P. (2020). *Pengujian Black Box pada Aplikasi Penjualan Berbasis Web Menggunakan Teknik Boundary Value Analysis*. *February*. https://doi.org/10.32493/informatika.v4i4.3841

Sinulingga, A. R., Zuhri, M., Mukti, R. B., & Saifudin, A. (2020). *Pengujian Black Box pada Sistem Aplikasi Informasi Data Kinerja Menggunakan Teknik Equivalence Partitions*. *3*(1), 9–14.