

BAB 3

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Proses pembuatan sistem ini akan dilakukan dengan menggunakan metode iteratif, di mana setiap tahapannya mencakup perencanaan, pembangunan, evaluasi, dan implementasi. Dalam perancangan *web* ini, pendekatan iteratif diterapkan dengan pola perputaran yang terjadi pada setiap modul secara bertahap. Salah satu komponen utama yang akan dikembangkan adalah *Parent Portal*, sebuah platform berbasis *web* yang dirancang khusus untuk membantu orang tua dalam memantau perkembangan akademik anak-anak secara langsung. Dengan adanya *Parent Portal*, orang tua dapat dengan mudah mengakses berbagai informasi penting terkait aktivitas belajar anak-anak, termasuk kehadiran, nilai, dan tugas yang harus diselesaikan. Portal ini bertujuan untuk meningkatkan keterlibatan orang tua dalam proses pendidikan anak, sehingga komunikasi antara sekolah dan orang tua menjadi lebih efektif serta transparan. Fitur utama yang akan tersedia dalam *Parent Portal* mencakup akses *real time* terhadap data akademik, laporan kehadiran, serta pemberitahuan penting yang memungkinkan orang tua untuk selalu mendapatkan informasi terbaru mengenai perkembangan anak-anak. Dengan pendekatan iteratif, setiap fitur akan dikembangkan secara bertahap dan diperbaiki berdasarkan umpan balik dari *stakeholder*, sehingga sistem yang dihasilkan dapat terus disempurnakan sesuai dengan kebutuhan pengguna.

3.1.1 Pendekatan Pengembangan Sistem

Proses pembuatan sistem ini dilakukan dengan menggunakan metode *iterative*, di mana setiap tahapannya mencakup perencanaan, pembangunan, evaluasi, dan implementasi. Dalam perancangan *web* ini, pendekatan iteratif diterapkan dengan pola perputaran yang terjadi pada setiap modul secara bertahap [1] [14]. Setiap iterasi dilakukan untuk menghasilkan versi sistem yang lebih sempurna, berdasarkan masukan dan pengujian yang dilakukan di tahap sebelumnya.

3.1.2 Tujuan dan Fokus Pengembangan Sistem

Salah satu komponen utama yang dikembangkan adalah *Parent Portal*, yaitu *platform* berbasis *web* yang dirancang untuk memberikan akses langsung terhadap informasi akademik siswa. Melalui *portal* ini, tersedia berbagai data penting seperti kehadiran, nilai, dan tugas yang belum diselesaikan. Kehadiran *Parent Portal* diharapkan dapat meningkatkan keterlibatan dalam proses pendidikan, serta memperkuat komunikasi antara pihak sekolah dan wali murid secara lebih efektif dan transparan.

Fitur utama yang tersedia dalam *Parent Portal* mencakup akses *real-time* terhadap data akademik, laporan kehadiran, serta pemberitahuan penting yang memungkinkan wali murid untuk selalu mendapatkan informasi terbaru mengenai perkembangan siswa. Dengan pendekatan iteratif, setiap fitur akan dikembangkan secara bertahap dan diperbaiki berdasarkan umpan balik dari *stakeholder*, sehingga sistem yang dihasilkan dapat terus disempurnakan sesuai dengan kebutuhan pengguna.

3.1.3 Fitur Unggulan dalam *Parent Portal*

1. Kehadiran Siswa

Halaman kehadiran menampilkan riwayat kehadiran siswa, mencakup kehadiran harian, keterlambatan, serta izin ketidakhadiran. Informasi ditampilkan dalam bentuk grafik atau tabel yang mudah dipahami, disertai notifikasi otomatis apabila terjadi ketidakhadiran tanpa keterangan. Fitur ini memudahkan pemantauan terhadap kedisiplinan siswa dan memungkinkan tindak lanjut secara cepat apabila ditemukan pola absensi yang tidak wajar.

2. Nilai Akademik dan Rekapitulasi Perkembangan

Portal ini memberikan akses terhadap nilai ujian, tugas, serta proyek yang telah dikerjakan. Selain itu, tersedia rekapitulasi perkembangan akademik dari waktu ke waktu, sehingga memungkinkan pemantauan terhadap peningkatan atau penurunan prestasi pada setiap mata pelajaran.

3. Informasi Sekolah

Fitur ini menyediakan berbagai informasi penting terkait institusi pendidikan, seperti alamat sekolah, nomor telepon, dan nomor faks. Informasi ini ditampilkan secara jelas agar orang tua dapat dengan mudah menghubungi

pihak sekolah jika diperlukan.

4. **Jadwal Pembelajaran**

Fitur jadwal harian memberikan akses langsung terhadap informasi kegiatan belajar setiap hari, mencakup daftar mata pelajaran yang diikuti pada hari tersebut beserta nama pengajar yang bertanggung jawab.

5. **Pemantauan Tugas dan Status Penyelesaian**

Sistem menampilkan daftar lengkap tugas yang diberikan oleh guru dalam setiap mata pelajaran, meliputi tugas yang telah dikumpulkan, dikumpulkan melewati batas waktu, serta yang belum dikerjakan. Sistem juga menyediakan fitur notifikasi untuk mengingatkan pengguna mengenai keterlambatan atau tugas yang belum diselesaikan.

6. **Notifikasi *Real-time***

Sistem akan mengirimkan notifikasi langsung ke *email* atau aplikasi *mobile* jika ada informasi penting, seperti tugas baru, nilai ujian yang sudah keluar, ketidakhadiran anak, atau pesan dari guru.

7. **Akses *Multi-Platform***

Akses terhadap *Parent Portal* tersedia melalui berbagai jenis perangkat, seperti komputer, tablet, dan ponsel, sehingga memungkinkan pemantauan dilakukan secara fleksibel dari mana saja.

3.1.4 **Proses Pengujian dan Validasi Sistem**

Sistem yang dikembangkan tidak di-*hosting* secara publik karena kebijakan internal dari institusi tempat penelitian dilakukan. Proses *deployment* dan pengujian dilakukan secara internal oleh mentor pengawas proyek, yang kemudian mengatur agar aplikasi hanya dapat diakses melalui jaringan lokal tertentu. Hal ini bertujuan untuk menjaga kerahasiaan serta keamanan data selama masa pengembangan dan evaluasi sistem.

Pengujian sistem dilakukan oleh staf internal sekolah yang juga merupakan pengguna langsung dari *Parent Portal*. Beberapa di antaranya adalah staf yang juga berperan sebagai orang tua siswa, sehingga dapat mengevaluasi sistem berdasarkan dua perspektif: sebagai staf sekolah dan sebagai wali murid. Uji coba dilakukan dengan menjalankan sistem di lingkungan yang menyerupai kondisi operasional

sebenarnya, di mana pengguna diminta untuk mencoba fitur-fitur utama seperti melihat nilai, memantau kehadiran, dan mengakses jadwal pembelajaran siswa.

Meskipun tidak menggunakan metode formal seperti *black-box testing* atau *white-box testing*, pendekatan ini tetap memungkinkan dilakukannya validasi fungsionalitas dan kenyamanan penggunaan sistem berdasarkan umpan balik nyata dari pengguna akhir. Proses ini juga menjadi bagian dari tahapan iteratif, di mana hasil evaluasi dari pengguna internal digunakan untuk melakukan perbaikan dan pengembangan lebih lanjut terhadap sistem.

Dengan fitur-fitur ini, *Parent Portal* bertujuan untuk meningkatkan keterlibatan orang tua dalam pendidikan anak, menjaga konsistensi prestasi akademik siswa, serta membangun komunikasi yang lebih baik antara orang tua, siswa, dan guru. Sistem ini dirancang tidak hanya sebagai alat pemantau, tetapi juga sebagai jembatan yang memperkuat hubungan antara rumah dan sekolah dalam menciptakan lingkungan belajar yang lebih terarah dan kolaboratif.

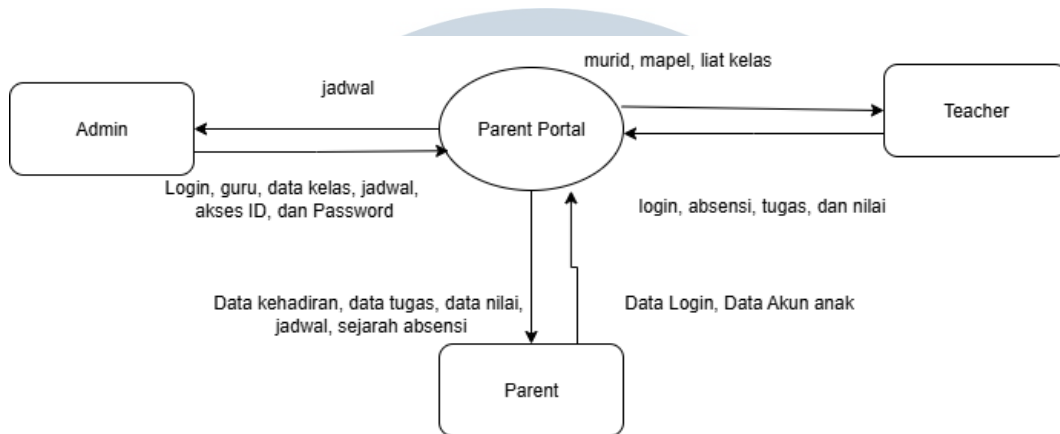
3.2 Perancangan Aplikasi

Perancangan aplikasi sistem informasi sekolah ini menggunakan metode iteratif, yaitu pendekatan pengembangan perangkat lunak yang dilakukan secara bertahap melalui siklus-siklus (iterasi) berulang. Setiap iterasi mencakup tahapan perencanaan, perancangan, pengembangan, pengujian, dan evaluasi dari sebagian fitur sistem, dengan tujuan untuk memperoleh umpan balik secara dini dan terus-menerus memperbaiki sistem berdasarkan masukan yang diterima.

Metode ini dipilih karena cocok untuk proyek yang kompleks dan memerlukan fleksibilitas terhadap perubahan kebutuhan pengguna, khususnya dalam konteks pengembangan *Parent Portal* yang menghubungkan berbagai pihak seperti guru, siswa, dan orang tua. Pada setiap iterasi, dilakukan penambahan fungsionalitas baru atau penyempurnaan terhadap modul yang telah ada, dengan cakupan yang jelas dan terukur.

Proses perancangan sistem dimulai dengan pembuatan *Data Flow Diagram* (DFD), *Flowchart*, *Database Schema*, serta *Entity Relationship Diagram* (ERD) sebagai landasan dalam membangun sistem yang terstruktur dan konsisten secara logika.

3.2.1 Context Diagram



Gambar 3.1. DFD Level 0

Gambar 3.1 merupakan *Data Flow Diagram (DFD) Level 0* dari sistem *Parent Portal*, yang menggambarkan aliran data utama antara sistem dengan aktor eksternal, yaitu *Admin*, *Teacher*, dan *Parent*.

1. Admin ↔ Parent Portal

- **Data masuk ke sistem:** *Admin* bertanggung jawab memasukkan data seperti login, data guru, data kelas, jadwal pelajaran, serta akses *ID* dan *password* untuk siswa.
- **Data keluar dari sistem:** Sistem mengembalikan informasi jadwal yang telah diatur sebelumnya kepada *admin*.

Fungsi utama dari *Admin* adalah membangun struktur awal sistem yang mencakup kelas, jadwal, dan akun pengguna.

2. Teacher ↔ Parent Portal

- **Data masuk ke sistem:** *Teacher* menginput data seperti login, absensi, tugas, dan nilai siswa.
- **Data keluar dari sistem:** Sistem menyediakan daftar murid, mata pelajaran, dan informasi kelas kepada *teacher* sebagai dasar untuk mengelola proses pembelajaran.

Dengan demikian, *teacher* berperan aktif dalam mengisi informasi akademik yang akan ditampilkan pada *Parent Portal*.

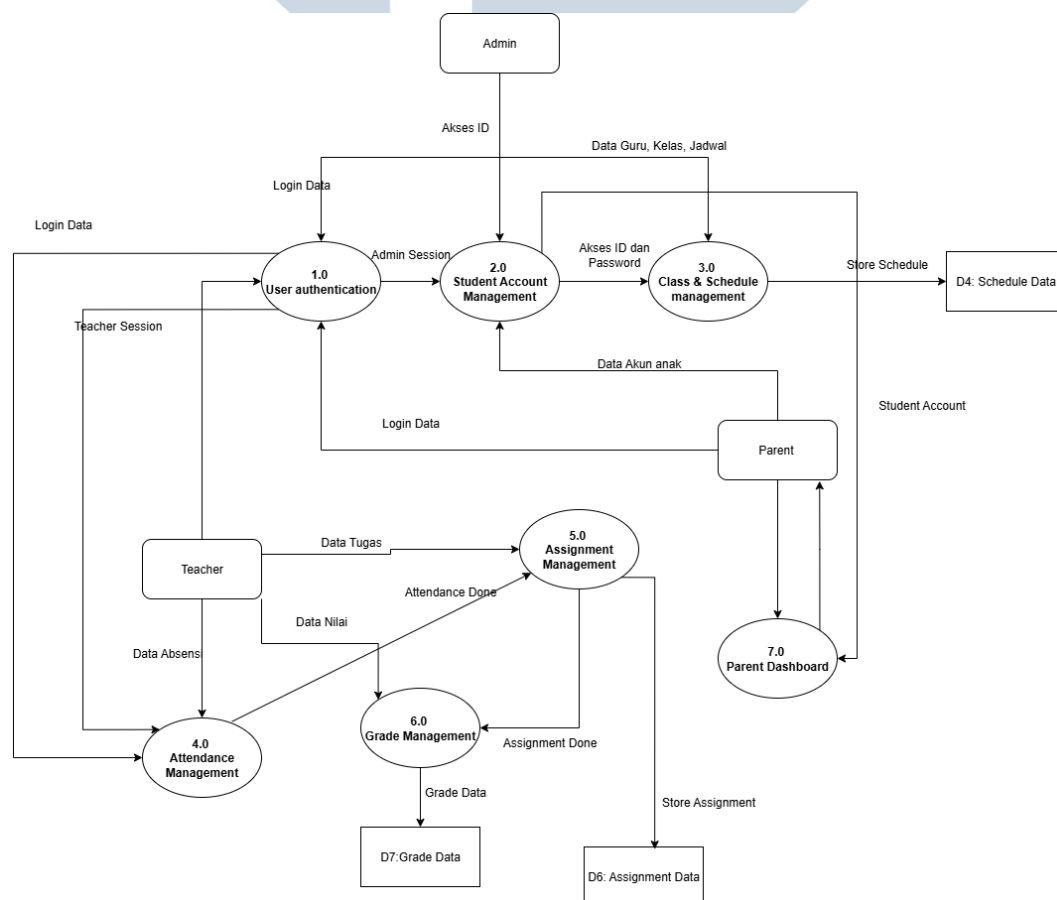
3. Parent ↔ Parent Portal

- **Data masuk ke sistem:** *Parent* melakukan login ke sistem dan dapat menambahkan akun anak mereka menggunakan akses *ID*.
- **Data keluar dari sistem:** Sistem memberikan akses kepada *parent* untuk melihat data kehadiran, nilai, tugas, jadwal, serta riwayat absensi siswa.

Tujuan utama dari fitur ini adalah untuk memudahkan *parent* dalam memantau perkembangan akademik anak mereka secara menyeluruh dan *real-time*.

Secara keseluruhan, diagram ini menunjukkan bahwa sistem *Parent Portal* berperan sebagai pusat kendali data, yang menghubungkan proses akademik dari sekolah dengan orang tua secara langsung dan efisien.

3.2.2 DFD



Gambar 3.2. DFD Level 1

Gambar 3.2 merupakan *Data Flow Diagram (DFD)* Level 1 menggambarkan aliran data utama yang terjadi antara pengguna sistem dan proses-proses inti dalam sistem informasi sekolah. Sistem ini melibatkan empat entitas eksternal, yaitu *Admin*, *Teacher*, *Parent*, dan *Student*, yang masing-masing berinteraksi dengan sistem melalui proses-proses tertentu. Seluruh interaksi ini berpusat pada sistem yang disebut *Parent Portal*.

Entitas *Parent* berperan dalam melakukan *login* untuk mengakses informasi, menambahkan akun siswa menggunakan *access ID* dan *access password*, serta melihat data seperti nilai, tugas, kehadiran, dan jadwal. *Teacher* bertugas untuk menginput data kehadiran melalui proses *take attendance*, membuat tugas melalui proses *create assignment*, serta mengisi nilai siswa melalui proses *enter grades*. Semua tugas dan nilai dikaitkan dengan mata pelajaran dan kelas yang diajarkan oleh *teacher* tersebut.

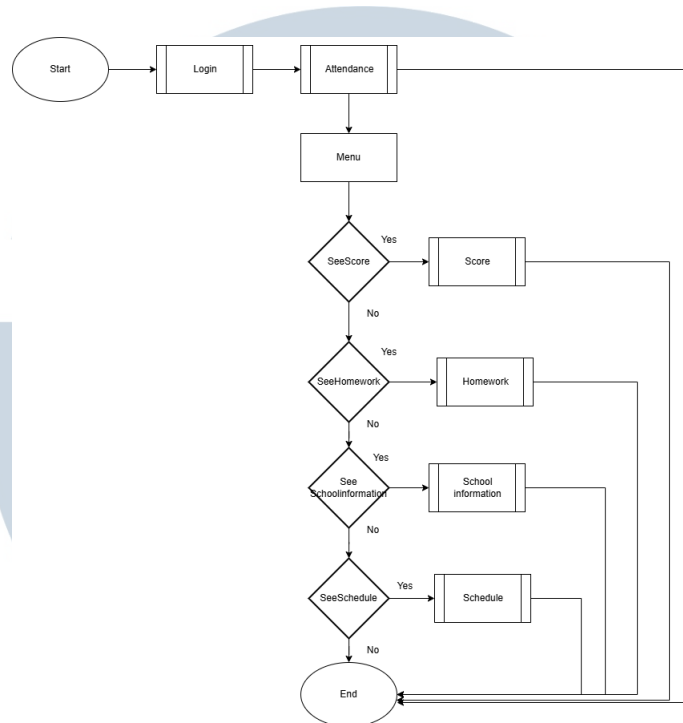
Sementara itu, *Student* masuk ke dalam kelas dan kehadirannya akan otomatis dicatat melalui sistem. Selain itu, *student* juga akan mendapatkan tugas dan jadwal berdasarkan mata pelajaran yang telah dipilih.

Admin berfungsi sebagai pengelola utama sistem. Proses yang dilakukan meliputi pembuatan data kelas, pengaturan guru dan mata pelajaran, penyusunan jadwal, serta pembuatan *access ID* dan *password* yang akan digunakan oleh *parent* untuk menambahkan akun siswa ke dalam sistem. Proses penyusunan jadwal dilakukan dengan menentukan guru, mata pelajaran, kelas, ruang, durasi pertemuan, dan hari. Setelah jadwal diatur, sistem akan mendeteksi jika terjadi bentrok dan hanya akan menyimpan jadwal yang valid.

Sistem menyimpan informasi penting ke dalam beberapa data store. Data login semua jenis pengguna disimpan dalam *User Accounts*. Data absensi siswa disimpan dalam *Attendance*. Nilai siswa yang telah diinput oleh guru disimpan dalam *Grades*. Informasi jadwal pelajaran yang dibuat oleh *admin* dan *teacher* disimpan dalam *Schedule*. Relasi antara kelas, mata pelajaran, dan guru disimpan dalam *Courses*.

Dengan pemetaan seperti ini, sistem mampu memberikan informasi secara *real-time* kepada *parent* mengenai aktivitas akademik anaknya, serta membantu pihak sekolah dalam pengelolaan data yang terstruktur.

3.2.3 Flowchart

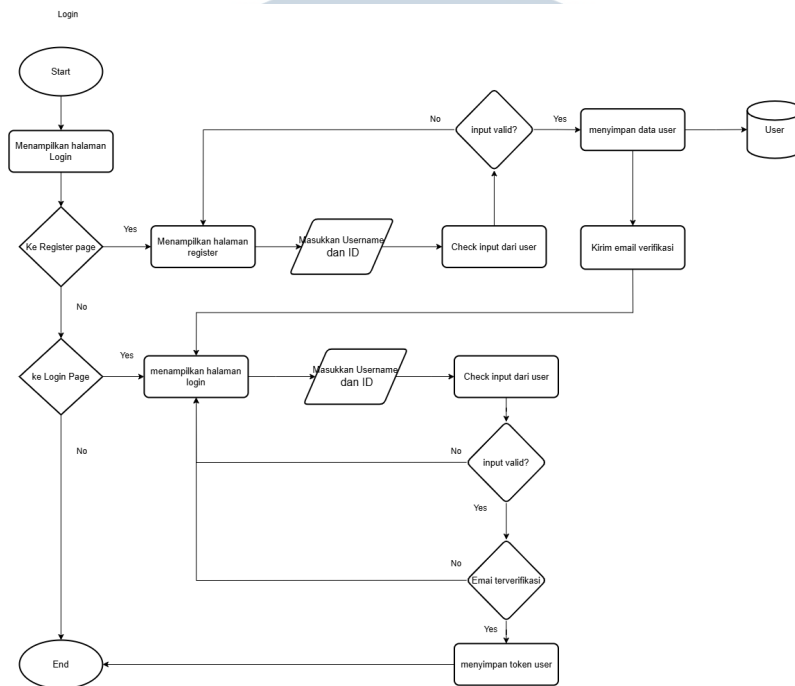


Gambar 3.3. Aplikasi

Gambar 3.3 menunjukkan *flowchart* keseluruhan dari aplikasi *Parent Portal* yang dikembangkan dalam penelitian ini. *Flowchart* ini menggambarkan alur proses sistem secara menyeluruh, dimulai dari tahap awal ketika pengguna mengakses sistem melalui halaman login, hingga pengguna keluar dari sistem. Diagram ini mencakup semua fitur utama yang tersedia dalam aplikasi, termasuk autentikasi pengguna (baik sebagai orang tua maupun admin), navigasi antar modul, dan interaksi antara pengguna dengan fungsi-fungsi inti seperti melihat nilai, kehadiran, jadwal, dan tugas siswa.

Selain itu, *flowchart* ini juga menunjukkan logika pengambilan keputusan dan alur data yang menghubungkan setiap komponen sistem, sehingga dapat memberikan pemahaman yang utuh mengenai bagaimana setiap bagian dari aplikasi saling berinteraksi dan saling mendukung dalam mendukung tujuan utama, yaitu memfasilitasi orang tua dalam memantau perkembangan akademik anak secara efektif. Dengan demikian, gambar ini berperan penting sebagai representasi visual dari struktur dan logika kerja sistem informasi yang dirancang.

3.2.4 Login dan Register



Gambar 3.4. Login dan Register

Gambar 3.4 menampilkan halaman autentikasi yang terdiri dari *login* dan pendaftaran pengguna. Sistem menggunakan pendekatan iteratif dengan tahapan: *requirement, design and development, testing, dan implementation*.

A Iterasi Pertama

Requirement: Pada iterasi awal, kebutuhan utama yang ingin dipenuhi adalah menyediakan sistem autentikasi pengguna yang meliputi fitur *login, register, dan forgot password*. Sistem ini dirancang untuk membedakan hak akses pengguna seperti orang tua, siswa, guru, dan admin sekolah.

Design and Development: Desain awal mencakup pembuatan halaman *login* dan *register* dengan form isian seperti *email, password, nama lengkap, dan tipe pengguna*. Proses pengembangan dilakukan dengan menggunakan pendekatan formulir yang terhubung ke *database* untuk melakukan verifikasi akun, serta menggunakan logika *session* untuk mengatur hak akses berdasarkan tipe pengguna yang *login*.

Testing: Pengujian dilakukan dengan membuat beberapa akun percobaan untuk tipe pengguna berbeda dan mencoba proses *login*, *logout*, serta verifikasi *email*. Hasil pengujian menunjukkan bahwa fitur berjalan dengan baik, tetapi dari sisi tampilan masih terlihat kurang rapi dan tidak konsisten.

Implementation: Sistem autentikasi ini berhasil diimplementasikan secara fungsional dan telah mampu membedakan akses antara pengguna. Namun, perlu dilakukan perbaikan dari sisi antarmuka.

Output: Pengguna dapat mendaftar akun, menerima *email* verifikasi, dan *login* ke sistem sesuai dengan tipe akun masing-masing. Hak akses yang ditampilkan juga sesuai dengan peran pengguna.

B Iterasi Kedua

Requirement: Berdasarkan masukan dari pengujian sebelumnya, perlu dilakukan peningkatan pada sisi tampilan *form login* dan *register*, terutama pada penempatan elemen yang dirasa tidak intuitif.

Design and Development: Desain ulang dilakukan pada halaman *login* dan *register* dengan memperbaiki struktur, memperjelas label isian, menyelaraskan komponen *form*, serta menambahkan elemen visual seperti ikon dan *border* yang lebih tegas untuk memudahkan pengguna dalam mengisi *form*.

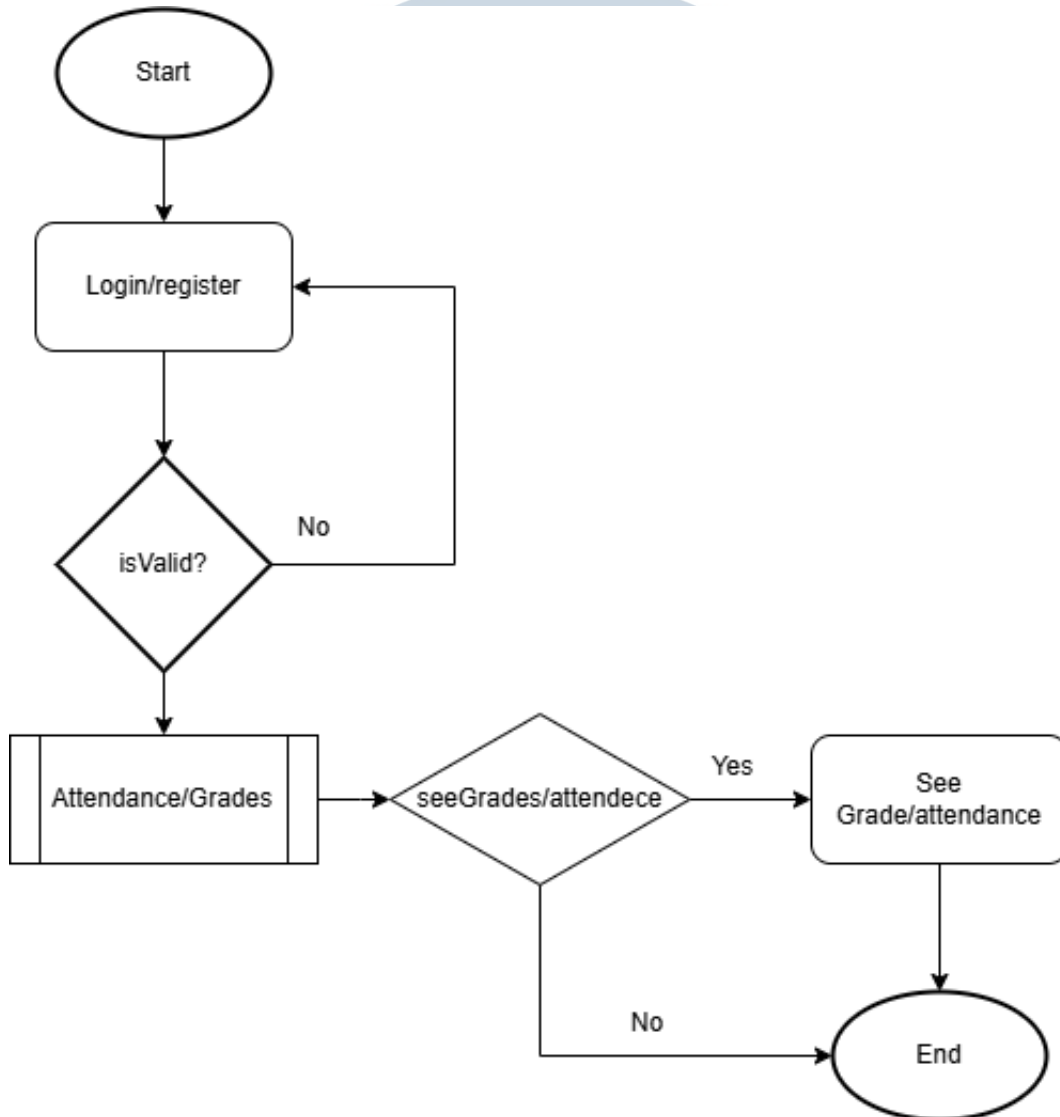
Testing: Pengujian dilakukan secara langsung dengan membandingkan versi lama dan baru. Pengguna merasa bahwa *form* kini lebih mudah dibaca dan diisi, serta tampilannya lebih modern dan konsisten.

Implementation: Hasil revisi tampilan diterapkan ke dalam sistem produksi, menggantikan versi lama.

Output: Form *login* dan *register* kini memiliki tampilan yang lebih profesional dan user-friendly, yang meningkatkan pengalaman pengguna secara keseluruhan.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

3.2.5 Nilai dan Kehadiran



Gambar 3.5. Score

Gambar 3.5 menampilkan semua mata pelajaran yang diambil oleh siswa, dilengkapi dengan informasi nilai dan status kehadiran pada hari tersebut.

A Iterasi Pertama

Requirement: Fitur ini bertujuan untuk menampilkan data akademik siswa, terutama nilai dan kehadiran. Kebutuhan ini muncul dari pentingnya transparansi data akademik bagi orang tua siswa.

Design and Development: Dikembangkan halaman khusus yang menampilkan tabel data nilai dan kehadiran siswa berdasarkan data dari guru. Tabel terdiri dari kolom mata pelajaran, nilai, status kehadiran (Hadir, Tidak Hadir, Terlambat), dan tanggal. Data diambil dari database menggunakan prosedur tersimpan (*stored procedure*) yang menyesuaikan dengan ID siswa yang dipilih.

Testing: Pengujian awal menunjukkan bahwa data muncul namun tampilan kurang menarik dan sulit dibaca karena warna dan font yang digunakan belum optimal.

Implementation: Versi awal fitur ini telah berfungsi, namun perlu pengembangan tampilan lebih lanjut.

Output: Orang tua sudah dapat melihat informasi kehadiran dan nilai anaknya setiap harinya dalam bentuk tabel meskipun masih terbatas dari sisi tampilan.

B Iterasi Kedua

Requirement: Pada iterasi ini ditemukan bahwa data kehadiran tidak ditampilkan secara akurat untuk beberapa siswa. Hal ini disebabkan oleh ketidaksesuaian parameter dalam prosedur pengambilan data.

Design and Development: Perbaikan dilakukan pada prosedur tersimpan di sisi *server* dan logika *query* yang digunakan pada halaman. *Parameter* ID siswa dan tanggal disesuaikan agar menampilkan data yang tepat.

Testing: Hasil pengujian menunjukkan bahwa data kehadiran kini muncul sesuai jadwal dan tidak lagi menampilkan data siswa yang salah.

Implementation: Perbaikan diterapkan secara menyeluruh di halaman kehadiran dan nilai.

Output: Data kehadiran siswa tampil secara real-time dan akurat berdasarkan ID siswa yang dipilih oleh orang tua.

C Iterasi Ketiga

Requirement: Masalah baru ditemukan pada data nilai, yakni nilai dari guru tidak tampil karena proses penyimpanan nilai ke database belum berjalan sempurna.

Design and Development: Tim pengembang memperbaiki logika penyimpanan nilai, memastikan bahwa nilai yang dimasukkan oleh guru dapat tersimpan dan ditampilkan di halaman orang tua. Validasi input nilai juga

ditambahkan agar tidak terjadi kesalahan format.

Testing: Dilakukan pengujian dengan memasukkan nilai dari sisi guru dan memeriksa apakah nilai tersebut muncul di halaman orang tua.

Implementation: Perbaikan ini berhasil diimplementasikan dan diuji.

Output: Orang tua kini dapat melihat nilai anaknya yang dimasukkan oleh guru tanpa adanya kesalahan atau keterlambatan data.

D Iterasi Keempat

Requirement: Saran dari pengguna menunjukkan bahwa tampilan tabel nilai dan kehadiran masih terasa kaku dan tidak responsif pada perangkat seluler.

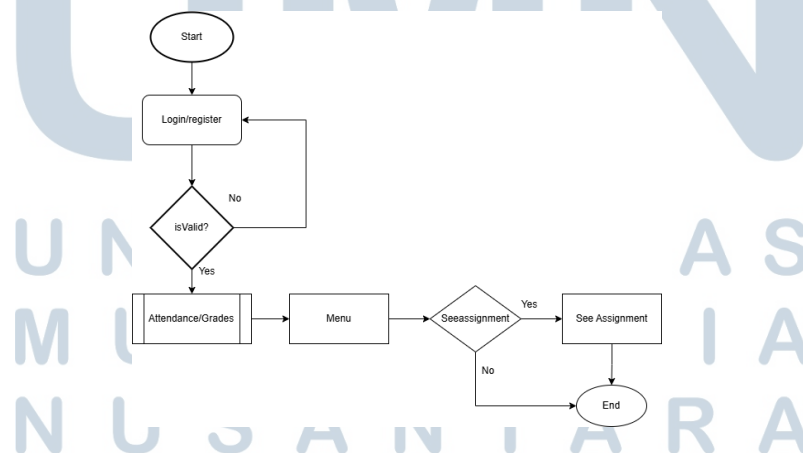
Design and Development: Tim memperbarui antarmuka dengan CSS yang lebih fleksibel dan menggunakan desain berbasis *grid* dan *media query* agar tampilan menyesuaikan ukuran layar perangkat.

Testing: Pengujian dilakukan di berbagai perangkat seperti laptop, tablet, dan ponsel.

Implementation: Desain baru ini diterapkan dan menggantikan versi sebelumnya.

Output: Tampilan data akademik kini jauh lebih responsif, informatif, dan mudah digunakan baik di *desktop* maupun di perangkat seluler.

3.2.6 Kehilangan Tugas



Gambar 3.6. Assignment

Gambar 3.6 menampilkan daftar tugas yang belum dikerjakan atau telah melewati batas waktu pengerjaan oleh siswa. Informasi mencakup nama guru, mata

pelajaran, dan status tugas seperti *Pending* atau *Missing*.

A Iterasi Pertama

Requirement: Fitur ini bertujuan agar orang tua dapat memantau tugas yang diberikan kepada siswa, termasuk tanggal pengumpulan, status pengerjaan, dan deskripsi tugas. Kebutuhan ini muncul karena orang tua ingin lebih terlibat dalam mendampingi anak dalam proses belajar di rumah.

Design and Development: Halaman tugas dirancang dengan tampilan tabel yang menampilkan daftar tugas berdasarkan *ID* siswa yang dipilih. Data diambil dari tabel tugas yang diinput oleh guru, lalu difilter berdasarkan *ID* siswa yang bersangkutan.

Testing: Pada tahap pengujian, ditemukan bahwa beberapa data tugas tidak muncul untuk siswa tertentu karena kesalahan dalam parameter pengambilan data.

Implementation: Meski tampilannya sudah sesuai, data yang ditampilkan masih tidak konsisten.

Output: Halaman tugas sudah tersedia, namun belum sepenuhnya menampilkan informasi yang lengkap atau akurat.

B Iterasi Kedua

Requirement: Perlu dilakukan perbaikan pada query dan prosedur tersimpan agar data tugas yang ditampilkan akurat berdasarkan siswa yang dipilih oleh orang tua.

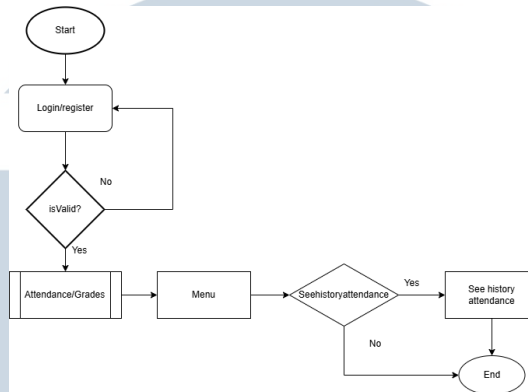
Design and Development: Perbaikan dilakukan dengan memodifikasi prosedur tersimpan *spGetDataWithParam2* agar menerima *parameter* yang sesuai (misalnya, tipe data *int* untuk *ID* siswa). Filter data disesuaikan dengan tabel relasi tugas dan siswa.

Testing: Setelah pengujian dilakukan, data tugas muncul dengan benar sesuai siswa yang dipilih dari *dropdown*.

Implementation: Fungsi ini berhasil diterapkan dan diuji di halaman portal orang tua.

Output: Orang tua kini dapat melihat daftar tugas lengkap, termasuk status pengerjaan dan *deadline*-nya, untuk setiap anak.

3.2.7 Sejarah Kehadiran



Gambar 3.7. History Attendance

Gambar 3.7 menampilkan riwayat kehadiran siswa selama satu semester, mencakup informasi tanggal, mata pelajaran, dan status kehadiran. Status terbagi menjadi beberapa kategori: *Present*, *Tardy*, dan *Absence*. Data ditampilkan dalam bentuk tabel yang terstruktur, memudahkan pelacakan kedisiplinan siswa berdasarkan aktivitas dalam kelas.

A Iterasi Pertama

Requirement: Fitur ini ditujukan untuk menampilkan riwayat kehadiran siswa secara lengkap, tidak hanya berdasarkan hari ini, namun mencakup seluruh semester. Hal ini penting untuk evaluasi jangka panjang oleh orang tua.

Design and Development: Dibuat halaman baru dengan tabel kehadiran lengkap. Tabel ini menampilkan tanggal, status kehadiran (hadir, sakit, izin, alpa), dan catatan jika ada.

Testing: Pengujian menunjukkan bahwa halaman terlalu berat saat memuat banyak data sekaligus, terutama jika rentang waktu cukup panjang.

Implementation: Versi awal ditampilkan tetapi dengan performa yang belum optimal.

Output: Orang tua dapat melihat kehadiran siswa, namun masih mengalami keterlambatan dalam pemuatan data.

B Iterasi Kedua

Requirement: Diperlukan optimasi tampilan dan performa pemanggilan data agar riwayat kehadiran dapat diakses lebih cepat.

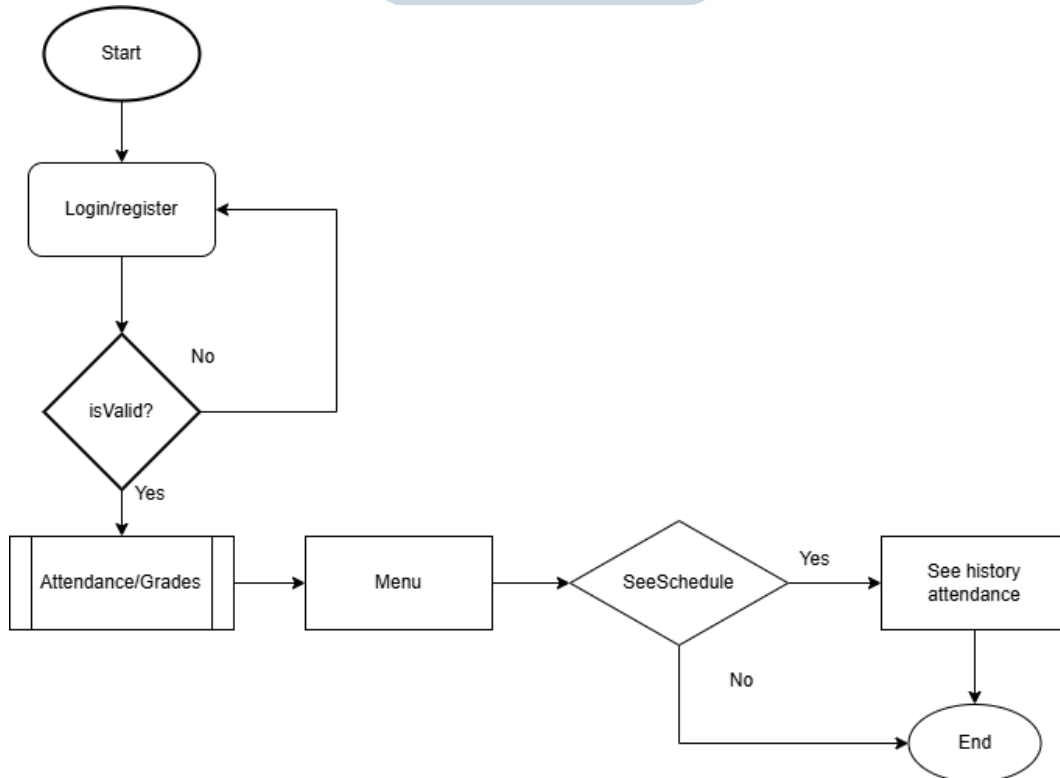
Design and Development: Penerapan fitur paginasi dan filter tanggal (misal: filter per bulan) ditambahkan pada tabel. Ini bertujuan agar jumlah data yang ditampilkan dapat dikontrol dan tidak memberatkan pemrosesan.

Testing: Hasil pengujian menunjukkan bahwa dengan adanya paginasi dan filter, data ditampilkan lebih cepat dan pengguna dapat menyaring data sesuai kebutuhan.

Implementation: Fitur ini diimplementasikan dan diuji lintas perangkat.

Output: Riwayat kehadiran siswa kini lebih mudah dibaca dan diakses dengan cepat, memberikan fleksibilitas bagi orang tua untuk melakukan pemantauan.

3.2.8 Jadwal



Gambar 3.8. Schedule

Gambar 3.8 merupakan jadwal harian siswa berdasarkan pilihan dari dropdown. Informasi yang ditampilkan mencakup hari, waktu, nama mata pelajaran, dan ruang kelas.

A Iterasi Pertama

Requirement: Tujuan dari fitur ini adalah menampilkan jadwal kelas harian/mingguan berdasarkan siswa yang dipilih, agar orang tua mengetahui aktivitas belajar anak setiap hari.

Design and Development: Halaman jadwal dibuat menggunakan GridView yang menampilkan jam pelajaran, mata pelajaran, dan guru pengajar. Data ditarik dari tabel jadwal menggunakan stored procedure yang menerima ID siswa.

Testing: Hasil awal menampilkan jadwal, namun tidak sesuai dengan siswa yang dipilih. Masalah ditemukan pada pengambilan parameter siswa.

Implementation: Versi pertama berjalan tetapi masih menunjukkan data jadwal dari siswa yang salah.

Output: Jadwal berhasil ditampilkan namun belum akurat.

B Iterasi Kedua

Requirement: Perlu memastikan pengambilan data berdasarkan siswa yang dipilih secara dinamis dari dropdown list.

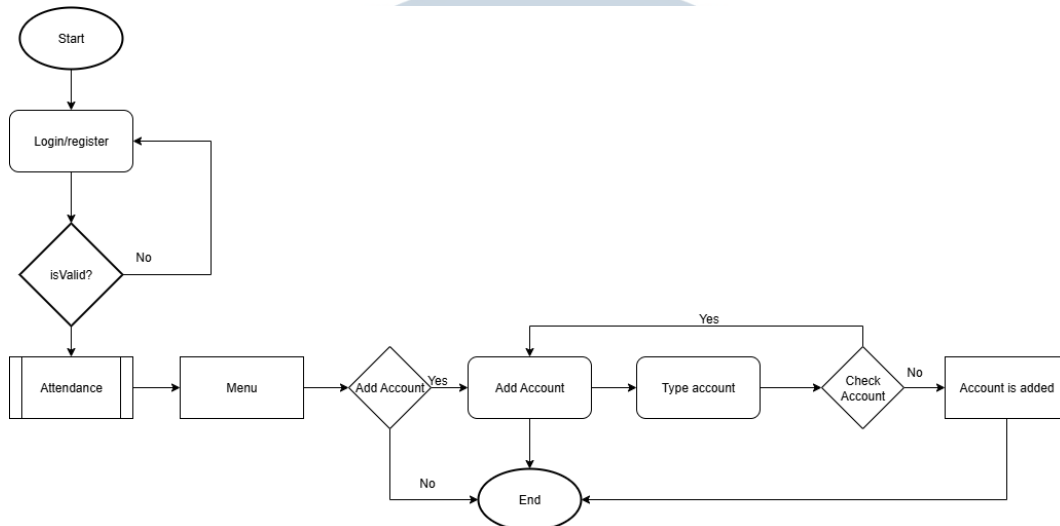
Design and Development: Perbaikan dilakukan pada bagian kode C# yang mengirim parameter ID siswa ke prosedur tersimpan. Dropdown siswa dihubungkan ke session login orang tua untuk menampilkan hanya anak-anak dari akun tersebut.

Testing: Setelah perbaikan, pengujian menunjukkan bahwa jadwal kini tampil sesuai siswa yang dipilih.

Implementation: Versi perbaikan ini diterapkan.

Output: Jadwal kelas kini dapat diakses berdasarkan siswa yang dipilih, menampilkan informasi yang sesuai dan berguna bagi orang tua.

3.2.9 Penambahan Akun



Gambar 3.9. Account

Gambar 3.9 digunakan oleh orang tua untuk mengelola data siswa yang ditambahkan ke akun *parent portal*. Fitur utama mencakup penambahan data siswa berdasarkan NIS serta penghapusan data siswa dari akun.

A Iterasi Pertama

Requirement: Fitur ini memungkinkan orang tua untuk menambahkan data anak-anak secara manual agar bisa dikaitkan ke dalam sistem.

Design and Development: Halaman manajemen akun dibuat dengan form input anak (nama, gender, kelas) dan tombol tambah/hapus anak. Data siswa disimpan dengan email induk sebagai identifikasi relasi.

Testing: Pengujian menemukan bahwa setelah menambahkan siswa, data tidak langsung muncul di dropdown halaman lain seperti Nilai atau Kehadiran.

Implementation: Siswa berhasil ditambahkan, namun tidak otomatis sinkron ke halaman lain.

Output: Orang tua dapat menambah siswa, namun sistem belum sepenuhnya sinkron antar halaman.

B Iterasi Kedua

Requirement: Perlu dilakukan sinkronisasi data siswa agar siswa yang baru ditambahkan dapat langsung muncul di halaman Nilai, Tugas, dan Kehadiran.

Design and Development: Kode backend pada halaman lain diperbarui agar mengambil data siswa berdasarkan email pengguna yang sedang login (bukan hanya ID siswa yang sudah tersimpan sebelumnya).

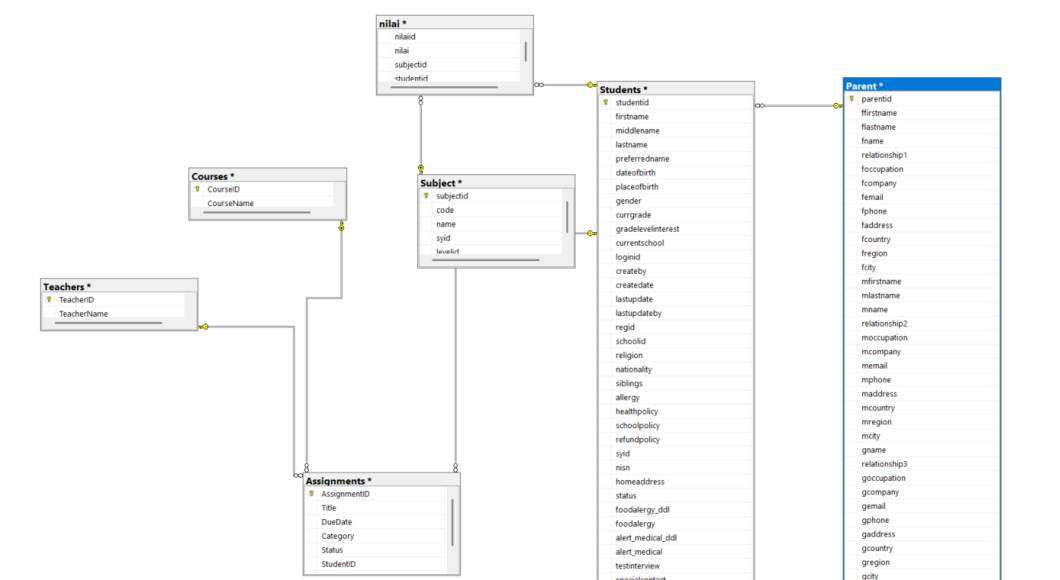
Testing: Setelah perbaikan, siswa yang ditambahkan langsung muncul di dropdown secara dinamis.

Implementation: Sinkronisasi berhasil dilakukan dan diuji pada seluruh modul.

Output: Semua halaman yang menampilkan data berdasarkan siswa kini secara otomatis memperbarui daftar siswa setelah orang tua menambahkan data anak.

3.2.10 Database Schema

Berikut adalah gambar dari *database* yang digunakan dari sistem informasi sekolah.



Gambar 3.10. Database

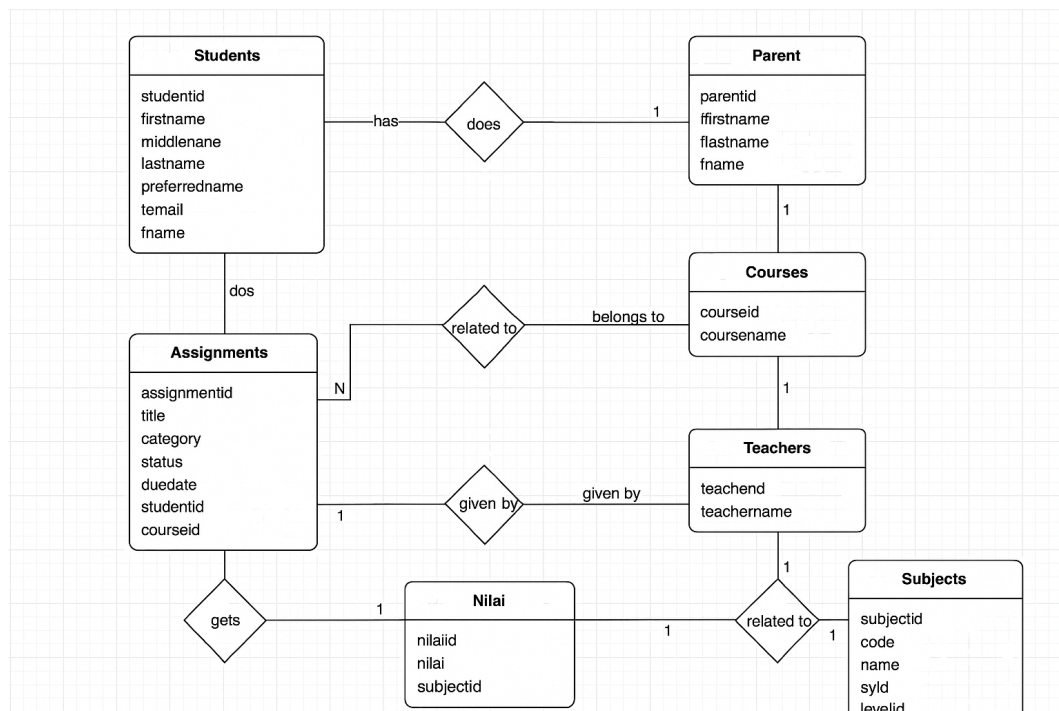
Gambar 3.10 menampilkan struktur basis data yang digunakan dalam

pengembangan sistem informasi sekolah ini. Basis data tersebut dirancang secara relasional untuk mendukung integrasi antar modul yang ada dalam sistem. Beberapa tabel utama yang digunakan meliputi *Students*, *Teachers*, *Courses*, nilai, *Assignment*, *Parent*, dan *Subject*.

Tabel *Students* menyimpan data siswa yang terdaftar dalam sistem, sedangkan tabel *Teachers* mencatat informasi mengenai guru yang mengajar. Tabel *Courses* digunakan untuk mengelola daftar mata pelajaran yang ditawarkan. Informasi mengenai nilai disimpan secara terstruktur untuk memudahkan pemantauan akademik, sementara tabel *Assignment* mencatat berbagai tugas yang diberikan oleh guru kepada siswa. Tabel *Parent* memuat informasi akun orang tua yang memiliki hak akses untuk memantau aktivitas belajar. Sebagai pelengkap, tabel *Subject* digunakan untuk mengelompokkan jenis mata pelajaran yang tersedia dalam sistem.

Struktur basis data ini menjadi fondasi utama dalam mendukung kinerja aplikasi agar berjalan secara efisien dan terorganisir, serta memungkinkan integrasi fitur-fitur penting yang menunjang kebutuhan pengguna.

3.2.11 ERD



Gambar 3.11. ERD

Gambar 3.11 menunjukkan rancangan Entity Relationship Diagram (ERD) untuk sistem informasi sekolah berbasis *web* yang dikembangkan. ERD ini menggambarkan hubungan antar entitas utama dalam sistem, yang terdiri dari **Students**, **Parents**, **Assignments**, **Courses**, **Teachers**, **Nilai**, dan **Subjects**. Berikut adalah penjelasan masing-masing entitas dan relasinya:

- **Students:** Menyimpan data siswa seperti *studentid*, *firstname*, *middlename*, *lastname*, *preferredname*, *temail*, dan *fname*. Setiap siswa dapat memiliki satu orang tua (relasi *has* dan *does*) dan dapat mengerjakan banyak tugas (relasi *does* ke **Assignments**).
- **Parent:** Menyimpan informasi orang tua dengan atribut seperti *parentid*, *firstname*, *lastname*, dan *fname*. Setiap orang tua dapat memiliki satu atau beberapa anak (*Students*).
- **Assignments:** Menyimpan data tugas yang diberikan kepada siswa. Memiliki atribut *assignmentid*, *title*, *category*, *status*, *duedate*, *studentid*, dan *courseid*. Tugas-tugas ini terkait dengan **Courses** (*related to*) dan diberikan oleh **Teachers** (*given by*).
- **Courses:** Menyimpan data mata pelajaran yang diambil siswa, terdiri dari *courseid* dan *coursename*. Mata pelajaran dimiliki oleh guru (**Teachers**) dan memiliki hubungan dengan tugas (**Assignments**).
- **Teachers:** Menyimpan data guru, terdiri dari *teacherid* dan *teachername*. Guru memberikan tugas kepada siswa dan mengajar suatu **Course**.
- **Nilai:** Menyimpan nilai siswa dengan atribut *nilaiid*, *nilai*, dan *subjectid*. Setiap nilai terhubung dengan entitas **Assignments** dan **Subjects**.
- **Subjects:** Menyimpan informasi mata pelajaran seperti *subjectid*, *code*, *name*, *syld*, dan *levelid*. Masing-masing nilai terkait dengan satu mata pelajaran.

Hubungan antar entitas menunjukkan bahwa sistem dirancang untuk melacak aktivitas belajar siswa secara komprehensif, mulai dari penugasan, nilai, hingga keterkaitan antara siswa, guru, mata pelajaran, dan orang tua. Desain ini mendukung kebutuhan pengawasan akademik secara *real-time* oleh orang tua melalui *Parent Portal*.