# BAB II
# KAJIAN LITERATUR

## 2.1 Sistem Informasi

Sistem informasi adalah gabungan dari beberapa komponen seperti proses, prosedur dan alat yang saling terkoordinasi yang digunakan untuk mengumpulkan, memproses, menyimpan, mengambil, menganalisis dan menyampaikan informasi untuk membantu mengambil solusi dari satu atau banyak masalah untuk mencapai tujuan dari organisasi [8].

Sistem informasi secara teknis dapat dipahami sebagai sekumpulan komponen yang saling terhubung untuk mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi guna mendukung pengambilan keputusan dalam sebuah organisasi [9]. Dengan menerapkan sistem informasi pada sekolah, pengelolaan data seperti absensi dan nilai dapat dilakukan secara lebih terstruktur dan efisien dibandingkan cara manual [5].

### 2.1.1 Fungsi dan Tujuan Sistem Informasi

Sistem informasi memiliki fungsi dan tujuan yang penting dalam menjalankan sebuah organisasi khusus nya pada sekolah seperti meningkatkan efisiensi operasional, mempermudah pengambilan keputusan dan pengendalian internal. Sistem informasi juga menyediakan manajemen data yang lebih terstruktur dibandingkan cara manual, hal ini memungkinkan organisasi untuk merespon secara efektif terhadap lingkungan kerja yang dinamis. Berikut adalah fungsi dan tujuan utama sistem informasi [10] :

1. Efisiensi operasional

   Sistem informasi membuat banyak pekerjaan lebih sederhana dan dapat dijalankan secara otomatis, hal tersebut yang membuat sistem informasi dapat mengurangi kesalahan manual dan memungkinkan karyawan fokus pada pekerjaan yang lebih membutuhkan tenaga manusia. Sistem informasi juga memungkinkan integrasi dan analisis data, membantu organisasi mengidentifikasi tren yang terjadi dilapangan kerja serta meningkatkan komunikasi dalam berkolaborasi melalui IS (Information Systems) sehingga koordinasi antar departemen lebih optimal yang akan menghasilkan alokasi sumber daya menjadi lebih baik dan pengurangan pemborosan [10].

2. Dukungan pengambilan keputusan

   Sistem informasi menyediakan data yang akurat dan dapat disalurkan tepat waktu, hal ini memungkinkan karyawan mengambil keputusan lebih cepat dan lebih baik dalam menyelesaikan masalah berdasarkan data yang telah tersedia. Sistem informasi juga

dapat membantu mengidentifikasi alternatif dan mengevaluasi hasil agar manajemen lebih efektif dalam mengambil keputusan [9].

3. Pengendalian internal

   Pada organisasi, pengendalian internal sangat penting agar organisasi tersebut dapat berjalan dengan lancar tanpa mengalami kerugian, sistem informasi menyediakan mekanisme pengendalian internal yang lebih baik dari pada dilakukan secara manual terutama pada pengelolaan data operasional sekolah, dengan memberikan visibilitas yang lebih tinggi terhadap aktivitas dan aliran data. Hal ini memungkinkan pengelolaan data lebih terstruktur, mencegah terjadinya kesalahan input, mendeteksi kesalahan pencatatan, dan memastikan kepatuhan terhadap peraturan dan kebijakan yang berlaku. Sistem informasi juga menyediakan alat untuk mengukur kinerja yang akurat seperti indikator kinerja utama dan analitik, memungkinkan organisasi memantau dan mengevaluasi setiap operasional yang ada. Fitur perbaikan berkelanjutan yang ada pada sistem informasi juga memungkinkan organisasi secara proaktif mengidentifikasi area yang perlu peningkatan dan penyesuaian yang dilakukan tepat waktu serta meningkatkan efisiensi proses secara keseluruhan. Dari hal-hal tersebut dapat disimpulkan bahwa penerapan sistem informasi yang efektif bisa menjadi fondasi penting dalam menjaga integritas operasional dan mencapai tujuan operasional organisasi secara berkelanjutan [10], [11]. Dalam konteks sekolah, pengendalian internal melalui sistem informasi dapat diterapkan pada pencatatan absensi dan nilai siswa agar data yang dihasilkan lebih akurat dan dapat dipertanggungjawabkan [5].

### 2.1.2 Komponen Sistem Informasi

Sistem informasi adalah gabungan dari komponen-komponen yang saling terhubung untuk mengumpulkan, menyimpan, memproses, dan menyajikan data sehingga menjadi informasi yang bermanfaat. Berikut penjelasan tentang komponen-komponen tersebut [12]:

1. Perangkat keras (*Hardware*)

   Perangkat keras atau *hardware* pada komponen sistem informasi adalah semua perangkat yang memiliki fisik dan digunakan dalam sistem informasi seperti server, komputer, perangkat penyimpanan, dan perangkat jaringan. Perangkat keras digunakan untuk membantu menjalankan perangkat lunak dan memproses informasi.

2. Perangkat lunak (*Software*)

   Perangkat lunak adalah program dan aplikasi yang dijalankan pada perangkat keras dengan fungsi tertentu. Dalam sistem informasi, perangkat lunak mencakup sistem operasi, basis data, dan perangkat lunak lainnya yang mendukung operasi sistem.

3. Basis data (*Database*)

   Basis data adalah kumpulan data yang terorganisir dan tersimpan secara sistematis, sehingga memudahkan akses, pengelolaan, dan pembaruan data. Basis data merupakan komponen penting pada sistem informasi untuk membantu proses penyimpanan, pengambilan, dan memanipulasi data. Beberapa sistem manajemen basis data yang populer dan sering digunakan untuk mengelola basis data yaitu MySQL, Oracle, dan Microsoft SQL Server.

4. Jaringan (*Network*)

   Jaringan yang dimaksud pada komponen sistem informasi adalah infrastruktur yang memungkinkan komunikasi dan pertukaran data antar perangkat dalam suatu organisasi. Jaringan digunakan untuk menghubungkan berbagai komponen sistem informasi dan menyalurkan informasi dengan efisien.

5. Prosedur (*Procedur*)

   Dalam konteks sistem informasi, prosedur merupakan langkah-langkah atau aturan yang terstruktur dan sistematis yang harus diikuti agar pengolahan data dapat menghasilkan informasi yang berguna. Prosedur yang baik akan menjaga dan memastikan kualitas hasil dari informasi yang dikelola.

6. Manusia (*Human*)

   Komponen ini merujuk pada pengguna sistem informasi, administrasi, pengembang, dan pemangku kepentingan lainnya yang terlibat dalam sistem informasi tersebut.

## 2.2 Analisis dan Perancangan Sistem

Dalam pengembangan sebuah sistem informasi, terdapat dua tahap fundamental yang harus dilalui sebelum proses implementasi dimulai, yaitu analisis sistem dan perancangan sistem. Kedua tahap ini merupakan pondasi utama yang menentukan arah dan kualitas sistem yang akan dibangun. Analisis sistem berfokus pada pemahaman mendalam terhadap kondisi dan kebutuhan yang ada, sedangkan perancangan sistem menerjemahkan hasil analisis tersebut ke dalam cetak biru teknis yang dapat diimplementasikan oleh pengembang. Subbab ini membahas tujuan dari analisis dan perancangan sistem, teknik perancangan menggunakan *Unified Modeling Language* (UML), serta metodologi pengembangan yang digunakan dalam penelitian ini, yaitu metode *Waterfall*.

### 2.2.1 Tujuan Analisis dan Perancangan Sistem

Analisis dan perancangan sistem informasi merupakan suatu proses yang kompleks yang digunakan untuk mengembangkan dan memelihara sistem informasi berdasarkan

tujuan, struktur, dan proses yang dimiliki oleh suatu organisasi [13]. Proses ini tidak hanya mencakup pemahaman teknis, tetapi juga melibatkan pemahaman mendalam terhadap kebutuhan organisasi dan pengguna yang akan menggunakan sistem tersebut.

Analisis sistem adalah penjabaran dari suatu sistem yang utuh ke dalam berbagai bagian komponennya dengan tujuan untuk mengidentifikasi dan mengevaluasi berbagai masalah atau hambatan yang muncul pada sistem yang sedang berjalan [5]. Dalam proses analisis, terdapat beberapa tahapan yang umumnya dilakukan, yaitu mengidentifikasi masalah, mengidentifikasi batasan masalah, menganalisis kebutuhan sistem, serta mendefinisikan komponen-komponen sistem yang perlu dibangun atau direvisi [13].

Adapun perancangan sistem merupakan tahap lanjutan setelah analisis, di mana pemahaman yang diperoleh dari proses analisis diterjemahkan ke dalam rancangan teknis melalui penggabungan kebutuhan perangkat lunak dan perangkat keras. Tujuan utama perancangan sistem adalah untuk memenuhi kebutuhan pengguna sistem serta memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada tim pengembang [13]. Dengan melakukan analisis dan perancangan yang matang, risiko kegagalan sistem dapat diminimalisir dan sistem yang dihasilkan akan lebih sesuai dengan kebutuhan pengguna di lapangan.

### 2.2.2 Teknik Perancangan (UML)

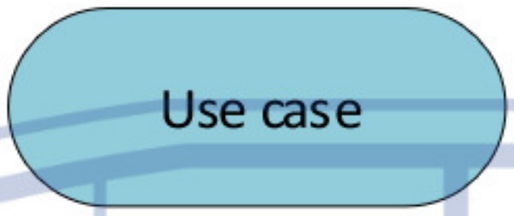
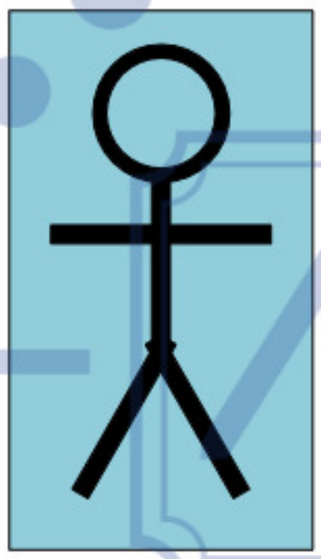
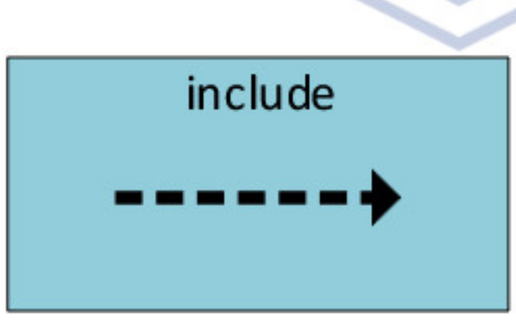
*Unified Modeling Language* (UML) adalah bahasa standar yang digunakan untuk memodelkan, mendeskripsikan, dan mengembangkan perangkat lunak dengan pendekatan berbasis objek. UML memberikan representasi visual yang memungkinkan pemahaman yang lebih jelas terhadap struktur dan perilaku suatu sistem. Dengan UML, proses analisis dan desain perangkat lunak menjadi lebih terstruktur, sehingga memudahkan komunikasi antara pengembang, analis, serta pemangku kepentingan lainnya. UML juga berfungsi sebagai alat bantu dalam mendokumentasikan sistem secara sistematis dan konsisten [14]. UML menyediakan berbagai jenis diagram yang masing-masing memiliki fungsi spesifik untuk menggambarkan aspek tertentu dari sistem. Adapun jenis diagram UML yang digunakan dalam penelitian ini adalah sebagai berikut [15]:

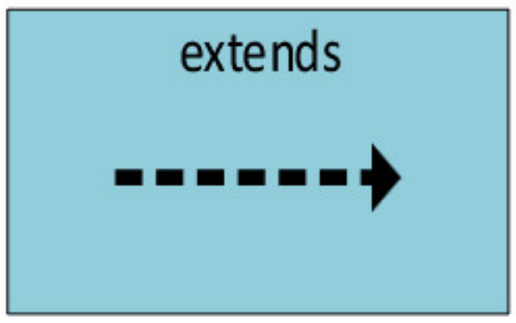
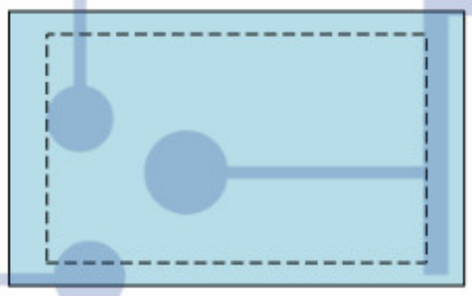
1. *Use Case Diagram*

   *Use case diagram* digunakan untuk mendeskripsikan interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibangun. Diagram ini berfungsi untuk menggambarkan kebutuhan sistem dari sudut pandang pengguna, termasuk berbagai fungsi atau layanan yang disediakan oleh sistem. Dengan *use case diagram*, pengembang

dapat mengidentifikasi fitur utama sistem serta bagaimana pengguna berinteraksi dengannya.

Table 2.1 Notasi Use Case Diagram

| No | Simbol | Nama Notasi | Keterangan |
|---|---|---|---|
| 1 |  | Use case | Menggambarkan fungsionalitas atau layanan yang disediakan oleh sistem kepada pengguna. |
| 2 |  | Aktor (Actor) | Entitas di luar sistem yang berinteraksi dengan sistem, dapat berupa pengguna atau sistem lain. |
| 3 |  | Asosiasi (Association) | Menunjukkan hubungan interaksi antara aktor dengan use case dalam sistem. |
| 4 |  | Include | Menunjukkan bahwa satu use case selalu memanggil use case lain sebagai bagian wajib dari eksekusinya. |

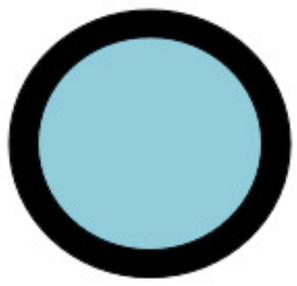
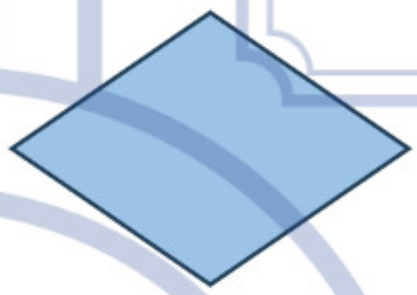
| | | | |
|---|---|---|---|
| 5 |  | Extend | Menunjukkan bahwa satu use case dapat memperluas perilaku use case lain secara kondisional. |
| 6 |  | Sistem (System Boundary) | Kotak yang menggambarkan batas ruang lingkup sistem dan memisahkan komponen dalam sistem dari aktor luar. |

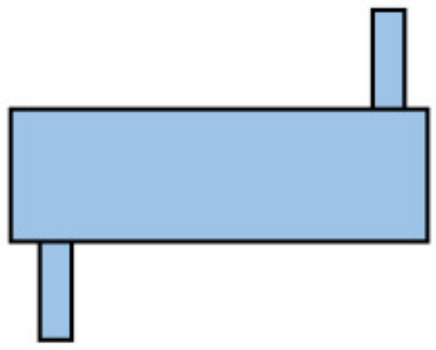
2. *Activity Diagram*

*Activity diagram* digunakan untuk memvisualisasikan alur kerja atau aktivitas dalam sistem perangkat lunak. Diagram ini menggambarkan bagaimana suatu proses berlangsung dari awal hingga akhir, termasuk kondisi percabangan dan keputusan yang terjadi selama proses. *Activity diagram* sering digunakan untuk menjelaskan urutan eksekusi proses bisnis atau aliran data dalam sistem sebagaimana dibahas lebih lanjut pada Sub Bab 2.2.3.

Table 2.2 Notasi Activity Diagram

| No | Simbol | Nama Notasi | Keterangan |
|---|---|---|---|
| 1 |  | Initial Node | Lingkaran hitam penuh yang menandai titik awal dari suatu alur aktivitas. |

| | | | |
|---|---|---|---|
| 2 |  | Final Node (Activity Final) | Lingkaran hitam di dalam lingkaran kosong yang menandai titik akhir dari seluruh alur aktivitas. |
| 3 |  | Action/Activity | Persegi panjang bersudut membulat yang menggambarkan sebuah aktivitas atau proses yang dilakukan. |
| 4 |  | Decision Node | Belah ketupat yang menggambarkan percabangan kondisi dengan dua atau lebih jalur keputusan. |
| 5 |  | Control Flow | Garis panah yang menunjukkan arah alur dari satu aktivitas ke aktivitas berikutnya. |

| | | | |
|---|---|---|---|
| 6 |  | Fork / Join | Garis tebal horizontal. Fork memisahkan alur menjadi beberapa alur paralel; Join menggabungkan kembali alur-alur paralel tersebut. |

### 2.2.3 Metodologi *Waterfall*

Metode *Waterfall* merupakan salah satu model *Software Development Life Cycle* (SDLC) yang paling umum digunakan dalam pengembangan sistem informasi. Model ini menggunakan pendekatan sistematis dan berurutan, di mana setiap tahapan harus diselesaikan secara tuntas sebelum dapat melanjutkan ke tahapan berikutnya [16]. Pendekatan linier ini memastikan bahwa setiap aspek sistem dikembangkan secara menyeluruh dan terencana, sehingga kualitas sistem yang dihasilkan dapat terjaga dengan baik. Tahapan-tahapan dalam metode *Waterfall* dapat dijabarkan sebagai berikut [16]:

1. Analisis Kebutuhan (*Requirement Analysis*)

   Tahap ini berfokus pada pemahaman mendalam terhadap masalah dan kebutuhan pengguna agar sistem yang dirancang dapat berjalan secara efisien. Pada penelitian ini, tahap analisis dilakukan melalui wawancara dan observasi langsung untuk memahami permasalahan pada proses absensi dan pengolahan nilai yang masih berjalan secara manual.

2. Perancangan Sistem (*System Design*)

   Pemahaman dari tahap analisis kebutuhan diterjemahkan ke dalam rancangan teknis sistem, meliputi perancangan antarmuka, struktur basis data, dan arsitektur aplikasi.

3. Implementasi (*Implementation*)

   Sistem yang telah dirancang pada tahap sebelumnya mulai dibangun dan dikodekan sesuai dengan spesifikasi yang telah ditetapkan.

4. Pengujian (*Testing*)

Program yang telah diimplementasikan diuji untuk memastikan bahwa sistem berfungsi sesuai dengan kebutuhan dan bebas dari kesalahan. Pada penelitian ini, pengujian dilakukan menggunakan metode *black box testing*.

5. Pemeliharaan (*Maintenance*)

   Tahap akhir yang mencakup perbaikan dan pembaruan sistem setelah sistem dioperasikan untuk memastikan sistem tetap berjalan secara optimal. Pada penelitian ini, tahap pemeliharaan tidak termasuk dalam ruang lingkup penelitian dan akan menjadi tanggung jawab pihak sekolah setelah sistem diserahkan.

Metode *Waterfall* dipilih dalam penelitian ini karena kebutuhan dan ruang lingkup sistem telah terdefinisi secara jelas sejak awal berdasarkan hasil analisis kebutuhan bersama mitra. Model ini lebih cocok digunakan untuk proyek yang bersifat generik, di mana seluruh kebutuhan sistem dapat diidentifikasi dari awal dengan spesifikasi yang jelas, dibandingkan dengan metode iteratif yang lebih sesuai untuk kebutuhan yang masih berubah-ubah [15].

## 2.3 Absensi dan Penilaian Sekolah

Pengelolaan data akademik di lingkungan sekolah mencakup dua proses inti yang saling berkaitan, yaitu pencatatan kehadiran siswa dan pengolahan nilai. Kedua proses ini menjadi tolok ukur utama dalam mengevaluasi perkembangan belajar siswa selama satu periode pembelajaran. Pemahaman terhadap konsep absensi, struktur mata pelajaran, dan sistem penilaian yang berlaku di sekolah menengah kejuruan menjadi landasan teori yang diperlukan dalam perancangan sistem informasi akademik yang sesuai dengan kebutuhan dan alur kerja institusi pendidikan.

### 2.3.1 Absensi Sekolah

Absensi merupakan suatu tanda kehadiran yang menjadi bagian dari aktivitas pelaporan dalam sebuah institusi pendidikan [17]. Dalam konteks persekolahan, absensi tidak hanya berfungsi sebagai dokumentasi administratif, tetapi juga berperan sebagai indikator kedisiplinan dan motivasi siswa dalam mengikuti proses belajar mengajar [17]. Suryosubroto menyatakan bahwa daftar hadir bertujuan untuk mengetahui frekuensi kehadiran siswa sekaligus menjadi sarana kontrol kerajinan belajar, sedangkan Mulyasa menambahkan bahwa kehadiran siswa di sekolah bertujuan untuk mengembangkan bakat, menjalin komunikasi, memahami materi, serta membentuk karakter siswa [18]. Pencatatan absensi yang masih dilakukan secara manual umumnya rentan terhadap kesalahan pencatatan, sulit diarsipkan, dan tidak memungkinkan pemantauan kehadiran secara *real-*

*time* [17], sehingga penerapan sistem absensi berbasis digital terbukti berkontribusi positif dalam mengurangi kesalahan dan mempercepat proses pengelolaan data kehadiran siswa [5].

### 2.3.2 Mata Pelajaran Sekolah

Dalam Kurikulum Merdeka yang diterapkan di Sekolah Menengah Kejuruan (SMK), pembelajaran intrakurikuler dibagi menjadi dua kelompok utama, yaitu kelompok mata pelajaran umum yang berfungsi membentuk peserta didik menjadi pribadi yang utuh, dan kelompok mata pelajaran kejuruan yang berfungsi membangun kompetensi sesuai kebutuhan dunia kerja [19]. Selain itu, terdapat Praktik Kerja Lapangan (PKL) sebagai mata pelajaran wajib yang dilaksanakan paling sedikit satu semester di kelas XII, bertujuan memberikan kesempatan kepada peserta didik untuk menerapkan kompetensi teknis secara langsung di lingkungan kerja nyata [19], [20]. Setiap mata pelajaran dalam Kurikulum Merdeka memiliki Capaian Pembelajaran (CP) yang ditetapkan oleh pemerintah, kemudian diturunkan oleh guru menjadi Tujuan Pembelajaran (TP) yang lebih spesifik sebagai dasar penilaian dan deskripsi capaian kompetensi siswa di rapor [7].

### 2.3.3 Penilaian Akademik

Penilaian akademik merupakan proses sistematis untuk mengukur tingkat pencapaian belajar peserta didik berdasarkan kompetensi yang telah ditetapkan dalam kurikulum [20]. Dalam Kurikulum Merdeka, penilaian terbagi menjadi tiga jenis, yaitu asesmen diagnostik untuk menganalisis kondisi awal peserta didik, asesmen formatif untuk memantau perkembangan selama proses pembelajaran, dan asesmen sumatif untuk menilai ketercapaian tujuan pembelajaran pada akhir periode [21]. Nilai akhir rapor diperoleh dari pengolahan komponen formatif dan sumatif, dilengkapi dengan deskripsi naratif capaian per Tujuan Pembelajaran, penilaian Projek Penguatan Profil Pelajar Pancasila (P5), serta penilaian ekstrakurikuler dalam bentuk predikat [21]. Proses pengolahan nilai yang masih dilakukan secara manual menggunakan spreadsheet terpisah oleh masing-masing guru terbukti tidak efisien karena keterlambatan satu guru dapat menghambat seluruh proses rekapitulasi nilai, sebagaimana yang telah dibuktikan oleh Khoirudhin & Supriyono (2017) bahwa penggantian proses manual berbasis spreadsheet dengan aplikasi desktop terbukti mampu memudahkan pengelolaan data nilai secara lebih terstruktur dan efisien [4]. Pada penelitian ini, sistem yang dikembangkan berfokus pada pengolahan komponen nilai formatif dan sumatif serta rekapitulasi nilai akhir rapor. Adapun penilaian Projek Penguatan Profil Pelajar Pancasila (P5) dan penilaian ekstrakurikuler tidak termasuk dalam ruang lingkup sistem yang dibangun.

## 2.4 Aplikasi Desktop

Dalam konteks pengembangan sistem informasi, pemilihan platform aplikasi merupakan salah satu keputusan teknis yang berpengaruh langsung terhadap keandalan dan kemampuan operasional sistem di lapangan. Salah satu jenis platform yang umum digunakan dalam lingkungan kelembagaan adalah aplikasi desktop. Aplikasi desktop adalah perangkat lunak yang dirancang untuk diinstal dan dijalankan secara lokal pada komputer pengguna, sehingga seluruh proses komputasi dan penyimpanan data dilakukan di dalam perangkat itu sendiri. Berbeda dengan aplikasi berbasis web yang memerlukan koneksi internet dan server jarak jauh untuk dapat beroperasi, aplikasi desktop memanfaatkan sumber daya perangkat keras secara langsung, yaitu CPU, memori, dan penyimpanan lokal, sehingga menghasilkan waktu respons yang lebih cepat dan stabil [1]. Aplikasi desktop dijalankan di atas sistem operasi yang terpasang pada perangkat, seperti Windows, macOS, atau Linux, dengan karakteristik utama berupa kemampuan beroperasi secara offline tanpa bergantung pada koneksi internet. Karakteristik ini menjadikan aplikasi desktop sebagai pilihan yang tepat untuk lingkungan dengan keterbatasan koneksi internet, seperti sekolah yang berada di daerah dengan infrastruktur jaringan yang belum memadai [1].

Setiap platform pengembangan perangkat lunak memiliki kelebihan dan kekurangan yang perlu dipertimbangkan secara cermat sesuai dengan kondisi dan kebutuhan pengguna. Dari sisi kelebihan, aplikasi desktop memiliki latensi yang sangat rendah karena seluruh pemrosesan data dilakukan langsung pada perangkat pengguna tanpa memerlukan komunikasi melalui jaringan, sehingga menghasilkan waktu respons yang cepat dan konsisten dalam berbagai kondisi. Berdasarkan pengujian empiris, sistem berbasis desktop menunjukkan keunggulan pada indikator kecepatan akses sebesar 96% dan keandalan sistem mencapai 97% dibandingkan sistem berbasis web [1]. Keandalan ini bersumber dari fakta bahwa operasional sistem tidak bergantung pada ketersediaan jaringan atau server eksternal, sehingga gangguan konektivitas tidak akan mempengaruhi jalannya sistem. Seluruh fungsi utama aplikasi tetap dapat dijalankan sepenuhnya meskipun koneksi internet tidak tersedia, sehingga aktivitas pengelolaan data dapat terus berlangsung tanpa hambatan dalam kondisi jaringan yang tidak stabil sekalipun [1].

Di sisi lain, aplikasi desktop juga memiliki sejumlah kekurangan yang perlu dipertimbangkan. Pertama, keterbatasan aksesibilitas, di mana aplikasi hanya dapat diakses dari perangkat tempat aplikasi diinstal sehingga pengguna tidak dapat menggunakan sistem dari perangkat lain secara langsung. Kedua, distribusi pembaruan yang lebih kompleks karena setiap pembaruan sistem perlu didistribusikan dan diinstal secara manual pada

masing-masing perangkat, berbeda dengan aplikasi berbasis web yang pembaruannya cukup dilakukan di sisi server. Ketiga, risiko kehilangan data akibat kerusakan perangkat, di mana data yang disimpan secara lokal rentan hilang secara permanen apabila terjadi kerusakan fisik perangkat atau kesalahan pengguna tanpa adanya mekanisme pencadangan yang memadai [1]. Dalam penelitian ini, kekurangan ketiga tersebut diatasi melalui penerapan mekanisme sinkronisasi data ke layanan cloud, sehingga data yang tersimpan secara lokal pada masing-masing perangkat pengguna tetap terlindungi dari risiko kehilangan akibat kerusakan perangkat maupun human error, sebagaimana dibahas lebih lanjut pada Sub Bab 2.5.

Berdasarkan karakteristik dan pertimbangan di atas, aplikasi desktop dipilih sebagai platform pengembangan sistem dalam penelitian ini karena dinilai paling sesuai dengan kondisi dan kebutuhan operasional di SMK Taruna Tekno Nusantara. Dari sisi infrastruktur, koneksi internet di lingkungan sekolah yang tidak stabil dan terbatas menjadikan kemampuan operasional offline sebagai kebutuhan utama yang hanya dapat dipenuhi secara andal oleh aplikasi desktop [1]. Dari sisi pengguna, aplikasi desktop dinilai lebih mudah dioperasikan oleh pengguna yang memiliki keterbatasan literasi digital karena interaksinya bersifat lokal, langsung, dan tidak memerlukan pengelolaan akun berbasis browser maupun koneksi internet yang berkelanjutan [37]. Dari sisi operasional kelembagaan, pendekatan desktop juga mengeliminasi kebutuhan terhadap biaya hosting server, biaya pemeliharaan infrastruktur jaringan secara berkelanjutan, serta kebutuhan akan tenaga ahli IT khusus untuk mengelola server, sehingga beban operasional sekolah dapat ditekan secara signifikan [38].

## 2.5 Sistem *Hybrid Online-Offline*

Pendekatan hybrid merupakan salah satu strategi teknis yang diterapkan dalam pengembangan sistem ini untuk menjawab tantangan operasional di lingkungan dengan keterbatasan koneksi internet. Sistem hybrid online-offline adalah sistem yang dirancang untuk dapat beroperasi dalam dua kondisi konektivitas secara bergantian, yaitu kondisi offline ketika koneksi internet tidak tersedia, dan kondisi online ketika koneksi internet tersedia [22]. Dalam kondisi offline, sistem tetap dapat menjalankan seluruh fungsi utamanya dengan menyimpan data secara lokal pada perangkat, sehingga aktivitas harian pengguna tidak terganggu meskipun jaringan internet sedang tidak dapat diakses. Sementara dalam kondisi online, sistem memanfaatkan koneksi internet yang tersedia untuk melakukan sinkronisasi data ke server atau layanan cloud [22].

Pendekatan hybrid ini berbeda dari sistem yang sepenuhnya bergantung pada koneksi internet (fully online) maupun sistem yang sama sekali tidak terhubung ke internet (fully offline). Sistem fully online tidak dapat beroperasi tanpa jaringan sehingga sangat rentan terhadap gangguan konektivitas, sedangkan sistem fully offline tidak memiliki mekanisme pencadangan data ke luar perangkat sehingga data berisiko hilang secara permanen apabila terjadi kerusakan perangkat. Sistem hybrid menggabungkan keunggulan dari keduanya, yaitu keandalan operasional dari pendekatan offline dan keamanan data dari pendekatan online, sehingga sistem dapat tetap beroperasi secara penuh sekaligus menjamin integritas data jangka panjang [23]. Kombinasi kedua pendekatan ini menjadikan arsitektur hybrid sebagai solusi yang tepat untuk lingkungan seperti SMK Taruna Tekno Nusantara yang memiliki keterbatasan koneksi internet namun tetap membutuhkan keamanan dan integrasi data antar pengguna.

Komponen kunci dalam penerapan pendekatan hybrid adalah mekanisme sinkronisasi data. Sinkronisasi data merupakan bagian dari teknik replikasi data terdistribusi, di mana data yang tersimpan di perangkat secara lokal sebagai replikasi parsial akan disinkronkan secara berkala dengan basis data di server pusat ketika koneksi tersedia, dengan tujuan menjaga ketersediaan sistem dan memastikan konsistensi informasi [24]. Secara umum, mekanisme sinkronisasi pada sistem hybrid online-offline melibatkan tiga tahap utama [22], [23]. Tahap pertama adalah pencatatan perubahan data (change tracking), di mana setiap perubahan data yang dilakukan oleh pengguna selama sistem beroperasi secara offline dicatat dan disimpan di dalam basis data lokal sebagai antrian yang akan diproses pada saat sinkronisasi berlangsung. Tahap kedua adalah deteksi koneksi internet, di mana sistem secara periodik memeriksa ketersediaan koneksi internet dan secara otomatis memulai proses sinkronisasi begitu koneksi terdeteksi tersedia tanpa memerlukan interaksi manual dari pengguna. Tahap ketiga adalah transfer dan integrasi data ke cloud, di mana data yang telah dicatat pada masing-masing perangkat pengguna selama mode offline dikirimkan ke layanan cloud yang berfungsi bukan sekadar sebagai cadangan, melainkan sebagai pusat integrasi data yang menyatukan seluruh masukan dari berbagai perangkat pengguna. Untuk mencegah terjadinya tumpang tindih atau konflik data saat sinkronisasi berlangsung, sistem menerapkan pembatasan hak akses berbasis peran (Role-Based Access Control) di mana setiap pengguna hanya dapat menyinkronkan data sesuai dengan wewenang yang dimilikinya, sehingga konflik data antar pengguna selama proses sinkronisasi dapat dihindari secara arsitektural [3]

Dalam penelitian ini, mekanisme sinkronisasi diterapkan dengan menggunakan SQLite sebagai basis data lokal dan Firebase sebagai layanan cloud, yang masing-masing dibahas lebih lanjut pada Sub Bab 2.7.

## 2.6 Basis Data

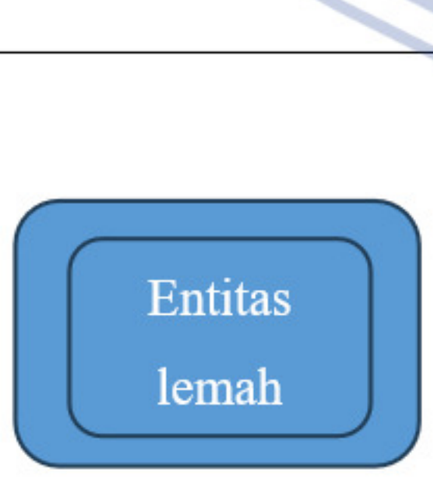
Dalam pengembangan perangkat lunak, terutama yang melibatkan pengelolaan volume informasi yang besar seperti sistem akademik, mekanisme penyimpanan informasi menjadi fondasi yang sangat krusial. Istilah database atau basis data pada dasarnya terbentuk dari perpaduan dua kata, yaitu "Basis" dan "Data". Basis dapat dimaknai sebagai pusat operasi atau lokasi utama yang berperan sebagai wadah penyimpanan informasi sekaligus pengelolaan sumber daya di dalam sebuah organisasi, sementara data merupakan representasi objektif dari berbagai fakta di dunia nyata yang menggambarkan suatu entitas, seperti individu (misalnya siswa, guru, atau staf), objek fisik, peristiwa, maupun kondisi tertentu [12]. Secara komprehensif, basis data didefinisikan sebagai sekumpulan data yang saling terhubung secara logis antar-elemen atau entitasnya, dan diorganisasikan dalam sebuah struktur yang sistematis [25]. Struktur arsitektur ini dirancang secara khusus dengan tujuan untuk mempermudah proses akses, pemrosesan, dan pemeliharaan data secara efisien, sehingga informasi yang dibutuhkan dapat diperoleh dengan cepat dan akurat sesuai kebutuhan sistem.

Untuk memahami bagaimana informasi didistribusikan dan diorganisasikan di dalam sebuah media penyimpanan, struktur dasar dari suatu basis data dapat diuraikan melalui tiga elemen arsitektur utama yang bekerja secara berkesinambungan [12], [25]. Elemen pertama adalah tabel (table atau file), yaitu elemen yang berisi sekumpulan data yang saling memiliki keterkaitan dan disusun ke dalam format tabular berupa susunan baris dan kolom [25]. Tabel berfungsi sebagai media utama untuk menyimpan dan memetakan data sesuai dengan kategorinya, di mana setiap baris merepresentasikan satu rekaman data yang bersifat unik, sedangkan setiap kolom mendefinisikan bidang atribut yang spesifik. Pada sistem akademik misalnya, data dikelompokkan ke dalam tabel-tabel terpisah seperti tabel profil siswa, tabel mata pelajaran, dan tabel rekam kehadiran absensi, yang masing-masing dapat dilengkapi dengan kunci utama (primary key) untuk menjaga integritas dan relasi antar data. Elemen kedua adalah field atau atribut, yaitu unit data pada tingkatan paling kecil di dalam basis data yang merepresentasikan satu atribut spesifik dan memiliki makna tertentu bagi penggunanya [25]. Setiap field menyimpan satu jenis informasi yang nantinya digabungkan untuk membentuk sebuah rekaman data yang lebih kompleks, seperti field Nomor Induk Siswa

(NIS), Nama Lengkap, dan Kelas pada pendataan siswa. Elemen ketiga adalah record atau rekaman, yaitu himpunan dari beberapa field yang disatukan karena memiliki keterhubungan informasi secara logis [12], di mana satu record mewakili satu entitas yang utuh dalam sebuah tabel, seperti keseluruhan data seorang siswa yang memuat NIS, Nama Lengkap, Kelas, dan Alamat secara bersamaan.

Dalam perancangan basis data, digunakan alat pemodelan bernama Entity Relationship Diagram (ERD), yaitu notasi grafis yang digunakan untuk menggambarkan struktur data serta hubungan antar entitas sebelum basis data diimplementasikan secara fisik. ERD berfungsi sebagai cetak biru awal yang memungkinkan pengembang memvisualisasikan relasi antar data secara terstruktur sehingga kesalahan logis dalam desain sistem dapat dihindari sejak dini. Dalam penelitian ini, ERD digunakan untuk merancang struktur basis data sistem absensi dan pengolahan nilai sebelum diimplementasikan ke dalam SQLite. ERD terdiri dari tiga komponen utama, yaitu entitas, atribut, dan relasi, yang masing-masing digambarkan dengan simbol tertentu [26]. Adapun simbol-simbol yang digunakan dalam penelitian ini disajikan pada tabel berikut:

Table 2.3 Simbol dan Notasi ERD

| No | Simbol | Nama Notasi | Keterangan |
|---|---|---|---|
| 1 |  | Entitas (Entity) | Objek nyata atau abstrak yang dapat diidentifikasi secara unik dalam sistem. Digambarkan dengan persegi panjang. |
| 2 |  | Entitas Lemah (Weak Entity) | Entitas yang tidak dapat diidentifikasi secara unik tanpa bergantung pada entitas lain. Digambarkan dengan persegi |

| | | | |
|---|---|---|---|
| | | | panjang di dalam persegi panjang. |
| 3 |  | Atribut (Attribute) | Karakteristik atau properti yang dimiliki oleh suatu entitas. Digambarkan dengan bentuk oval atau elips. |
| 4 |  | Atribut Kunci (Key Attribute) | Atribut yang nilainya unik dan dapat membedakan satu entitas dari entitas lainnya (primary key). Ditandai dengan nama atribut yang digarisbawahi di dalam oval. |
| 5 |  | Relasi (Relationship) | Hubungan yang terjadi antara dua atau lebih entitas. Digambarkan dengan belah ketupat (diamond). |
| 6 |  | Garis (Line) | Menghubungkan entitas dengan relasi, atau entitas dengan atributnya dalam diagram ERD. |
| 7 |  | Kardinalitas (Cardinality) | Menunjukkan jumlah hubungan |

| | | | |
|---|---|---|---|
| | | | antar entitas. Terdiri dari tiga jenis: one-to-one (1:1), one-to-many (1:N), dan many-to-many (M:N). |

Selain perancangan struktur data menggunakan ERD, tahapan penting lainnya dalam perancangan basis data adalah normalisasi. Normalisasi merupakan proses sistematis dalam merancang atau mereorganisasi skema tabel pada basis data relasional dengan tujuan mengurangi redundansi data dan menghindari anomali yang dapat terjadi saat proses penyisipan, pembaruan, maupun penghapusan data [36]. Proses ini dilakukan dengan cara menguraikan tabel yang memiliki struktur kompleks menjadi beberapa tabel yang lebih sederhana namun tetap saling terhubung secara relasional, sehingga integritas data dapat terjaga secara konsisten.

Normalisasi dilakukan secara bertahap melalui serangkaian aturan yang disebut bentuk normal (normal form). Bentuk normal pertama (1NF) mensyaratkan bahwa setiap kolom dalam tabel hanya boleh menyimpan nilai tunggal dan bersifat atomik, serta setiap baris harus bersifat unik [36]. Bentuk normal kedua (2NF) mensyaratkan bahwa tabel telah memenuhi 1NF dan setiap atribut non-kunci harus bergantung sepenuhnya pada seluruh kunci utama, bukan hanya sebagian darinya [36]. Bentuk normal ketiga (3NF) mensyaratkan bahwa tabel telah memenuhi 2NF dan tidak terdapat ketergantungan transitif, yaitu atribut non-kunci tidak boleh bergantung pada atribut non-kunci lainnya, melainkan harus bergantung langsung pada kunci utama [36]. Dalam penelitian ini, normalisasi diterapkan hingga bentuk normal ketiga (3NF) untuk memastikan struktur basis data yang dirancang bebas dari redundansi dan konsisten dalam mendukung operasional sistem absensi dan pengolahan nilai.

## 2.7 Teknologi yang digunakan

Dalam pengembangan sistem absensi dan pengolahan nilai pada penelitian ini, pemilihan teknologi yang tepat merupakan pertimbangan fundamental yang menentukan keandalan dan keberlanjutan sistem di lapangan. Setiap komponen teknologi dipilih berdasarkan kesesuaiannya dengan kebutuhan, khususnya kebutuhan sistem yang dapat beroperasi secara *offline* dengan mekanisme sinkronisasi data ke *cloud*. Sub bab ini

menguraikan tiga teknologi utama yang digunakan, yaitu *framework* Electron.js sebagai fondasi aplikasi *desktop*, SQLite sebagai basis data lokal, dan Firebase sebagai layanan *cloud* untuk sinkronisasi data.

### 2.7.1 Aplikasi Desktop Berbasis Web Technology

Aplikasi *desktop* berbasis *web technology* adalah pendekatan pengembangan perangkat lunak yang memanfaatkan teknologi web, yaitu HTML, CSS, dan JavaScript, untuk membangun aplikasi yang diinstal dan dijalankan secara lokal pada komputer pengguna sebagaimana layaknya aplikasi *desktop* konvensional [27]. Pendekatan ini berbeda dari paradigma pengembangan *desktop* tradisional yang mengharuskan pengembang mempelajari bahasa pemrograman khusus seperti C++ atau Java untuk setiap sistem operasi yang dituju.

*Framework* yang mengimplementasikan pendekatan ini dalam penelitian ini adalah Electron.js, sebuah *framework* sumber terbuka (*open-source*) yang dikembangkan dan dikelola oleh OpenJS Foundation [27]. Electron.js bekerja dengan menggabungkan dua komponen utama, yaitu mesin *rendering* Chromium yang berfungsi untuk menampilkan antarmuka berbasis HTML dan CSS, serta *runtime* Node.js yang memberikan akses penuh terhadap sumber daya sistem operasi seperti sistem berkas (*file system*), jaringan, dan memori perangkat [28]. Dengan arsitektur ini, kode yang ditulis menggunakan teknologi web dapat mengakses kapabilitas sistem tingkat rendah yang sebelumnya hanya dapat dilakukan oleh aplikasi *native*.

Secara arsitektural, Electron.js mengimplementasikan model multi-proses yang diturunkan dari Chromium, di mana struktur proses terbagi menjadi dua jenis utama [28]:

1. *Main Process*

   *Main process* berfungsi sebagai *entry point* aplikasi dan bertanggung jawab atas siklus hidup (*lifecycle*) aplikasi secara keseluruhan. Proses ini berinteraksi langsung dengan sistem operasi dan memiliki akses penuh ke API Node.js, termasuk operasi berkas, pengaturan jendela, dan komunikasi dengan layanan eksternal.

2. *Renderer Process*

   *Renderer process* bertugas untuk merender konten antarmuka pengguna di setiap jendela aplikasi menggunakan Chromium. Proses ini berjalan dalam konteks terisolasi dan berkomunikasi dengan *main process* melalui mekanisme komunikasi antar proses (*Inter-Process Communication*/IPC).

Penggunaan Electron.js menjadi relevan dalam penelitian ini karena sistem yang dibangun memerlukan akses langsung ke basis data lokal SQLite yang tersimpan di sistem

berkas komputer sekolah, sekaligus membutuhkan kemampuan melakukan koneksi jaringan untuk sinkronisasi data ke Firebase ketika internet tersedia. Kedua kapabilitas tersebut dapat dipenuhi sekaligus melalui Node.js yang terintegrasi di dalam Electron.js, tanpa mengorbankan kemampuan untuk membangun antarmuka pengguna yang responsif dan modern.

### 2.7.2 SQLite

SQLite adalah sebuah *library* yang mengimplementasikan mesin basis data relasional yang bersifat *self-contained*, *serverless*, *zero-configuration*, dan *transactional* [29]. Berbeda dengan sistem manajemen basis data relasional (*RDBMS*) konvensional seperti MySQL atau PostgreSQL yang beroperasi sebagai proses server tersendiri dan memerlukan konfigurasi serta administrasi yang kompleks, SQLite tidak memerlukan proses server yang berjalan secara terpisah. Seluruh basis data disimpan dalam satu berkas tunggal di sistem berkas komputer, sehingga proses pemasangan dan konfigurasi menjadi sangat sederhana [30].

Karakteristik utama SQLite yang membedakannya dari RDBMS lainnya adalah sebagai berikut [29], [30]:

1. *Serverless* SQLite tidak memerlukan proses server tersendiri untuk beroperasi. Aplikasi mengakses basis data secara langsung melalui pemanggilan fungsi dari *library* SQLite, sehingga tidak ada *overhead* komunikasi jaringan antar proses.
2. *Zero-configuration* SQLite tidak memerlukan instalasi, konfigurasi server, atau pemberian izin (*permission*) akun basis data. Basis data baru dapat dibuat cukup dengan membuat berkas baru di sistem berkas, menjadikannya sangat praktis untuk pengembangan dan *deployment*.
3. *Self-contained* SQLite hanya memiliki sedikit dependensi terhadap *library* eksternal maupun sistem operasi. Hal ini menjadikan SQLite sangat portabel dan dapat berjalan di berbagai platform tanpa modifikasi.
4. Ringan dan Efisien SQLite memiliki ukuran kode pustaka yang relatif kecil namun tetap mendukung hampir semua fitur SQL standar, termasuk transaksi ACID (*Atomicity, Consistency, Isolation, Durability*) yang menjamin integritas data [30]. Dalam pengujian performa terhadap dataset berskala besar, SQLite menunjukkan keunggulan dalam mengeksekusi query transaksional seperti *insert*, *update*, *delete*, dan *select* dibandingkan alternatif basis data embedded lainnya.

SQLite dipilih sebagai basis data lokal dalam penelitian ini karena kesesuaiannya yang tinggi dengan arsitektur sistem yang dirancang. Dalam pendekatan *hybrid offline-first*

yang diterapkan, seluruh transaksi data, mulai dari pencatatan absensi harian oleh wali kelas hingga pengisian nilai oleh guru, disimpan terlebih dahulu ke basis data SQLite yang terpasang pada masing-masing perangkat laptop guru atau wali kelas sebagai media penyimpanan lokal utama dalam kondisi offline [2]. Data yang terfragmentasi di setiap perangkat ini kemudian disinkronkan dan diintegrasikan secara terpusat ke Firebase ketika koneksi internet tersedia. Dengan demikian, tidak ada ketergantungan terhadap koneksi internet dalam alur kerja utama sistem.

### 2.7.3 Firebase

Firebase adalah sebuah platform *Backend-as-a-Service* (BaaS) yang dikembangkan oleh Google, yang menyediakan berbagai layanan *back-end* berbasis *cloud* yang dapat diintegrasikan langsung ke dalam aplikasi tanpa perlu membangun dan mengelola infrastruktur server secara mandiri [31]. Firebase dirancang untuk mempermudah pengembang dalam membangun aplikasi yang memerlukan fungsionalitas *back-end* seperti autentikasi pengguna, penyimpanan data, dan distribusi berkas, dengan cara mengekspos layanan-layanan tersebut melalui *Software Development Kit* (SDK) yang dapat digunakan langsung di sisi klien [31].

Dalam konteks penelitian ini, layanan Firebase yang digunakan adalah Firebase Realtime Database, sebuah basis data *cloud* NoSQL yang menyimpan data dalam format pohon JSON (*JSON tree*) dan menyinkronkan data antar klien secara *real-time* [22]. Data pada Firebase Realtime Database disimpan dalam format JSON (*JavaScript Object Notation*) dan dapat diakses dari berbagai perangkat melalui koneksi internet. Adapun karakteristik utama Firebase Realtime Database yang relevan dengan penelitian ini adalah sebagai berikut [2], [22]:

1. Sinkronisasi *Real-time* Firebase Realtime Database secara otomatis menyebarkan perubahan data ke semua klien yang terhubung secara *real-time* tanpa memerlukan *polling* atau permintaan berulang dari klien. Kapabilitas ini mendukung mekanisme sinkronisasi yang efisien antara basis data lokal SQLite di perangkat klien dan penyimpanan *cloud*.
2. Keamanan Berbasis Aturan (*Rules-based Security*). Sebagai lapisan keamanan pada sisi cloud, Firebase mengimplementasikan sistem kontrol akses berbasis aturan (Role-Based Access Control) yang memungkinkan pengembang mendefinisikan secara granular siapa yang diizinkan membaca atau menulis data pada node tertentu di basis data. Mekanisme ini melengkapi pembatasan akses yang telah diterapkan pada sisi aplikasi desktop,

sehingga validasi hak akses berbasis peran (RBAC) berlaku secara konsisten baik di sisi klien maupun di sisi server [3]..

3. Skalabilitas dan Keandalan Infrastruktur Google Sebagai produk Google, Firebase dikelola di atas infrastruktur *cloud* Google yang memiliki keandalan dan skalabilitas tinggi, sehingga ketersediaan data cadangan dapat terjamin tanpa beban pengelolaan server [31].

Firebase dipilih dalam penelitian ini bukan sekadar sebagai media pencadangan, melainkan sebagai basis data pusat (*Aggregator*) yang mengintegrasikan data dari banyak klien (laptop guru). Risiko kehilangan dapat dimitigasi secara signifikan dengan mengirimkan salinan data secara otomatis dari setiap perangkat lokal ke Firebase setiap kali koneksi internet tersedia, sebagaimana yang telah diuraikan dalam Sub Bab 2.5.

## 2.8 Pengujian Black Box

*Black box testing* atau yang dikenal juga sebagai *behavioral testing* merupakan metode pengujian perangkat lunak yang berfokus pada evaluasi fungsionalitas sistem tanpa mempertimbangkan struktur internal maupun kode program yang mendasarinya. Dalam pendekatan ini, penguji berinteraksi dengan sistem semata-mata berdasarkan spesifikasi dan persyaratan fungsional yang telah ditetapkan, sehingga tidak diperlukan pengetahuan tentang cara kerja sistem di dalamnya. Tujuan utama dari metode ini adalah untuk memastikan bahwa setiap fungsi perangkat lunak menghasilkan *output* yang sesuai terhadap *input* yang diberikan sesuai dengan kebutuhan dan harapan pengguna [32].

Pada penelitian ini, *black box testing* dipilih sebagai metode pengujian karena mampu memvalidasi fungsionalitas aplikasi sistem absensi dan pengolahan nilai dari sudut pandang pengguna akhir, yaitu guru, wali kelas, maupun admin, tanpa memerlukan pemahaman teknis mendalam terhadap kode sumber aplikasi yang dikembangkan.

### 2.8.1 Teknik Black Box Testing

Dalam pelaksanaannya, *black box testing* mencakup berbagai teknik pengujian yang dapat diterapkan sesuai dengan karakteristik sistem yang diuji [33]. Pada penelitian ini digunakan tiga teknik, yaitu:

1. Equivalence Partitioning

   *Equivalence partitioning* adalah teknik pengujian yang membagi data masukan (*input*) menjadi dua kategori: nilai yang *valid* dan nilai yang tidak *valid* (*invalid*). Pengujian dilakukan secara terpisah untuk masing-masing kategori, dengan asumsi bahwa semua nilai dalam satu kategori akan menghasilkan perilaku sistem yang sama. Dengan cara

ini, penguji dapat mereduksi jumlah kasus uji yang diperlukan tanpa mengurangi cakupan pengujian secara keseluruhan. Misalnya, pada fitur pencatatan absensi, kategori *valid* mencakup kode status yang dikenali sistem seperti Hadir, Sakit, Izin, dan Tanpa Keterangan, sedangkan kategori *invalid* mencakup input kosong atau kode status yang tidak dikenali sistem.

2. Boundary Value Analysis

   *Boundary value analysis* adalah teknik pengujian yang memfokuskan pengujian pada nilai-nilai di sekitar batas bawah dan batas atas dari suatu variabel *input*. Teknik ini didasari oleh kenyataan bahwa kesalahan pada perangkat lunak paling sering terjadi di sekitar nilai batas, bukan di tengah rentang nilai. Misalnya, pada fitur penginputan nilai akademik yang memiliki rentang 0 hingga 100, pengujian akan difokuskan pada nilai-nilai batas seperti −1, 0, 1, 99, 100, dan 101, untuk memastikan sistem menangani batas rentang dengan benar dan menolak nilai di luar rentang yang telah ditetapkan.

3. Use Case Testing

   *Use case testing* adalah teknik pengujian yang mensimulasikan skenario penggunaan nyata dari awal hingga akhir berdasarkan *use case* yang telah didefinisikan. Teknik ini berfokus pada alur interaksi pengguna dengan sistem secara menyeluruh, sehingga dapat memastikan bahwa seluruh fungsionalitas yang diharapkan berjalan dengan baik dalam konteks penggunaan sesungguhnya. Misalnya, skenario wali kelas yang melakukan pencatatan absensi harian, mulai dari login, memilih kelas dan tanggal, mengisi status kehadiran seluruh siswa, hingga menyimpan data, diuji secara end-to-end untuk memverifikasi bahwa seluruh rangkaian proses tersebut menghasilkan output yang sesuai dengan harapan.

### 2.8.2 Keuntungan dan Tantangan Black Box Testing

Penerapan *black box testing* pada pengembangan perangkat lunak memiliki sejumlah kelebihan dan tantangan sebagai berikut [34]:

1. Keuntungan *black box testing*

   a. Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman

      Salah satu kelebihan utama dari *black box testing* adalah bahwa penguji tidak perlu memiliki latar belakang teknis atau pemahaman mendalam tentang bahasa pemrograman yang digunakan dalam pengembangan perangkat lunak. Hal ini memungkinkan lebih banyak pihak, termasuk mereka yang memiliki latar belakang non-teknis, untuk turut berpartisipasi dalam proses pengujian.

b. Pengujian dilakukan dari sudut pandang pengguna

*Black box testing* berfokus pada bagaimana pengguna berinteraksi dengan perangkat lunak. Dengan pendekatan ini, penguji dapat mengidentifikasi inkonsistensi, *bug*, atau masalah *usability* yang mungkin tidak terlihat jika pengujian dilakukan dari sudut pandang teknis. Hal ini membantu memastikan bahwa perangkat lunak memenuhi kebutuhan dan harapan pengguna akhir.

c. Penguji dan pengembang dapat bekerja secara independen

Dalam proses *black box testing*, terdapat pemisahan yang jelas antara peran penguji dan pengembang. Penguji dan pengembang dapat bekerja secara independen dan paralel tanpa saling mengganggu satu sama lain. Hal ini memungkinkan pengujian dilakukan secara bersamaan dengan pengembangan, yang dapat mempercepat proses pengembangan perangkat lunak secara keseluruhan.

d. Penguji tidak perlu memeriksa kode

*Black box testing* tidak memerlukan penguji untuk memeriksa atau memahami kode sumber perangkat lunak. Ini mengurangi beban kerja penguji dan memungkinkan mereka untuk fokus pada pengalaman pengguna serta hasil akhir dari perangkat lunak yang diuji.

2. Tantangan *black box testing*

a. Kemungkinan kesalahan tidak terdeteksi

Salah satu kekurangan dari *black box testing* adalah bahwa pengujian ini mungkin tidak cukup teliti untuk mendeteksi semua kesalahan. Tanpa pengetahuan teknis yang mendalam, penguji berpotensi melewatkan *bug* yang bersifat kompleks atau masalah yang hanya dapat diidentifikasi melalui analisis kode program secara langsung.

b. Bagian *back-end* yang tidak diuji

*Black box testing* sering kali lebih fokus pada antarmuka pengguna dan interaksi pengguna dengan perangkat lunak. Akibatnya, bagian *back-end*, seperti logika bisnis, operasi *database*, dan mekanisme sinkronisasi data, berpotensi tidak teruji secara menyeluruh. Ini dapat menyebabkan masalah yang tidak terdeteksi di bagian yang tidak terlihat oleh pengguna.

c. Cakupan pengujian bergantung pada kelengkapan spesifikasi

Kualitas dan cakupan *black box testing* sangat bergantung pada kelengkapan spesifikasi kebutuhan yang telah didefinisikan sebelumnya. Apabila spesifikasi tidak lengkap atau ambigu, kasus uji yang dihasilkan pun tidak akan komprehensif, sehingga ada kemungkinan fungsionalitas tertentu tidak tercakup dalam pengujian.

### 2.8.3 Tahapan Pengujian Black Box

Pelaksanaan *black box testing* dilakukan melalui serangkaian tahapan yang sistematis guna memastikan bahwa seluruh fungsionalitas sistem diuji secara terstruktur dan komprehensif. Tahapan-tahapan tersebut adalah sebagai berikut [35]:

1. Analisis Kebutuhan Fungsional

   Tahap pertama adalah mengidentifikasi seluruh fungsionalitas sistem yang akan diuji berdasarkan kebutuhan yang telah didefinisikan sebelumnya. Pada penelitian ini, analisis kebutuhan fungsional dilakukan dengan mengacu pada use case yang telah dirancang, mencakup seluruh fitur sistem absensi dan pengolahan nilai seperti pencatatan kehadiran siswa, penginputan nilai akademik, dan pengelolaan data oleh admin.

2. Perancangan Kasus Uji (Test Case Design)

   Berdasarkan hasil analisis kebutuhan fungsional, disusun kasus uji (test case) yang mencakup skenario input beserta output yang diharapkan untuk setiap fungsi sistem. Perancangan test case dilakukan menggunakan teknik equivalence partitioning, boundary value analysis, dan use case testing sebagaimana telah diuraikan pada subbab 2.9.1.

3. Eksekusi dan Verifikasi

   Pada tahap ini, setiap kasus uji yang telah dirancang dijalankan secara langsung pada sistem. Penguji memasukkan data input sesuai skenario yang telah ditentukan dan mencatat output aktual yang dihasilkan sistem, kemudian membandingkannya dengan output yang diharapkan.

4. Evaluasi dan Pelaporan

   Tahap akhir adalah mengevaluasi keseluruhan hasil pengujian dengan membandingkan output aktual terhadap output yang diharapkan pada setiap kasus uji. Hasil evaluasi didokumentasikan dalam bentuk tabel pengujian yang memuat kolom skenario uji, data masukan, hasil yang diharapkan, hasil aktual, dan kesimpulan berupa status lulus atau tidak lulus.

## 2.9 SMK Taruna Tekno Nusantara

SMK Taruna Tekno Nusantara adalah sebuah institusi pendidikan yang berdiri pada tahun 2015 dengan nomor NPSN sekolah 70035993 berada dalam naungan Yayasan Taruna Tekno Arifah dengan kementrian pembina yaitu Kementrian Pendidikan Dasar dan Menengah, terletak di Desa Ujung Labuhen, Kecamatan Namo Rambe, Kabupaten Deli Serdang, Provinsi Sumatera Utara [6]. Sebagai sekolah menengah kejuruan, SMK Taruna

Tekno Nusantara menawarkan empat program keahlian utama yaitu Teknik Komputer dan Jaringan (TKJ), Rekayasa Perangkat Lunak (RPL), Penerbangan, dan Ketarunaan. Berikut merupakan rincian lebih lanjut tentang sekolah ini:

1. Kurikulum: SMK Taruna Tekno Nusantara saat ini menggunakan kurikulum Merdeka, yaitu kurikulum yang memberikan keleluasaan kepada pendidik untuk menerapkan pembelajaran yang lebih mendalam, sesuai dengan kebutuhan peserta didik, serta fokus pada penguatan karakter [7].
2. Fasilitas: SMK Taruna Tekno Nusantara dilengkapi dengan berbagai fasilitas yang menunjang proses pembelajaran siswa,antara lain ruang kelas, laboratorium komputer, dan lapangan olahraga.

Gambar 2.1 SMK Taruna Tekno Nusantara