

BAB II

LANDASAN TEORI

2.1 Definisi Sistem

Sistem adalah sekumpulan elemen atau komponen yang saling berhubungan, bekerja sama, dan berinteraksi untuk mencapai suatu tujuan tertentu. Sistem terbentuk dari beberapa subsistem yang saling mendukung, sehingga jika ada bagian yang tidak berjalan dengan baik maka tujuan sistem secara keseluruhan akan terganggu. (Muhammad Halmi Dar Sistem - Google Scholar, n.d.)

Dalam penelitian Perancangan aplikasi absensi berbasis Android di SD Mis Baiturrahman Labuhanbatu selatan, tahapan pemrosesan data dilakukan secara langsung. Data yang diperoleh kemudian melalui tahapan Masukkan (input), pengolahan (processing), keluaran (output). (Pratama et al., 2023)



Gambar 2.1 Aliran Suatu Sistem Gambar

2.2 Sistem Absensi Konvensional dan Modern

Sistem absensi konvensional umumnya mengacu pada metode pencatatan kehadiran yang dilakukan secara manual atau dengan menggunakan teknologi sederhana, tanpa bantuan sistem digital terintegrasi.

Menurut Kamel et al. (2020), absensi adalah proses pencatatan data yang bertujuan untuk mengetahui tingkat kehadiran seseorang dalam suatu kegiatan, atau lingkungan kerja. Absensi konvensional merupakan kegiatan pencatatan kehadiran yang dilakukan secara manual untuk mengetahui seseorang hadir atau tidak hadir.

Sistem absensi ini menggunakan metode secara manual untuk mencatat kehadiran seseorang, metode yang digunakan antara lain sebagai berikut:

1. Buku absensi digunakan guru untuk mencatat kehadiran siswa dengan menandatangani atau menceklis nama siswa dibuku absensi. Metode sederhana ini mudah dipahami, tetapi rentan terjadi kesalahan dalam penulisan, kehilangan buku, dan manipulasi data. Proses perhitungan kehadiran juga memakan waktu.
2. Daftar hadir umumnya sama dengan buku absensi, tapi biasanya berupa lembaran kertas yang dibagikan kepada siswa untuk ditandatangani. Metode ini juga rentan terjadi kesalahan dan kehilangan data.
3. Kartu absensi, meskipun umumnya kurang digunakan di sekolah dasar, ada beberapa sekolah yang menggunakan kartu absensi sederhana yang telah diisi oleh guru. Metode ini lebih sedikit terorganisir dari pada daftar hadir, tetapi masih rentan terjadi kesalahan kehilangan kartu absensi dan kesalahan dalam pencatatan kehadiran.

Sedangkan sistem absensi modern adalah sistem pencatatn kehadiran yang menggunakan teknologi digital dan otomatis untuk meningkatkan efesiensi, akurasi, dan keamanan data kehadiran.

Menurut Syahputra (2023), sistem absensi modern adalah mekanisme pencatatan kehadiran, waktu masuk, dan waktu keluar secara sistematis yang memanfaatkan teknologi digital, seperti aplikasi *mobile*, dan sistem berbasis web. Sistem ini dirancang untuk meningkatkan efesiensi dan akurasi dalam proses pengelolaan kehadiran pegawai, dan meminimalkan manipulasi data terhadap kesalahan pencatatan.

Meskipun sistem ini sederhana dan mudah dipahami, tetapi sistem absensi modern jauh lebih efektif disbanding dengan sistem absensi modern dalam hal akurasi, efesiensi, dan kemudahan penggunaan, keamanan data.

2.2.1 Karakteristik Sistem

Bentuk umumnya sebuah sistem terdiri dari *input*, *processing*, *output*. Adapun karakteristik suatu sistem sebagai berikut:

1. **Komponen Sistem (*Components*)**

Sistem terdiri dari beberapa elemen atau bagian yang saling berinteraksi dan bekerja sama membentuk satu kesatuan. Komponen ini bisa berupa subsistem yang menjalankan fungsi tertentu dan saling mempengaruhi dalam keseluruhan sistem.

2. **Batasan Sistem (*Boundary*)**

Batasan sistem adalah garis pemisah yang membedakan antara sistem dengan sistem yang lain. Batas ini menentukan ruang lingkup sistem

sehingga sistem dapat dipandang sebagai satu kesatuan yang tidak terpisahkan.

3. Lingkungan Luar Sistem (*Environment*)

Lingkungan luar adalah segala sesuatu di luar batas sistem yang dapat mempengaruhi operasi sistem, baik secara positif maupun negatif. Lingkungan ini penting untuk diperhatikan agar sistem dapat berfungsi dengan baik dan berkelanjutan.

4. Penghubung Sistem (*Interface*)

Penghubung adalah media atau sarana yang menghubungkan satu subsistem dengan subsistem lainnya, memungkinkan aliran sumber daya, informasi, atau energi dari satu bagian ke bagian lain dalam sistem.

5. Masukan Sistem (*Input*)

Masukan (*Input*) adalah energy yang dimasukkan ke dalam sistem tersebut.

6. Pengolahan Sistem (*Process*)

Pengelolaan sistem adalah suatu bagian pengolah yang akan merubah masukan menjadi keluaran.

7. Keluaran Sistem (*Output*)

Keluaran sistem adalah hasil dari energi yang telah diolah dan diproses menjadi keluaran yang berguna.

8. Sasaran atau Tujuan Sistem (*Objective/Goal*)

Setiap sistem memiliki tujuan atau sasaran yang ingin dicapai, keberhasilan sistem diukur dari tercapainya tujuan tersebut.

2.2.2 Klasifikasi Sistem

Sistem itu seperti kumpulan bagian yang saling berhubungan dan bekerja bersama untuk mencapai tujuan tertentu. Sistem bisa dibedakan menjadi beberapa jenis berdasarkan ciri-cirinya, yaitu:

1. **Sistem Fisik dan Sistem Abstrak**

Sistem fisik adalah sistem yang bisa dilihat dan sentuh, seperti komputer, mobil, atau pabrik. Sedangkan sistem abstrak adalah sistem yang berupa ide-ide atau konsep, misalnya sistem kepercayaan.

2. **Sistem Alam dan Sistem Buatan Manusia**

Sistem alam adalah sistem yang terjadi melalui sebuah proses alam, tidak dibuat oleh manusia, misalnya ekosistem hutan atau siklus air. Sedangkan sistem buatan manusia adalah sistem yang dibuat oleh manusia, seperti sistem sekolah, sistem komputer, atau sistem transportasi.

3. **Sistem Terbuka dan Sistem Tertutup**

Sistem terbuka adalah sistem yang selalu berinteraksi dengan lingkungan sekitar, menerima input dari luar dan memberikan output ke luar, misalnya tubuh manusia yang menerima makanan dan udara dari lingkungan. Sedangkan sistem tertutup adalah sistem yang tidak berhubungan dengan lingkungan luar dan tidak ada pengaruh dari luar ke

dalam sistem. Contohnya sulit ditemukan di kehidupan nyata, tapi bisa seperti mesin yang bekerja sendiri tanpa gangguan dari luar.

4. Sistem tertentu dan Sistem tak tertentu

Sistem tertentu adalah kumpulan bagian atau elemen-elemen yang sudah jelas dan spesifik, yang saling berhubungan dan bekerja sama untuk mencapai suatu tujuan yang sudah ditentukan.

Sistem tak tertentu adalah sistem yang masa depannya tidak bisa diprediksi dengan pasti karena terdapat unsur ketidakpastian atau *probabilitas*.

2.3 Android

Android adalah sebuah sistem operasi yang dibangun di atas *kernel Linux* dan dirancang khusus untuk perangkat mobile dengan layar sentuh seperti *smartphone* dan tablet. Sistem ini awalnya dikembangkan oleh Android Inc, dengan tujuan menyediakan sistem operasi untuk kamera digital, namun kemudian beralih fokus ke ponsel pintar.

Google mengambil alih proyek ini dan mengembangkan Android sebagai *platform open source* yang kini digunakan secara luas di berbagai perangkat elektronik. Peluncuran perangkat Android pertama terjadi pada tahun 2008, dan terus berkembang menjadi sistem operasi paling dominan di pasar perangkat seluler.

2.3.1 Sejarah Android

Android bermula pada tahun 2003 ketika sebuah perusahaan bernama Android Inc. didirikan oleh Andy Rubin, Rich Miner, Nick Sears, dan Chris White di Palo

Alto, California. Awalnya, Android dikembangkan sebagai sistem operasi untuk kamera digital, namun kemudian fokusnya bergeser ke pengembangan sistem operasi untuk telepon pintar karena pasar kamera digital dianggap kurang besar. Pada tahun 2005, *Google* mengakuisisi Android Inc. dan mulai mengembangkan sistem operasi ini berbasis kernel Linux dengan tujuan menyediakan *platform* terbuka dan fleksibel untuk perangkat seluler.

Pada 5 November 2007, Google bersama beberapa perusahaan teknologi membentuk *Open Handset Alliance* untuk mengembangkan dan mempromosikan Android sebagai sistem operasi *open source*. Peluncuran ponsel Android pertama, *T-Mobile G1 (HTC Dream)*, terjadi pada 22 Oktober 2008. Sejak saat itu, Android terus mengalami perkembangan pesat dan menjadi sistem operasi paling populer di dunia untuk perangkat seluler, bahkan melampaui iOS pada tahun 2012.

Android juga dikenal dengan penamaan versi-versinya berdasarkan nama makanan manis secara alfabetis, seperti *Cupcake* (1.5), *Donut* (1.6), hingga *Pie* (9.0). Namun setelah versi 9, *Google* mengganti penamaan menjadi angka saja, misalnya Android 10 dan seterusnya.

Bahasa pemrograman memiliki dua bahasa yaitu, java dan kotlin. Java adalah bahasa pemrograman utama yang digunakan untuk pengembangan aplikasi Android. Sedangkan Kotlin adalah bahasa pemrograman modern yang juga didukung oleh Android dan semakin populer di kalangan pengembang.

2.3.2 Perkembangan Versi *Android*

Perkembangan versi *Android* dimulai dari peluncuran versi pertama, *Android 1.0*, pada tahun 2008 yang membawa fitur dasar seperti integrasi *Gmail*, *Google Maps*, dan *browser web*. Pada tahun berikutnya, *Android* mulai menggunakan penamaan versi berdasarkan nama makanan manis, dimulai dengan *Android 1.5 Cupcake* yang menambahkan fitur seperti *keyboard virtual* dan widget layar utama. Versi-versi selanjutnya seperti *Donut*, *Eclair*, dan *Froyo* terus memperbaiki performa dan menambah fitur seperti navigasi *turn-by-turn* dan dukungan multi-akun.

Pada tahun 2014, *Android 5.0 Lollipop* memperkenalkan desain *Material Design* yang memberikan tampilan lebih modern dan mendukung fitur *multi-window*. Versi 6.0 *Marshmallow* menambahkan fitur keamanan seperti sensor sidik jari dan pengelolaan baterai yang lebih efisien. *Android 7.0 Nougat* meluncurkan mode *split-screen* dan peningkatan notifikasi, sedangkan *Android 8.0 Oreo* membawa fitur *picture-in-picture* dan *autofill API*.

Setelah *Android 9 Pie*, *Google* mengganti penamaan versi menjadi angka saja, dimulai dari *Android 10* yang memperkenalkan mode gelap, hingga *Android 15* yang dirilis pada 2024 dengan fokus pada peningkatan personalisasi dan *multitasking*. Setiap versi *Android* terus menghadirkan inovasi untuk meningkatkan pengalaman pengguna dan performa perangkat.

2.3.3 *Android Virtual Device (AVD)*

Android Virtual Device (AVD) adalah sebuah alat yang memungkinkan pengembang untuk menjalankan dan menguji *Android* secara virtual di komputer

tanpa harus menggunakan perangkat fisik seperti *smartphone* atau tablet. AVD perangkat Android asli, sehingga pengembang dapat melihat bagaimana aplikasi berjalan di berbagai tipe dan versi perangkat. Dengan menggunakan AVD, proses pengujian aplikasi menjadi lebih praktis dan fleksibel karena bisa dilakukan langsung dari komputer.



Gambar 2.2 Android Virtual Device (AVD)

2.3.4 Java Development Kit (ADV)

Java Development Kit (JDK) adalah paket perangkat lunak yang berisi berbagai alat penting untuk membuat, mengompilasi, dan menjalankan program berbasis bahasa pemrograman Java. Di dalam JDK terdapat komponen utama seperti *Java Runtime Environment* (JRE) yang memungkinkan aplikasi Java berjalan, *compiler* untuk mengubah kode sumber menjadi kode yang bisa dijalankan oleh mesin virtual java, serta berbagai pustaka dan alat bantu lain seperti *debugger* dan dokumentasi otomatis. *Java Development Kit* merupakan alat utama yang digunakan oleh para pengembang untuk membangun aplikasi java di berbagai platform, karena menyediakan semua kebutuhan mulai dari penulisan kode hingga pengujian dan peluncuran aplikasi.

2.3.5 *Android Software Development Kit* (SDK)

Android Software Development Kit (SDK) adalah kumpulan alat dan pustaka yang digunakan oleh pengembang untuk membuat aplikasi Android. SDK ini menyediakan berbagai komponen penting seperti API, emulator untuk menguji aplikasi tanpa perangkat fisik, alat debugging untuk menemukan dan memperbaiki kesalahan, serta alat untuk membangun dan menyiapkan aplikasi agar siap dirilis. Dengan *Android* SDK, pengembang dapat menulis, menguji, dan mengoptimalkan aplikasi agar kompatibel dengan berbagai versi Android secara lebih mudah dan efisien.

2.4 *XAMPP*

Xampp adalah perangkat lunak *open-source* yang dapat diinstal di berbagai sistem operasi seperti *Windows*, *Linux*, dan *macOs*. *Xampp* dirancang untuk mempermudah pengembangan dalam menginstal dan mengkonfigurasi server web *Apache* beserta komponen-komponen yang lain tanpa adanya kesulitan. *Xampp* memiliki fungsi sebagai server web lokal, memungkinkan pengguna untuk menguji dan mengembangkan aplikasi web secara offline.



Gambar 2.3 XAMPP

2.5 *Android Studio*

Menurut Sari, S.P., dan Prabowo, H. (2020), Android adalah sistem operasi yang menyediakan platform terbuka bagi para pengembang untuk membuat, mengubah, mengembangkan, dan menyebarluaskan aplikasi sehingga memungkinkan pengembangan aplikasi sendiri. Penggunaan media berbasis Android dapat memberikan inovasi baru dalam belajar, menumbuhkan minat, meningkatkan motivasi, dan rangsangan fitur-fitur *Android Studio*. *Android studio* memiliki beberapa fitur, sebagai berikut:

1. Editor kode, adalah editor yang canggih dengan fitur seperti penyelesaian otomatis, penyorotan sintaks, dan *refactoring*.
2. Integrasi dengan Git, untuk mempermudah pengelolaan versi kode sumber.
3. Desain antarmuka, untuk menyediakan alat desain visual untuk membuat antarmuka pengguna (UI) dengan *drag-and-drop*.
4. Emulator, memungkinkan pengembang untuk menguji aplikasi di berbagai perangkat virtual tanpa memerlukan perangkat fisik.
5. Alat profiling untuk membantu dalam menganalisis kinerja aplikasi dan penggunaan sumber daya.

2.5.1 **Komponen-komponen Android Studio**

Selain fitur- fitur Android studio, Android studio memiliki beberapa komponen sebagai berikut:

1. *Plugin* dan *Ekstensi* adalah Android Studio yang mendukung berbagai *plugin* yang dapat memperluas fungsionalitas IDE. Pengembang dapat menambahkan alat tambahan sesuai kebutuhan proyek mereka.
2. Manajemen proyek adalah Android Studio yang menyediakan struktur proyek yang terorganisir, memudahkan pengembang dalam mengelola file dan sumber daya aplikasi.
3. Integrasi dengan Firebase adalah Android Studio memungkinkan integrasi yang mudah dengan Firebase, platform pengembangan aplikasi yang menyediakan berbagai layanan seperti analitik, autentikasi, dan penyimpanan data.
4. Dukungan Multibahasa, selain java dan kotlin, Android studio juga mendukung pengembangan aplikasi menggunakan bahasa pemrograman lain, seperti C++ melalui Android NDK (Native Development Kit).
5. Alat Pengujian yaitu Android studio yang dilengkapi dengan alat untuk pengujian unit dan pengujian fungsional, seperti *Espresso* dan *JUnit*, yang membantu memastikan kualitas aplikasi.



Gambar 2.4 Android Studio

2.5.2 Aspek Penting dalam Android Studio di Luar Komponen UI

Android studio tidak hanya tentang menyusun tampilan aplikasi (komponen *UI*). Aspek penting lainnya seperti:

1. Arsitektur aplikasi, menentukan bagaimana kode terstruktur dan data yang dikelola (misalnya menggunakan *MVVM* atau *MVP*).
2. Manajemen Proyek, meliputi pengaturan file, dan proses build.
3. *Debugging* dan Pengujian, krusial untuk menentukan dan memperbaiki *bug* serta memastikan kualitas aplikasi.
4. Integrasi dengan Layanan Eksternal, seperti *Firebase* atau Google Maps memperluas fungsional.
5. *Gradle* , yaitu sistem *build* Android yang perlu dipahami untuk mengelola dependensi dan konfigurasi.
6. Kontrol Versi, misalnya *Git* penting untuk kolaborasi tim dan manajemen kode.
7. Optimasi Performa dan Penerbitan Aplikasi, yang dilakukan ke Google Play Store untuk melengkapi siklus pengembangan.

Semua aspek ini sama pentingnya dengan komponen visual dalam membangun aplikasi Android Studio.

2.6 MySQL (My Structured Query Language)

MySQL adalah salah satu sistem manajemen basis data relasional (Relational Database Management System/RDBMS) yang bersifat open source. *MySQL*

digunakan untuk mengelola, menyimpan, dan memanipulasi data yang terhubung dengan aplikasi berbasis web. (Indriani et al., 2022)

MySQL tersedia dalam dua jenis lisensi, yaitu lisensi perangkat lunak bebas (*Free Software*) dan *lisensi shareware* yang memiliki batasan penggunaan. Hal ini memungkinkan MySQL dipakai secara gratis untuk keperluan pribadi maupun komersial tanpa harus membayar lisensi, sehingga sangat populer di kalangan pengembang aplikasi web dan sistem berbasis data

Secara umum, MySQL banyak digunakan untuk berbagai keperluan seperti data warehousing, *e-commerce*, aplikasi logging, dan pengembangan aplikasi web dinamis, terutama yang dikombinasikan dengan bahasa pemrograman seperti PHP untuk membuat aplikasi berbasis database yang interaktif dan responsive.

2.6.1 UML (Unified Modeling Language)

Berikut ini merupakan definisi UML (*Unified Modeling Language*) dari beberapa ahli, diantaranya sebagai berikut:

1. Menurut Ronal et al. (2022), UML adalah alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain sistem secara visual, yang berisi sintaks untuk memodelkan sistem software berbasis objek.
2. Nurulfikri (Jurnal Informatika Terpadu, 2024), UML adalah bahasa pemodelan grafis yang menjadi standar untuk memodelkan sistem dengan metodologi pemodelan berorientasi objek. UML distandarkan oleh *Object Management Group* (OMG) dan pertama kali dipopulerkan oleh *Grady Booch*, *James Rumbaugh*, dan *Ivar Jacobson*. UML memiliki empat fungsi

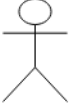
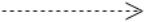
utama: *virtualizing* (alat komunikasi antar tim pengembang), *specifying* (memodelkan sistem secara tepat), *constructing* (memetakan model ke bahasa pemrograman), dan *documenting* (dokumentasi sistem).

UML sering digunakan dalam berbagai konteks pengembangan sistem, seperti pembuatan aplikasi data penduduk, sistem pelayanan administrasi, dan pengembangan sistem informasi laboratorium, dengan tujuan meningkatkan efisiensi dan efektivitas pengelolaan data serta meminimalisir kesalahan informasi. UML juga menjadi alat bantu dalam metode pengembangan sistem seperti *Rational Unified Process* (RUP) yang terdiri dari beberapa fase mulai dari identifikasi kebutuhan hingga pengujian sistem.

2.6.2 Use Case Diagram

Use Case Diagram merupakan berfungsi menyatakan interaksi pengguna dengan sistem secara visual dan menyajikan seluruh fungsi sistem yang tersedia

Menurut (Ejournal, n.d.)(Ejournal, n.d.)

Simbol	Nama	Keterangan
	Aktor	Aktor merupakan Entitas yang berinteraksi dengan sistem, seperti "Guru", "Siswa", dan "Admin".
	Use Case	Fungsi atau layanan yang ditawarkan oleh sistem, seperti "Menandai Kehadiran", "Lihat Laporan Kehadiran", "Kelola Data Siswa", dan "Menghasilkan Laporan".
	Association	Garis yang menghubungkan aktor dengan use case, menunjukkan interaksi antara keduanya.
	Include	<i>Include</i> adalah sebuah hubungan yang menunjukkan suatu proses (<i>use case</i>) yang selalu membutuhkan dan memanggil proses lain setiap kali dijalankan.
	Generalization	<i>Generalization</i> merupakan hubungan antara 2 <i>use case</i> atau 2 aktor di mana 1 elemen (anak) mewarisi sifat, perilaku, dan struktur dari elemen lain yang lebih umum (induk).

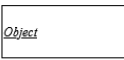
Gambar 2.5 Simbol Use Case Diagram

2.6.3 Sequence Diagram

Sequence Diagram merupakan diagram yang digunakan untuk menggambarkan interaksi antar objek dalam sebuah sistem secara berurutan dan terinci, menampilkan pesan atau perintah yang akan dikirim beserta dengan waktu pelaksanaannya. Diagram ini memiliki dua dimensi yaitu, vertikal yang menunjukkan waktu dan horizontal yang menunjukkan objek atau aktor yang terlibat. Setiap objek memiliki *lifeline* yang menggambarkan

waktu aktifnya, dan interaksi antar objek ditunjukkan dengan panah pesan dari satu *lifeline* ke *lifeline* lainnya.

Sequence Diagram biasanya digunakan untuk memvisualisasikan urutan langkah dalam suatu proses sebagai respons terhadap suatu kejadian, sehingga menggambarkan interaksi antar objek dalam suatu proses berdasarkan urutan waktu. (Tiga et al., 2021)

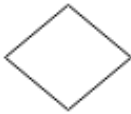
Simbol	Nama	Keterangan
	Aktor	Aktor merupakan Entitas yang berinteraksi dengan sistem, seperti "Guru", "Siswa", dan "Admin".
	<i>Lifeline</i>	<i>Lifeline</i> berfungsi menggambarkan keberadaan sebuah objek dalam basis waktu.
	<i>Message</i>	Menggambarkan sebuah komunikasi antar objek.
	<i>Object</i> (Partisipan)	Object merupakan <i>instance</i> dari sebuah <i>class</i> yang dituliskan secara tersusun dan horizontal.
	<i>Activation</i>	<i>Activation</i> berfungsi untuk menggambarkan tempat mulai dan berakhirnya pesan.

Gambar 2.6 Simbol *Sequence* Diagram

2.6.4 *Activity* Diagram

Activity Diagram adalah salah satu jenis diagram dalam UML yang biasanya digunakan untuk memodelkan alur kerja (*workflow*) atau aktivitas dalam suatu sistem. Diagram ini memperlihatkan urutan aktivitas, keputusan, serta kondisi yang

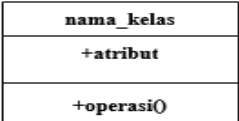
terjadi selama proses berlangsung, sehingga sangat efektif untuk mendokumentasikan proses bisnis atau prosedur dalam pengembangan perangkat lunak. (Puturu, 2022)

Simbol	Nama	Fungsi
	<i>Initial State</i>	Sebuah diagram yang memiliki status awal.
	<i>Activity</i>	Sebuah aktivitas yang dilakukan oleh sistem, aktivitas biasanya diawali dengan kata kerja.
	<i>Join/Percabangan</i>	Percabangan yang dimana ada pilihan aktivitas yang lebih dari 1.
	<i>Final State</i>	Status akhir yang dilakukan oleh sistem.

Gambar 2.7 Activity Diagram(Intern, 2021)

2.6.5 Class Diagram

Class Diagram merupakan salah satu jenis diagram struktur pada *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan struktur sistem secara statis, termasuk deskripsi kelas, atribut, metode, serta hubungan antar kelas dalam suatu sistem(Ramdany, 2024).

Simbol	Nama	Fungsi
	Tabel	Simbol ini adalah untuk sebuah kelas pada struktur sistem. Penulisan tidak diperbolehkan menggunakan spasi. Simbol memiliki 3 susunan, yaitu kotak pertama nama kelas, kotak kedua atribut dan kotak terakhir operasi.
	Generalization	Generalization digunakan untuk menghubungkan kelas dengan arti umum-khusus. Jadi jika ada kelas yang memiliki makna kelas umum dan makna kelas khusus dapat menggunakan simbol ini.

Gambar 2.8 Class Diagram

2.7 Penelitian Terdahulu

Peneliti & Tahun	Judul Penelitian	Metode Penelitian	Temuan Penting	Relevansi	Referensi
Puspa, A., Ramadhani , R., & Setiawan, D. (2023)	Pengembangan Aplikasi Absensi Berbasis Android dengan Android Studio	<i>Research and Development</i> (R\&D) dengan pendekatan kualitatif.	Aplikasi absensi ini berhasil dikembangkan menggunakan Android Studio, memberikan kemudahan bagi pihak sekolah dalam mengelola kehadiran siswa dan mempercepat laporan	Penelitian ini serupa dalam penggunaan Android Studio dan konteks pendidikan, sehingga dapat dijadikan referensi teknis dan konseptual untuk Pengembangan Aplikasi Absensi Berbasis	Puspa, A., Ramadhani, R., & Setiawan, D. (2023). Pengembangan Aplikasi Absensi Berbasis Android dengan Android Studio. Jurnal Teknik Informatika dan Sistem

			absensi.	Android di SD Mis Baiturrahman Labuhanbatu Selatan.	Informasi, 11(1), 31–40.
Sahrir, S., & Thamrin, A. N. (2024)	Pengembangan Aplikasi Absensi Mahasiswa Realtime Menggunakan SMS Gateway Berbasis Android	R\&D dengan pendekatan prototyping dan uji black-box serta validasi pengguna.	Aplikasi berhasil digunakan dengan baik, memberikan fitur notifikasi kehadiran secara <i>real-time</i> melalui SMS, dan mendapat respons positif dari pengguna.	Menjadi referensi dalam menambahkan fitur real-time dan notifikasi dalam sistem absensi Android di lingkungan pendidikan.	Sahrir, S., & Thamrin, A. N. (2024). Pengembangan Aplikasi Absensi Mahasiswa Realtime Menggunakan SMS Gateway Berbasis Android. Jurnal Manajemen Informatika (JMP), 9(1), 55–62.
Fawwazi, N., & Hidayat, M. (2023)	Rancang Bangun Aplikasi Kehadiran Asisten Praktikum Menggunakan	<i>Agile software development</i> dengan uji kelayakan pengguna.	Aplikasi berhasil dibangun dalam beberapa sprint, dan hasil evaluasi	Relevan dari sisi metode pengembangan perangkat lunak (<i>Agile</i>) dan implementasi	Fawwazi, N., & Hidayat, M. (2023). Rancang Bangun Aplikasi Kehadiran

	Metode <i>Agile</i> .		menunjukkan tingkat kepuasan pengguna di atas 90%.	Android dalam lingkungan pendidikan.	Asisten Praktikum Berbasis Android Menggunakan Metode <i>Agile</i> . Jurnal Misi, 9(2), 134–141.
Sufandi, S., Siswanto, S., & Hasan, H. (2022)	Pengembangan Aplikasi Presensi Berbasis Android dan Web di Politeknik Negeri Pontianak	Metode Waterfall, dengan implementasi GPS dan dokumentasi foto sebagai bukti kehadiran.	Aplikasi meningkatkan akurasi dan transparansi dalam pencatatan kehadiran karena verifikasi dilakukan melalui GPS dan foto.	Dapat dijadikan referensi untuk mengembangkan fitur validasi berbasis lokasi atau gambar dalam sistem absensi di sekolah dasar.	Sufandi, S., Siswanto, S., & Hasan, H. (2022). Pengembangan Aplikasi Presensi Berbasis Android dan Web di PNP. Jurnal Pendidikan Informatika dan Sains, 11(1), 22–29.
Ramdany, D. (2021)	Desain Class Diagram dalam Aplikasi <i>Mobile</i> Sekolah	Studi deskriptif dengan pendekatan rekayasa perangkat	Penggunaan class diagram yang tepat membantu dalam mempercepat	Bermanfaat dalam tahap perancangan sistem absensi, terutama pada bagian desain	Ramdany, D. (2021). Desain Class Diagram dalam Aplikasi <i>Mobile</i> Sekolah. Jurnal

		lunak berbasis UML.	proses pengembangan aplikasi dan meminimalisir kesalahan database.	class diagram.	Rekayasa Teknologi, 7(2), 76–82.
--	--	---------------------	--	----------------	----------------------------------

2.8 Sejarah Singkat SD Mis Baiturrahman

Pendidikan Dasar MIS BAITURRAHMAN, yang berdiri sejak 0000-00-00, merupakan lembaga pendidikan dasar tingkat MI yang berstatus Swasta, dan berada di bawah pembinaan Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi. Sekolah ini melayani pendidikan formal bagi anak-anak di wilayah Kec. Torgamba, dengan fokus utama pada penguatan literasi, numerasi, serta pembentukan karakter yang berbasis pada Profil Pelajar Pancasila.

Dalam menjalankan kegiatan belajar mengajar, Mis Baiturrahman Labuhanbatu Selatan menerapkan pendekatan pembelajaran aktif, kreatif, dan menyenangkan. Kurikulum yang digunakan memungkinkan siswa untuk belajar sesuai dengan tahap perkembangannya, serta menumbuhkan minat dan bakat secara optimal.

Beragam kegiatan ekstrakurikuler juga disediakan sebagai media pengembangan diri siswa, seperti pramuka, seni, olahraga, serta kegiatan sosial yang membangun empati dan kepedulian terhadap sesama.

Seiring waktu, Mis Baiturrahman Labuhanbatu Selatan telah melahirkan lulusan-lulusan yang tidak hanya berprestasi secara akademik, tetapi juga memiliki kepribadian yang baik dan siap menghadapi tantangan di jenjang pendidikan berikutnya. Dengan komitmen untuk terus meningkatkan mutu pendidikan, sekolah

ini menjadi pilihan terpercaya masyarakat Kec. Torgamba dalam menyekolahkan anak-anak mereka.