

BAB II

KAJIAN LITERATUR

2.1 Grand Tiga Sunggal

Grand Tiga Sunggal merupakan salah satu lembaga pendidikan non-formal yang berdiri sejak tahun 1994 dan beralamat di Jl.Sunggal No.374, Kec. Medan Sunggal, Kota Medan Sumatera Utara 20128. Program – program bimbingan belajar yang ditawarkan mencakup calistung, semua mata pelajaran dan bahasa Inggris. Program ini menargetkan siswa Sekolah Dasar (SD), Sekolah Menengah Pertama (SMP), dan Sekolah Menengah Atas (SMA) dengan pendekatan pembelajaran yang disesuaikan dengan karakteristik perkembangan kognitif setiap jenjang.

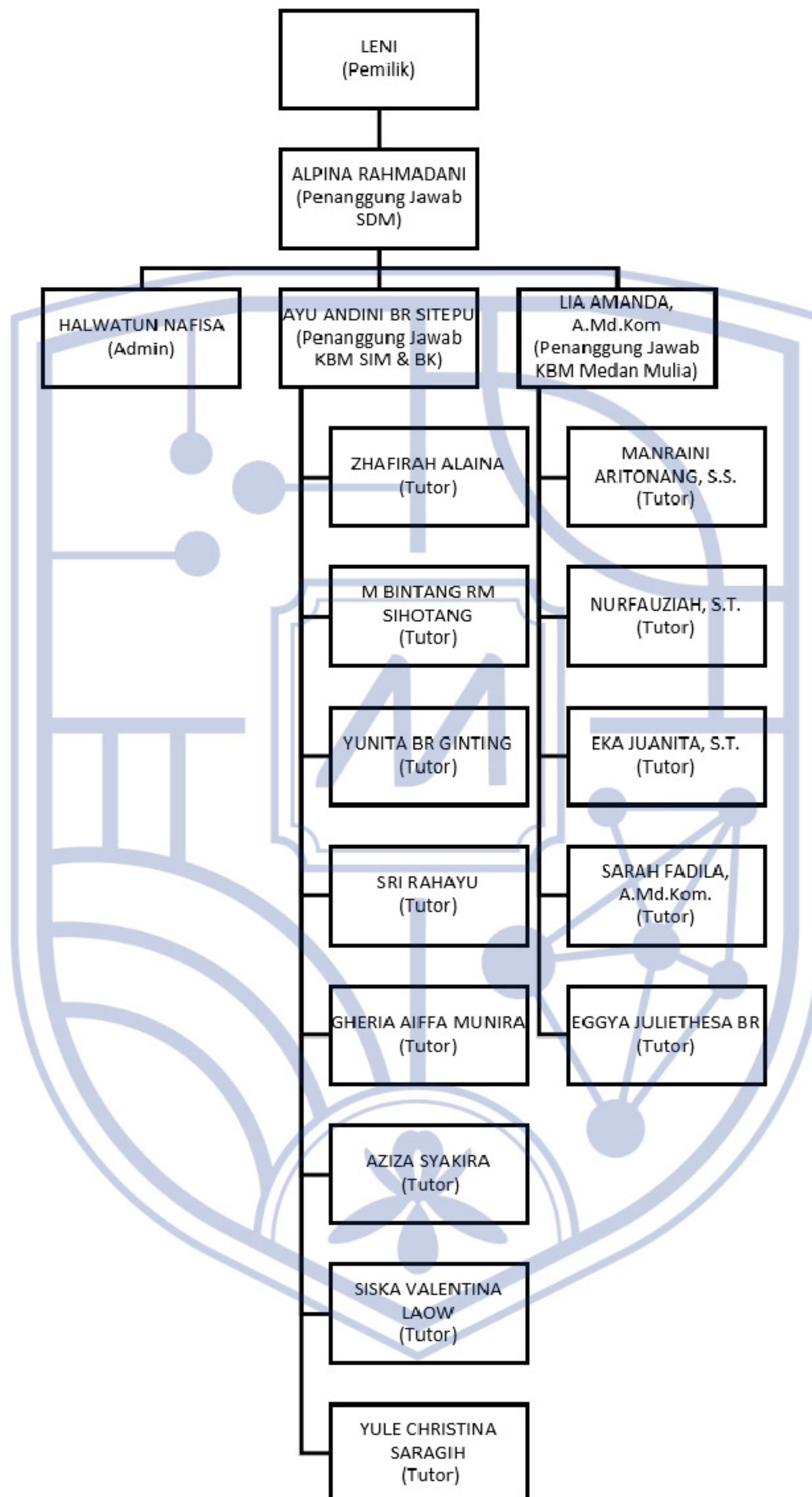
Dalam menjalankan kegiatan operasionalnya, Grand Tiga Sunggal memiliki visi dan misi sebagai landasan utama dalam penyelenggaraan layanan pendidikan. Grand Tiga Sunggal memiliki visi untuk menjadi lembaga bimbingan belajar yang dibutuhkan dan dipercaya oleh siswa melalui penyelenggaraan layanan pendidikan yang berkualitas, sehingga mampu mencetak siswa yang cerdas, mandiri, berwawasan ke depan, berakhlak mulia, serta memiliki rasa cinta tanah air.

Untuk mendukung pencapaian visi tersebut, misi bimbel ini adalah memberikan layanan yang berkualitas, penerapan metode pembelajaran yang efektif, pembinaan karakter yang sesuai nilai Pancasila.

2.1.1 Struktur Organisasi Grand Tiga Sunggal

Grand Tiga Sunggal memiliki struktur organisasi yang tersusun secara hierarkis untuk mendukung pengelolaan lembaga secara efektif. Struktur ini terdiri dari pemilik sebagai penanggung jawab utama, penanggung jawab SDM yang mengelola tenaga kerja, penanggung jawab kegiatan belajar mengajar (KBM) yang mengatur proses pembelajaran, admin yang menangani administrasi operasional, serta tutor sebagai tenaga pengajar.

Pembagian peran tersebut menunjukkan adanya koordinasi yang jelas dalam mendukung kelancaran operasional dan peningkatan kualitas layanan bimbingan belajar.

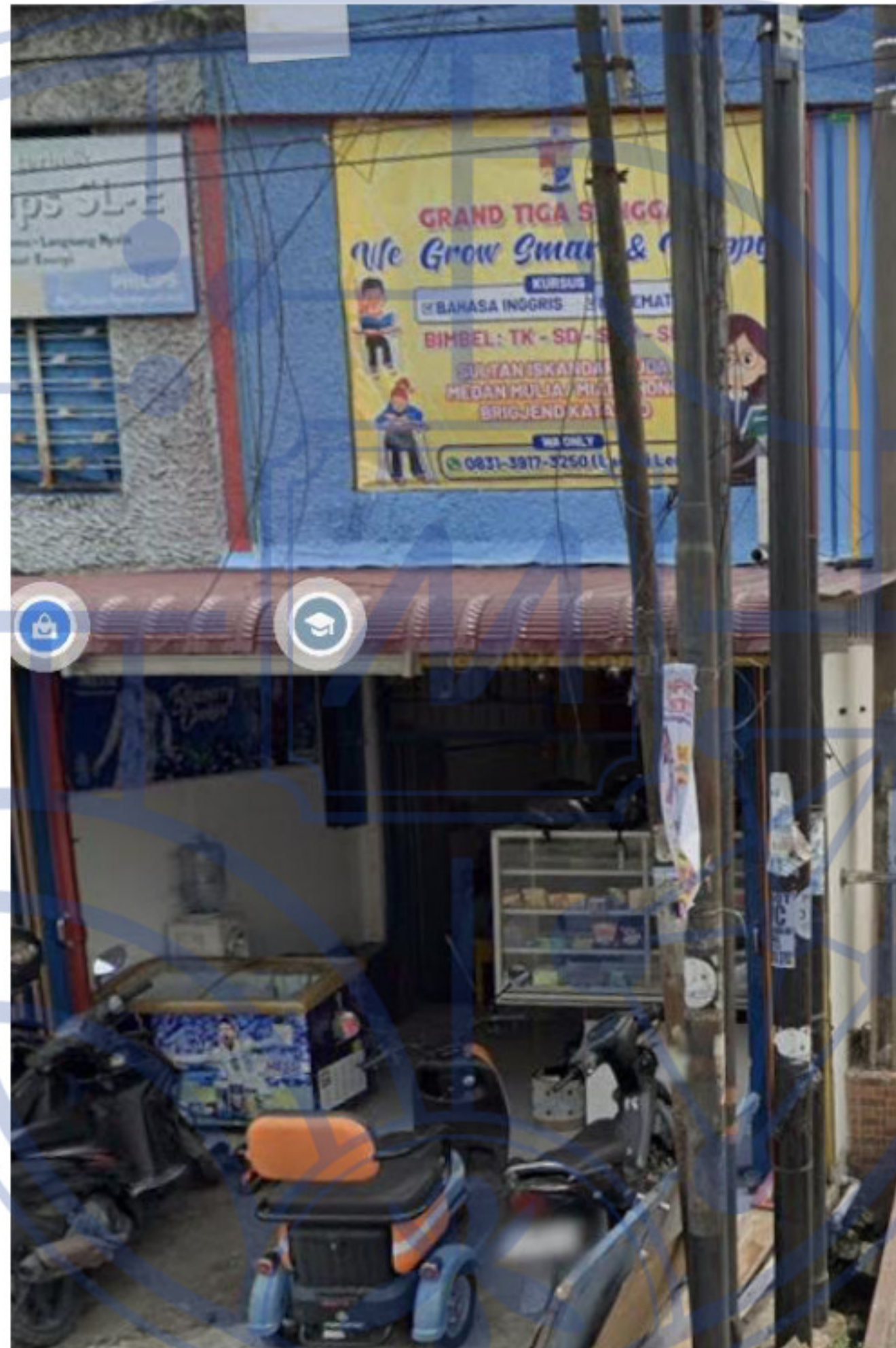


Gambar 2.1 Struktur Organisasi Grand Tiga Sunggal

2.1.2 Lokasi Grand Tiga Sunggal

Grand Tiga Sunggal berlokasi di Jl. Sunggal No.374, Kecamatan Medan Sunggal, Kota Medan, Sumatera Utara 20128. Lokasi ini berada pada area yang strategis dan mudah diakses oleh siswa dari berbagai wilayah di sekitarnya.

Untuk memberikan gambaran yang lebih jelas mengenai letak geografis bimbel, berikut ditampilkan peta lokasi Grand Tiga Sunggal yang diambil dari layanan Google Maps.



Gambar 2.2 Lokasi Grand Tiga Sunggal pada Google Maps [13]

2.2 Sistem Informasi

Sistem informasi merupakan gabungan dari *hardware*, *software*, *database*, *prosedur*, dan pengguna untuk mengumpulkan, mengolah, menyimpan, dan menyajikan data menjadi informasi yang bermanfaat. Tujuannya untuk meningkatkan efisiensi dan efektivitas pengelolaan data sehingga informasi yang dihasilkan akurat dan tepat waktu [14].

Sistem informasi terdiri dari enam blok yang diuraikan sebagai berikut [15]:

1. Blok Masukan (*Input Block*)

Merupakan bagian sistem informasi yang berfungsi untuk menerima data yang akan diolah

2. Blok Model (*Model Block*)

Merupakan bagian sistem yang berfungsi untuk mengolah data masukan, logika, dan model matematika sehingga menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (*Output Block*)

Keluaran merupakan informasi berkualitas yang dihasilkan untuk mendukung kebutuhan operasional dan pengambilan keputusan.

4. Blok Basis Data (*Database Block*)

Merupakan kumpulan data yang saling terintegrasi dan disimpan pada perangkat keras komputer, yang pengelolaannya dilakukan dengan bantuan perangkat lunak.

5. Blok Teknologi (*Technology*)

Komponen yang berfungsi sebagai *tool box* dalam sistem informasi yang mendukung proses input, pengolahan data, penyimpanan dan akses data, penyajian keluaran, serta pengendalian sistem

6. Blok Kendali (*Control Block*)

Berfungsi untuk mengendalikan dan mengawasi jalannya sistem informasi agar terhindar dari kesalahan maupun gangguan sistem.

2.3 Manajemen

Manajemen merupakan suatu proses sistematis yang dilakukan untuk mengarahkan, mengoordinasikan, serta memanfaatkan seluruh sumber daya organisasi agar tujuan yang telah ditetapkan dapat dicapai secara efektif dan efisien. Penerapan manajemen tidak hanya berkaitan dengan proses pengambilan keputusan, tetapi juga mencakup penyusunan rencana, pengorganisasian sumber daya, pelaksanaan kegiatan, serta pengendalian terhadap setiap aktivitas organisasi atau lembaga. Dengan pengelolaan yang baik, lembaga mampu meningkatkan kualitas pelayanan, mempermudah proses evaluasi, serta menghasilkan informasi yang mendukung pengambilan keputusan [16].

Penerapan konsep manajemen juga menjadi bagian penting dalam penyelenggaraan pendidikan, baik pada jalur formal maupun nonformal. Pada lembaga pendidikan, manajemen berfungsi sebagai pedoman dalam mengatur seluruh sumber daya pendidikan agar proses pembelajaran, administrasi, serta pelayanan kepada peserta didik dapat berlangsung secara

efektif. Oleh karena itu, keberhasilan suatu lembaga pendidikan sangat dipengaruhi oleh kemampuan pengelola dalam melaksanakan fungsi-fungsi manajemen secara terpadu [17].

Dalam konteks pendidikan nonformal, manajemen dilaksanakan melalui tahapan perencanaan, pengorganisasian, pelaksanaan, pengawasan, dan evaluasi terhadap seluruh program yang diselenggarakan. Setiap fungsi tersebut saling berkaitan untuk memastikan bahwa kegiatan pendidikan berjalan sesuai tujuan, memenuhi kebutuhan masyarakat, serta mampu meningkatkan kualitas layanan pendidikan secara berkelanjutan [18].

2.4 Bimbingan Belajar

Belajar merupakan upaya yang dilakukan untuk memperoleh pengetahuan dan pemahaman terhadap suatu materi tertentu [19]. Aktivitas belajar tidak hanya berlangsung di lingkungan sekolah atau pertutoran tinggi, tetapi dapat dilakukan di luar jam pembelajaran formal, salah satunya melalui kegiatan bimbingan belajar. Bimbingan belajar merupakan bentuk bantuan yang diberikan kepada siswa dalam mengatasi berbagai permasalahan belajar agar proses pembelajaran dapat berlangsung secara optimal sesuai dengan kemampuan, bakat, dan minat yang dimiliki [20].

Bimbingan belajar bertujuan untuk membantu siswa dalam mengoptimalkan potensi, minat, dan bakat yang dimiliki baik dalam bidang akademik maupun pengembangan kepribadian. Selain itu, bimbingan belajar diarahkan untuk membentuk kebiasaan belajar yang efektif, menentukan strategi belajar yang sesuai, meningkatkan motivasi belajar, serta mendukung siswa dalam mencapai tujuan pendidikan dan cita-citanya [21].

Bimbingan belajar memberikan manfaat bagi siswa, baik dalam aspek akademik maupun psikologis. Melalui kegiatan bimbingan belajar, siswa memperoleh bantuan dalam memahami materi pelajaran sehingga dapat meningkatkan rasa percaya diri. Dari sisi psikologis bimbingan belajar membantu mengatasi kejenuhan belajar yang kerap muncul selama proses pembelajaran. Dengan adanya pendampingan dan arahan dari tutor, siswa dapat menghadapi tantangan mental dan emosional secara lebih baik, sehingga proses belajar dapat berlangsung lebih efektif dan berkelanjutan [22].

Selain aspek akademik, bimbingan belajar juga memiliki aktivitas administrasi yang kompleks, meliputi:

1. Pendaftaran siswa: pencatatan data pribadi, jenjang pendidikan, dan program yang diikuti.

2. Manajemen pembayaran: rekap biaya bimbingan, status pembayaran, serta penyusunan laporan keuangan.
3. Absensi siswa dan tutor: pencatatan kehadiran untuk memantau keterlibatan siswa dan menghitung honor tutor.
4. Penggajian tutor: perhitungan gaji berdasarkan jumlah kelas, kehadiran, dan kebijakan lembaga.
5. Pengelolaan jadwal kelas: penentuan waktu belajar, distribusi tutor, serta koordinasi ruang kelas.

Untuk mendukung kelancaran aktivitas tersebut, lembaga bimbingan belajar membutuhkan sistem informasi terintegrasi yang mampu:

1. Menyediakan database terpusat untuk menyimpan data siswa, tutor, jadwal, dan pembayaran.
2. Menghadirkan modul administrasi untuk pendaftaran, absensi, dan penggajian.
3. Menyediakan modul keuangan untuk pencatatan pembayaran dan laporan otomatis.
4. Menyediakan *dashboard* informasi yang dapat diakses oleh siswa, orang tua, tutor, dan admin.
5. Menambahkan fitur bantuan (*chatbot*) untuk menjawab pertanyaan umum terkait jadwal, program, dan informasi lembaga.

2.5 Website

Website adalah kumpulan halaman yang saling terhubung dan berisi informasi dalam bentuk teks, gambar, audio, video, maupun animasi, yang dapat diakses melalui jaringan internet dengan menggunakan protocol *Hypertext Transfer Protocol* (HTTP) serta perangkat lunak peramban (*browser*) [23].

Dalam sebuah *Website* ada beberapa unsur yang saling berkaitan dan mendukung proses penyajian informasi kepada pengguna, yaitu [24]:

1. Domain

Domain adalah alamat unik yang digunakan untuk mengidentifikasi dan mengakses sebuah *Website*. Domain berfungsi sebagai pengganti IP (*Internet Protocol*) agar lebih mudah diingat oleh pengguna

2. *Hosting*

Hosting adalah layanan infrastruktur berbasis server yang digunakan untuk menyimpan dan mengelola data *Website* melalui jaringan internet. Layanan ini berfungsi untuk menyediakan sumber daya komputasi yang memungkinkan *Website* menampilkan konten dan merespons permintaan pengguna ketika alamat *Website* diakses, sehingga *Website* dapat beroperasi secara optimal.

3. Bahasa Program

Bahasa program merupakan sarana yang digunakan oleh pengembang untuk memberikan instruksi kepada komputer dalam membangun dan menjalankan suatu sistem. Dalam pengembangan *Website*, beberapa bahasa yang umum digunakan antara lain HTML, CSS, dan JavaScript, serta bahasa lain yang dipilih sesuai dengan kebutuhan dan fungsi sistem yang dikembangkan.

4. Konten

Konten merupakan informasi yang disajikan dalam sebuah *Website* dengan tujuan memberikan informasi, mengarahkan, serta memengaruhi interaksi pengunjung. Konten dapat berupa teks, gambar, video, maupun audio, dan kualitasnya berperan penting dalam menarik perhatian pengguna serta meningkatkan kunjungan ulang. Selain itu, konten yang berkualitas dan terstruktur dengan baik juga menjadi salah satu faktor utama dalam optimasi mesin pencari (*Search Engine Optimization/SEO*), sehingga dapat meningkatkan visibilitas *Website* pada hasil pencarian.

5. Tampilan

Tampilan merupakan aspek visual *Website* yang mencakup tata letak, warna, tipografi, dan elemen antarmuka. Tampilan yang dirancang dengan baik dapat meningkatkan kenyamanan pengguna, memudahkan navigasi, serta mendukung efektivitas penyampaian informasi kepada pengunjung *Website*.

Website untuk bimbingan belajar tidak hanya berfungsi sebagai media informasi, tetapi juga sebagai platform transaksional dan manajerial. Melalui integrasi domain, hosting, bahasa program, konten, dan tampilan yang tepat, *Website* dapat [25]:

1. Menyediakan modul administrasi untuk pendaftaran siswa dan pengelolaan data.
2. Menyajikan modul akademik berupa jadwal kelas, materi pembelajaran, dan evaluasi hasil belajar.

3. Mendukung modul keuangan untuk pencatatan pembayaran, pengeluaran operasional, dan laporan keuangan otomatis.
4. Memfasilitasi interaksi antara siswa, tutor, dan orang tua melalui fitur komunikasi daring dan *dashboard* khusus yang disesuaikan untuk masing-masing pengguna

Dengan demikian, *Website* menjadi sarana strategis yang tidak hanya menyampaikan informasi, tetapi juga mengintegrasikan seluruh aktivitas operasional lembaga bimbingan belajar dalam satu platform terpadu [25].

2.6 RAD

Rapid Application Development (RAD) adalah metode pengembangan perangkat lunak yang berfokus pada siklus pembuatan yang cepat dan iteratif. Kunci utamanya adalah pembuatan prototipe berdasarkan masukan langsung dari pengguna (klien) untuk mencapai hasil yang memuaskan dalam waktu singkat [26].

Pemilihan Rapid Application Development sebagai landasan metodologi didasarkan pada kemampuannya menghasilkan pengembangan prototipe dengan cepat, melibatkan partisipasi pengguna yang aktif, serta menerapkan iterasi pendek yang sesuai dengan validasi kebutuhan operasional secara berkelanjutan. Dengan menggunakan RAD, modul inti sistem dapat diuji langsung oleh pengguna sejak tahap awal, sehingga mengurangi risiko ketidaksesuaian fitur dan mempercepat perwujudan manfaat fungsional.

Berikut adalah empat tahap inti dalam metode RAD [27]:

1. *Requirements Planning*

Requirements Planning (Perencanaan Kebutuhan) adalah tahap fondasi dalam RAD yang bertujuan untuk membangun pemahaman dan kesepakatan yang komprehensif antara pengembang dan klien mengenai masalah, tujuan, serta ruang lingkup sistem. Tahap ini dilakukan melalui serangkaian *workshop* atau wawancara mendalam untuk menggali semua kebutuhan operasional, sekaligus memastikan bahwa tujuan sistem selaras dengan kebutuhan bisnis yang sebenarnya.

Sebagai bagian dari tahap *Requirements Planning*, dilakukan analisis sistem berjalan (*current system analysis*) untuk mempelajari proses bisnis yang sedang berlangsung, mengidentifikasi permasalahan yang ada, serta menentukan kebutuhan sistem yang akan dikembangkan. Analisis ini dilakukan melalui observasi dan wawancara sehingga pengembang

memperoleh pemahaman mengenai kondisi sistem yang sedang digunakan sebagai dasar dalam merancang solusi yang sesuai dengan kebutuhan pengguna [28].

Keuntungan menggunakan Perencanaan Kebutuhan adalah pembentukan fondasi proyek yang kokoh dan terukur sejak awal, yang secara signifikan mengurangi risiko *scope creep* (perambahan lingkup) dan kesalahpahaman di fase selanjutnya. Dengan mendokumentasikan semua kebutuhan secara eksplisit dan mendapatkan persetujuan bersama, baik tim pengembang maupun klien memiliki acuan yang objektif untuk mengevaluasi kemajuan dan hasil akhir. Proses kolaboratif ini juga mempercepat iterasi selanjutnya karena tim tidak perlu terus-menerus kembali menanyakan tujuan dasar atau memperdebatkan fitur yang sudah disepakati.

Keberhasilan tahap perencanaan secara langsung menentukan efisiensi iterasi selanjutnya. Sebuah perencanaan yang komprehensif dan realistis memungkinkan tim untuk memprioritaskan fitur-fitur yang paling bernilai bisnis tinggi untuk dibangun dalam prototipe awal. Dengan demikian, risiko *mismatch* fitur dapat diminimalisir sejak dini, dan sumber daya proyek dapat dialokasikan secara lebih efektif untuk memastikan manfaat operasional dapat segera dirasakan oleh mitra.

2. User Design

User Design (Desain Sistem) adalah fase iteratif untuk menerjemahkan kebutuhan yang telah direncanakan menjadi cetak biru (*blueprint*) visual dan teknis yang siap dikembangkan. Tahap ini dimulai dengan pembuatan desain awal seperti *wireframe* antarmuka pengguna (UI), diagram alur, dan model data, yang kemudian secara berkala disajikan dan divalidasi langsung oleh klien. Desain setiap modul akan diperiksa dan disempurnakan berdasarkan umpan balik langsung dari pengguna akhir, memastikan kesesuaian dengan kebiasaan kerja mereka sebelum masuk ke pemrograman.

Keuntungan menggunakan perancangan sistem yang iteratif adalah penghematan biaya dan waktu yang besar, karena kesalahan atau ketidaksesuaian desain dapat ditemukan dan diperbaiki pada fase konsep yang relatif murah, bukan selama pengkodean yang lebih mahal. Dengan melibatkan klien dalam review desain, metode RAD memastikan bahwa arah pengembangan selalu sesuai dengan ekspektasi. Pendekatan ini menghasilkan sistem yang intuitif dan sesuai dengan alur kerja nyata pengguna, sehingga meningkatkan adopsi dan kepuasan terhadap produk yang dikembangkan.

Dengan tujuan untuk mengunci desain yang telah divalidasi pada akhir tahap perancangan, dihasilkan suatu desain sistem yang telah disetujui bersama (*approved design*). Desain ini bukan lagi sekadar konsep, melainkan sebuah panduan teknis yang detail dan stabil, mencakup *Entity Relationship Diagram* (ERD) untuk struktur *database*, diagram alur proses bisnis, serta *mock-up* antarmuka yang lengkap. Kestabilan desain ini sangat penting untuk memungkinkan tahap pengembangan berikutnya berjalan dengan lancar dan terarah, meminimalkan kebingungan dan perubahan mendadak selama proses pemrograman.

3. *Construction*

Construction (Pengembangan) adalah inti dari proses RAD yang mengubah desain final menjadi produk perangkat lunak fungsional melalui siklus pembangunan prototipe yang cepat dan berulang. Pada tahap inilah kode program untuk modul-modul sistem mulai ditulis dan dirakit menjadi sebuah prototipe kerja (*working prototype*) yang dapat diuji. Ciri khas RAD terletak pada siklus singkatnya: segera setelah sebuah komponen atau modul selesai dikembangkan dalam versi beta, ia langsung diserahkan kepada klien untuk diuji dan dikonfirmasi, sehingga umpan balik dapat segera diintegrasikan ke dalam iterasi berikutnya.

Keuntungannya adalah efisiensi dan reduksi risiko. Tim pengembang dapat mendeteksi dan memperbaiki bug atau penyimpangan kebutuhan dengan sangat cepat, sehingga meningkatkan kualitas produk dan mengurangi risiko teknis. Sementara bagi klien, siklus ini memungkinkan mereka untuk melihat progres nyata, memberikan konfirmasi langsung, serta merasakan manfaat operasional dari modul tertentu lebih awal. Interaksi ini membangun kepercayaan, rasa kepemilikan, dan memastikan bahwa perkembangan sistem selalu selaras dengan harapan.

Tahap Pengembangan dalam RAD tidak sekadar membangun kode, tetapi merupakan mekanisme penjaminan kualitas dan keselarasan yang berkelanjutan. Pendekatan ini secara praktis meminimalkan ketidakpastian proyek. Setiap iterasi yang singkat membawa klien selangkah lebih dekat ke sistem akhir yang benar-benar berfungsi dan bermanfaat, mewujudkan prinsip dasar RAD untuk menyajikan manfaat operasional dalam waktu yang lebih singkat.

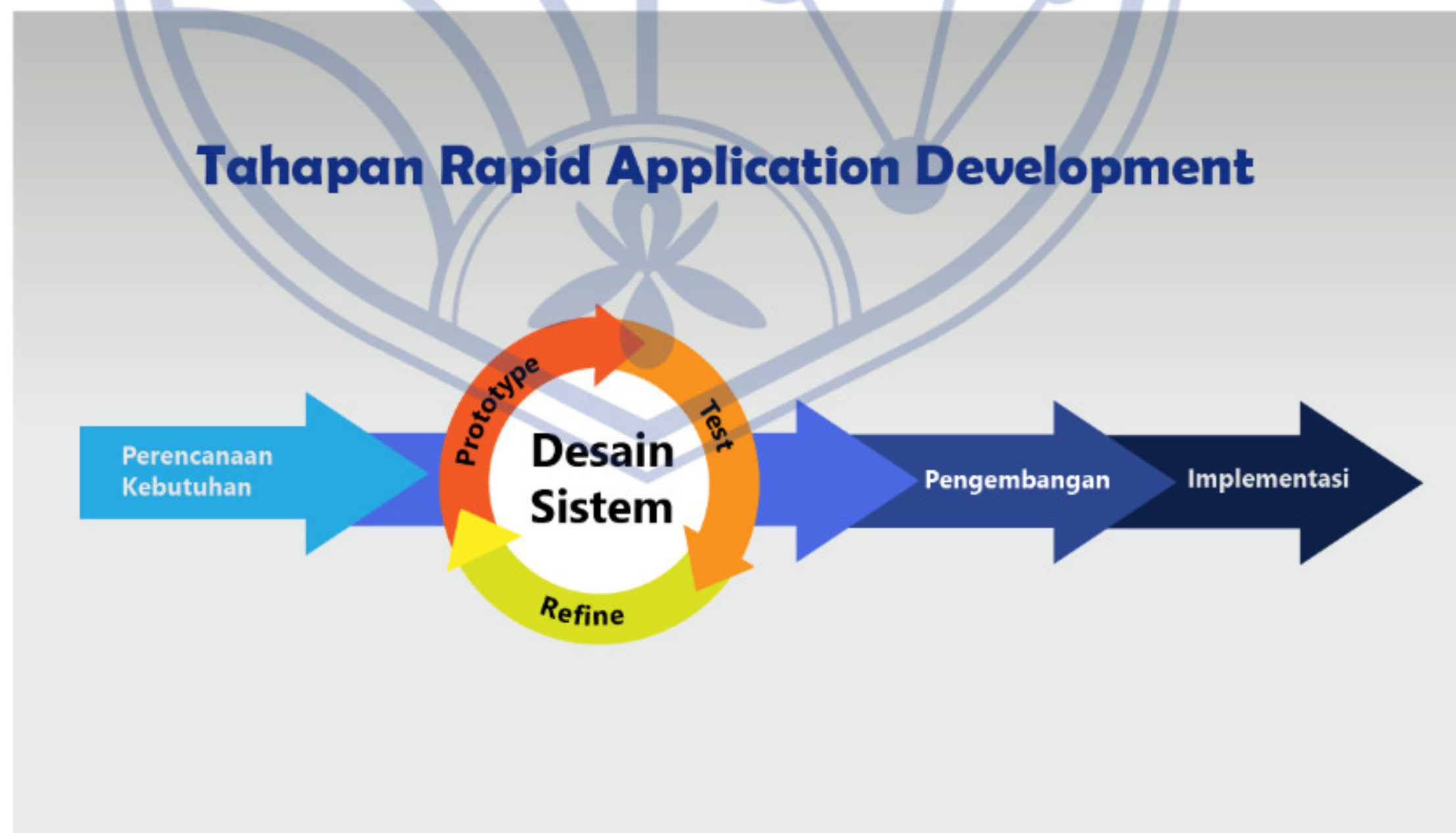
4. *Cutover*

Cutover (Implementasi) adalah tahap penerapan dan transisi sistem yang telah selesai dikembangkan ke dalam lingkungan operasional nyata pengguna. Tahap ini tidak hanya mencakup aktivitas teknis seperti peluncuran (*deployment*) perangkat lunak ke server produksi,

migrasi data, dan konfigurasi akhir, tetapi juga aspek kelembagaan yang sangat kritis, yaitu memastikan kesiapan organisasi dan pengguna untuk mengadopsi sistem baru. Pada konteks ini, pelatihan pengguna akhir dan penyiapan dokumentasi menjadi aktivitas paralel yang sama pentingnya dengan instalasi teknis.

Keuntungan menggunakan pendekatan Implementasi yang terstruktur dalam RAD adalah transisi yang lebih mulus dan sukses dari sistem lama ke baru, dengan gangguan operasional yang minimal. Dukungan dan pelatihan yang memadai memastikan pengguna merasa percaya diri dan mampu memanfaatkan sistem secara maksimal sejak hari pertama. Secara strategis, klien tidak hanya menerima sebuah perangkat lunak, tetapi sebuah solusi yang siap pakai beserta kapabilitas dalam organisasi untuk mengoperasikannya, yang memaksimalkan *Return on Investment* (ROI) dan percepatan perwujudan manfaat bisnis.

Untuk memastikan manfaat tersebut berkelanjutan, implementasi dalam RAD memiliki ciri khas berupa aktivitas pemantauan (*monitoring*) pasca-peluncuran yang aktif. Dengan demikian, tim pengembang tidak serta merta bubar setelah sistem "*go-live*", tetapi tetap siaga untuk memantau kinerja sistem, menangani masalah teknis, dan mengumpulkan umpan balik tambahan dari pengguna yang telah menggunakan sistem dalam skala dan intensitas penuh. Umpan balik ini, meskipun sistem telah diimplementasikan, dapat menjadi masukan berharga untuk rencana penyempurnaan atau pengembangan fase berikutnya, menegaskan bahwa siklus peningkatan berkelanjutan adalah bagian dari filosofi RAD.



Gambar 2.3 Tahapan RAD [29]

2.7 Teknik Pengembangan Sistem

Teknik pengembangan sistem digunakan sebagai pendekatan dalam merancang dan mengembangkan sistem secara sistematis agar sesuai dengan kebutuhan pengguna. Pada penelitian ini, digunakan pemodelan *Unified Modeling Language (UML)* sebagai alat bantu dalam menggambarkan proses dan interaksi sistem.

2.7.1 Use Case Diagram

Use Case Diagram adalah diagram UML yang menggambarkan fungsi, ruang lingkup, dan interaksi antara aktor dengan sistem secara visual [30]. *Use Case Diagram* memvisualisasikan layanan (*Use Case*) yang disediakan sistem serta siapa saja yang berinteraksi dengan layanan tersebut. *Use Case Diagram* memudahkan komunikasi kebutuhan antara pengembang dan pemangku kepentingan serta menjadi dasar pembuatan skenario pengujian. Berikut masing-masing komponen:

1. Aktor

Aktor adalah entitas atau pengguna yang berinteraksi dengan sistem, dapat berupa manusia, sistem lain, atau perangkat keras [30]. Aktor merepresentasikan peran yang melakukan atau memicu *Use Case* dalam sistem.

2. Sistem

Sistem adalah batasan konteks aplikasi atau layanan yang dikembangkan yang menerima input dari aktor dan memberikan output sebagai respons. Sistem menentukan ruang lingkup fungsi yang dimodelkan dalam *Use Case Diagram*.

3. Use Case

Use Case adalah tindakan atau layanan yang disediakan sistem dan biasanya digambarkan dalam bentuk oval untuk menunjukkan fungsi yang dapat dijalankan. *Use Case* menjelaskan urutan aksi yang menghasilkan keluaran yang dapat diamati oleh aktor.

4. Relationship

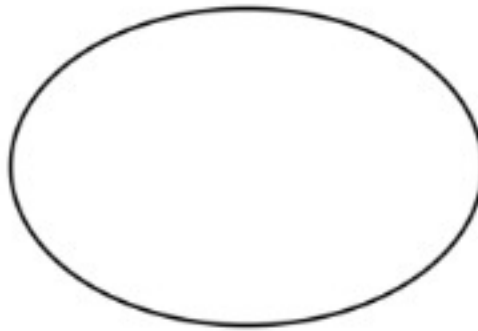
Relationship adalah hubungan antara aktor dan *Use Case* atau antar *Use Case*, meliputi asosiasi, *include*, *extend*, dan generalisasi untuk menunjukkan keterkaitan fungsional [30]. Relasi ini membantu memetakan dependensi dan variasi perilaku dalam diagram.

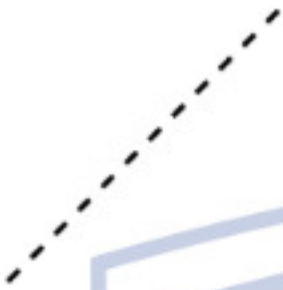
2.7.2 Activity Diagram

Activity Diagram adalah representasi grafis dari alur kerja atau proses yang menunjukkan urutan langkah, keputusan, dan paralelisme dalam suatu proses bisnis atau sistem.

Activity Diagram berguna untuk memetakan proses operasional seperti pendaftaran, pembayaran, dan absensi sehingga alur kerja menjadi jelas bagi pengembang dan pengguna [30]. Simbol yang digunakan dalam *Activity Diagram* dapat dilihat pada Tabel 2.1.

Tabel 2.1 Simbol *Activity Diagram*

Simbol	Keterangan
 Awal	Menandai titik awal alur aktivitas.
 Aktivitas	Menunjukkan aktivitas atau tindakan yang dilakukan.
 <i>Decision</i>	Menandai percabangan alur berdasarkan kondisi/keputusan.
 Akhir	Menandai titik akhir alur aktivitas.
 Aktor	Menunjukkan peran atau pemilik aktivitas dalam swimlane.
 <i>Use Case / Sub- Activity</i>	Menandakan sub-proses atau <i>Use Case</i> terkait dalam alur.

 <i>Flow line</i>	Menunjukkan arah aliran kontrol atau urutan aktivitas.
 <i>Dependency/Note/Asynchronous</i>	Menunjukkan hubungan tidak langsung, catatan, atau alur asinkron.
 <i>action/flow label</i>	Menunjukkan aliran kontrol dengan keterangan kondisi atau catatan singkat pada jalur tersebut

2.7.3 MySQL

MySQL adalah sistem manajemen basis data relasional (*Relational Database Management System* atau *RDBMS*) berbasis *open-source* yang menggunakan *Structured Query Language* (SQL). MySQL berfungsi sebagai media penyimpanan data relasional yang akurat, stabil, dan terstruktur. Sistem ini bekerja dengan memetakan informasi ke dalam tabel-tabel terorganisir yang saling terhubung melalui kunci relasi (*foreign key*), sehingga sangat efektif untuk mengintegrasikan berbagai data operasional yang kompleks

Dalam implementasinya, sistem database ini menawarkan performa yang stabil serta fleksibilitas tinggi dalam memanipulasi data melalui operasi CRUD (*Create, Read, Update, Delete*). Penerapan transaksi relasional pada MySQL juga memastikan konsistensi dan keamanan data pada setiap operasi kritis yang terjadi di dalam sistem. Selain itu, pengaturan indeks yang terstruktur pada kolom database secara signifikan dapat mengoptimalkan kinerja pencarian (*query*), sehingga proses penarikan atau rekapitulasi data dari lapisan basis data menuju antarmuka aplikasi dapat berjalan dengan cepat dan efisien

Dalam penelitian ini, MySQL digunakan untuk mengelola seluruh data operasional *Website* bimbingan belajar, seperti data siswa, jadwal kelas, data tutor, serta transaksi pembayaran. Dengan struktur relasional dan optimasi *query* yang dimiliki MySQL, sistem dapat menyajikan informasi akademik secara cepat dan *real-time* kepada pengguna [31].

2.8 Rule-Based Chatbot

Rule-Based Chatbot merupakan jenis *chatbot* yang bekerja berdasarkan aturan dan pola yang telah ditentukan sebelumnya oleh pengembang. Pendekatan ini didasarkan pada konsep *rule-based system*, yaitu metode pemecahan masalah yang menggunakan aturan berbentuk logika *if-then*, di mana sistem akan mengevaluasi kondisi tertentu (*if*) dan menghasilkan respons (*then*) sesuai dengan aturan yang telah ditentukan. Proses ini dapat berlangsung secara berulang hingga ditemukan aturan yang paling sesuai atau hingga tidak ada aturan yang dapat diterapkan [32].

Sistem ini merespons pertanyaan pengguna dengan cara mencocokkan input dengan kata kunci atau struktur tertentu yang sudah disiapkan, sehingga sangat efektif untuk menangani pertanyaan yang bersifat rutin. Dalam penerapannya, *Rule-Based Chatbot* sering memanfaatkan teknik seperti pencocokan pola, *regular expression*, serta struktur *decision tree* untuk menentukan jawaban yang paling sesuai. Dengan mekanisme tersebut, *chatbot* dapat memberikan respons secara cepat dan konsisten, terutama untuk pertanyaan umum seperti informasi jadwal, layanan, atau pertanyaan yang sering diajukan. Selain itu, pendekatan ini tidak memerlukan proses pelatihan data yang kompleks sehingga lebih ringan untuk diimplementasikan [33].

Rule-Based Chatbot terdiri dari beberapa komponen utama, yaitu basis aturan (*rule base*), mesin inferensi (*inference engine*), dan basis pengetahuan (*knowledge base*). Basis aturan berisi kumpulan kondisi dan respons yang telah ditentukan, sedangkan mesin inferensi bertugas mencocokkan input pengguna dengan aturan yang tersedia untuk menghasilkan output yang sesuai.

Di sisi lain, *Rule-Based Chatbot* memiliki keterbatasan dalam memahami variasi bahasa dan konteks percakapan. Meskipun menggunakan input berupa bahasa alami, sistem ini tidak benar-benar memahami makna bahasa secara semantik, melainkan hanya melakukan pencocokan terhadap kata kunci atau pola yang telah ditentukan. Namun, tingkat akurasi untuk pertanyaan yang sudah terdefinisi tergolong tinggi, sehingga tetap banyak digunakan pada sistem yang membutuhkan kecepatan dan ketepatan dalam menjawab pertanyaan sederhana. Dalam konteks *Website* bimbingan belajar, penggunaan *Rule-Based Chatbot* dinilai cukup sesuai karena sebagian besar pertanyaan pengguna berkaitan dengan informasi yang terstruktur,

seperti jadwal les atau data pengajar, sehingga dapat dilayani dengan baik tanpa memerlukan teknologi yang lebih kompleks [34].

2.8.1 Mekanisme Kerja *Rule-Based Chatbot*

Rule-Based Chatbot bekerja melalui serangkaian tahapan pemrosesan input pengguna hingga menghasilkan respons yang sesuai. Proses ini dirancang secara sistematis untuk memastikan bahwa setiap pertanyaan yang diberikan oleh pengguna dapat dicocokkan dengan aturan yang tersedia dalam sistem [12].

Berikut mekanisme *Rule-Based chatbot*:

1. *Preprocessing Input*

Teks masukan dari pengguna akan disederhanakan terlebih dahulu melalui proses normalisasi, seperti mengubah huruf menjadi *lowercase* serta menghapus tanda baca. Tahap ini bertujuan untuk meningkatkan konsistensi data sebelum dilakukan proses pencocokan [33].

2. *Pattern Matching*

Mencocokkan input pengguna dengan pola atau kata kunci tertentu yang telah disimpan dalam basis aturan. Metode yang umum digunakan dalam pendekatan ini adalah *keyword matching*, di mana sistem akan mendeteksi keberadaan kata kunci tertentu dalam teks masukan [35].

3. *Rule Selection*

Jika ditemukan lebih dari satu kata kunci yang cocok, sistem akan menentukan aturan yang paling relevan berdasarkan prioritas atau kecocokan terbanyak.

Jika tidak ditemukan aturan yang sesuai, maka sistem akan menjalankan mekanisme *fallback* [36].

4. *Response Generation*

Setelah aturan dipilih, sistem akan menampilkan respons sesuai dengan template yang telah disediakan. Respons dapat berupa teks statis atau dikombinasikan dengan data dari sistem [37].

2.8.2 Representasi Algoritma *Rule-Based Chatbot*

Secara konseptual, *Rule-Based Chatbot* direpresentasikan menggunakan sekumpulan aturan (*production rules*) yang berbentuk logika IF–THEN. Setiap aturan akan mengevaluasi

kondisi tertentu berdasarkan input pengguna dan menghasilkan respons apabila kondisi tersebut terpenuhi. Representasi algoritma sederhana dari mekanisme tersebut adalah sebagai berikut:

1. Menerima masukan dari pengguna.
2. Membandingkan masukan dengan setiap aturan pada basis aturan.
3. Jika ditemukan aturan yang sesuai, sistem menampilkan respons yang telah ditentukan.
4. Jika tidak ditemukan aturan yang sesuai, sistem menampilkan respons bawaan (fallback response).
5. Proses selesai.

2.9 *Black Box Testing*

Black Box Testing adalah metode pengujian perangkat lunak yang digunakan untuk menguji fungsionalitas sistem tanpa memperhatikan struktur internal maupun kode program [38]. Metode ini didasarkan pada prinsip mengevaluasi output sistem berdasarkan beragam jenis input, baik valid maupun tidak valid, untuk memverifikasi bahwa seluruh fungsi berjalan sesuai spesifikasi yang telah ditetapkan [39]. Dalam praktiknya, penguji berperan sebagai pengguna akhir dengan interaksi terbatas pada antarmuka aplikasi, sehingga fokus pengujian diarahkan pada pengalaman pengguna dan perilaku sistem secara eksternal [23]. Tujuan utama dari pengujian ini adalah mendeteksi kesalahan fungsional, ketidaksesuaian antarmuka, kesalahan akses data, inkonsistensi logika bisnis, serta masalah respons sistem terhadap berbagai skenario input [39]. *Black Box Testing* tidak memerlukan pengetahuan teknis mengenai arsitektur sistem atau bahasa pemrograman, sehingga dapat dilakukan oleh penguji non-teknis atau pengguna bisnis yang memahami kebutuhan sistem. Salah satu teknik yang digunakan dalam *Black Box Testing* adalah *Equivalence Partitioning*, yang membagi domain input menjadi kelas data valid dan tidak valid untuk menghasilkan *test case representative* [40]. Metode ini efektif untuk mengidentifikasi defek perangkat lunak yang mungkin terlewat dalam pengujian berbasis kode (*White Box Testing*), serta memastikan sistem memenuhi harapan pengguna secara fungsional dan operasional [23]. Hasil pengujian ini menjadi dasar verifikasi kesiapan sistem sebelum diimplementasikan ke dalam lingkungan operasional, sehingga meningkatkan kepercayaan pengguna terhadap keandalan sistem yang dikembangkan [40].