

BAB II

KAJIAN LITERATUR

2.1 Sistem Informasi Akademik

Sistem informasi akademik merupakan sistem yang digunakan untuk membantu pengelolaan kegiatan akademik pada lembaga pendidikan dengan memanfaatkan teknologi informasi. Sistem ini berfungsi untuk mengolah data akademik sehingga informasi yang dibutuhkan dapat diperoleh secara cepat dan tepat. Keberadaan sistem informasi akademik juga membantu sekolah dalam meningkatkan pelayanan administrasi kepada pengguna. Dengan adanya sistem tersebut, pengelolaan data akademik dapat dilakukan secara lebih teratur dan efisien [7].

Sistem informasi akademik pada umumnya mengelola berbagai data yang berkaitan dengan kegiatan sekolah. Data tersebut meliputi data siswa, data guru, data mata pelajaran, jadwal mengajar, serta data lainnya yang disesuaikan dengan kebutuhan lembaga pendidikan. Pengelolaan informasi dalam sistem akademik membantu data tersusun dengan lebih rapi dan mudah diakses. Pengelolaan data yang terpusat dapat memudahkan proses pencarian, pembaruan, dan penyusunan informasi akademik [8].

Beberapa lembaga pendidikan, kegiatan akademik masih dilakukan secara manual, baik dalam pengolahan maupun penyimpanan data. Kondisi tersebut menyebabkan proses pencarian data membutuhkan waktu lebih lama karena masih bergantung pada arsip atau berkas yang tersimpan secara terpisah. Selain itu, pengelolaan data yang belum terintegrasi juga membuat pekerjaan menjadi kurang efektif dan efisien. Penerapan sistem informasi akademik berbasis *web* diperlukan untuk membantu pengelolaan data akademik agar lebih efektif dan terintegrasi [9].

Penerapan sistem informasi akademik memberikan manfaat bagi pihak sekolah, guru, maupun siswa. Sistem ini dapat menyediakan informasi seperti data siswa, data guru, jadwal pelajaran, nilai siswa, dan informasi akademik lainnya di dalam satu sistem. Melalui sistem berbasis *web*, pengguna dapat mengakses informasi sesuai kebutuhan dengan lebih praktis. Dengan adanya sistem tersebut kegiatan akademik dapat dikelola dengan lebih baik, teratur, dan efisien [10].

2.2 Website

Website merupakan media berbasis internet yang digunakan untuk menyajikan informasi dalam bentuk halaman digital yang saling terhubung dan dapat diakses melalui jaringan internet. *Website* digunakan sebagai untuk menyampaikan informasi secara *online* kepada pengguna melalui perangkat yang terhubung dengan internet. Informasi yang ditampilkan melalui *website* dapat berupa teks, gambar, video, maupun data lainnya sesuai kebutuhan pengguna. Selain sebagai media informasi, *website* juga dapat dimanfaatkan sebagai sarana publikasi, komunikasi, dan penyampaian layanan digital kepada masyarakat [11].

Website saat ini banyak dimanfaatkan oleh individu, organisasi, maupun instansi untuk memperkenalkan profil, menyampaikan informasi, serta mendukung berbagai aktivitas pengguna. Pada sistem informasi, *website* berperan sebagai antarmuka yang menghubungkan pengguna dengan sistem sehingga proses akses dan pengelolaan informasi dapat dilakukan dengan lebih mudah. Pemanfaatan *website* juga mendukung penyediaan informasi yang lebih terpusat, terstruktur, dan dapat diakses sesuai kebutuhan pengguna. *Website* menjadi bagian penting dalam pengembangan sistem informasi karena mampu mendukung kebutuhan informasi dan layanan digital di berbagai bidang [12].

Berdasarkan karakteristik pengelolaan kontennya, *website* diklasifikasikan menjadi dua jenis [13]:

1. *Website Statis*

Website yang menampilkan informasi dengan isi yang bersifat tetap. Perubahan informasi dilakukan secara manual dengan mengubah kode pada halaman *website*. Jenis *website* ini umumnya memiliki struktur yang sederhana sehingga proses akses halaman dapat berjalan lebih cepat. *Website* ini umumnya digunakan untuk menyampaikan informasi yang jarang mengalami perubahan.

2. *Website Dinamis*

Website yang menampilkan informasi melalui proses pengolahan data oleh sistem serta terhubung dengan basis data. Konten pada *website* ini dapat diperbarui sesuai kebutuhan tanpa harus mengubah kode halaman secara langsung. Selain itu, *website* ini juga mendukung berbagai fitur interaktif, seperti *login* pengguna, pencarian data, dan pengelolaan informasi secara berkala. Dalam sistem informasi akademik, jenis *website* ini digunakan untuk menampilkan data siswa, jadwal pelajaran, presensi, nilai, dan informasi sekolah secara terintegrasi.

2.3 Basis Data

Basis data (*database*) merupakan kumpulan data yang saling berhubungan dan disimpan secara terstruktur dalam sistem penyimpanan berbasis komputer. Basis data digunakan untuk menyimpan data dalam jumlah besar agar dapat dikelola dan diakses kembali secara teratur. Penyimpanan data dilakukan secara terpusat sehingga data dapat digunakan secara konsisten dalam suatu sistem informasi. Dengan adanya basis data, proses pencarian, pembaruan, dan penggunaan data dapat dilakukan dengan lebih mudah [14].

Basis data juga berfungsi sebagai media penyimpanan utama dalam sistem informasi yang mendukung pengelolaan data secara berkelanjutan. Tujuan penggunaannya adalah menyediakan data yang akurat, terintegrasi, dan mudah diakses oleh pengguna sistem. Dan juga membantu proses pengolahan serta penyajian informasi sehingga kegiatan operasional dapat berjalan lebih tertata. Basis data digunakan pada berbagai bidang, seperti pendidikan, bisnis, dan pemerintahan [15]. Basis data terdiri dari beberapa komponen utama, yaitu [16]:

1. Data

Data adalah fakta atau informasi mentah yang disimpan di dalam basis data. Data menjadi bahan utama yang akan diolah oleh sistem untuk menghasilkan informasi.

2. *Database Management System* (DBMS)

Database Management System (DBMS) adalah perangkat lunak yang digunakan untuk mengelola basis data. DBMS berfungsi mengatur proses penyimpanan, pengolahan, pengambilan, serta pengamanan data. Selain itu, DBMS berperan dalam menjaga integritas dan konsistensi data agar data dapat digunakan secara terstruktur dalam sistem informasi.

3. Perangkat Keras

Perangkat keras adalah sarana pendukung yang digunakan untuk menjalankan basis data dan DBMS. Perangkat keras meliputi *server*, media penyimpanan, dan perangkat pendukung lainnya yang berfungsi sebagai tempat penyimpanan data serta menjalankan proses pengolahan data.

4. Pengguna Sistem

Pengguna sistem adalah pihak yang berinteraksi langsung dengan basis data melalui aplikasi atau sistem informasi. Pengguna sistem terdiri dari administrator, guru, staf administrasi, dan pihak lain sesuai dengan hak akses yang diberikan. Peran pengguna sistem sangat penting dalam proses pengelolaan dan pemanfaatan data yang tersimpan dalam basis data.

Pada sistem informasi akademik, basis data berperan sebagai pusat penyimpanan data peserta didik, data pendidik, data kelas, jadwal pembelajaran, presensi, penilaian, serta informasi sekolah. Hubungan antar data tersebut memudahkan pengelolaan dan penyajian informasi akademik secara terintegrasi. Basis data juga membantu menjaga ketepatan dan keteraturan data akademik, terutama pada *website* yang digunakan oleh banyak pengguna [17].

2.4 Metode *Waterfall*

Metode *Waterfall* merupakan salah satu metode pengembangan perangkat lunak yang termasuk ke dalam *Software Development Life Cycle* (SDLC). Metode ini menggunakan pendekatan linier, sehingga setiap proses dikerjakan secara berurutan, dimana satu tahapan harus selesai sebelum tahapan berikutnya dimulai. Dengan pola tersebut, metode *Waterfall* memiliki alur kerja yang terstruktur dan memudahkan proses dokumentasi selama proses pengembangan sistem informasi. Tahapan pengembangan sistem dengan menggunakan metode *Waterfall* terdiri dari beberapa langkah, antara lain [18]:

1. Analisis Kebutuhan (*Requirement Analysis*)

Tahap ini dilakukan untuk mengumpulkan dan mengidentifikasi kebutuhan sistem berdasarkan kebutuhan pengguna serta tujuan sistem yang akan dibangun.

2. Perancangan Sistem (*System Design*)

Tahap ini dilakukan berdasarkan hasil analisis kebutuhan. Kegiatan pada tahap ini meliputi penyusunan struktur sistem, alur proses, rancangan basis data, dan tampilan sistem.

3. Implementasi (*Implementation*)

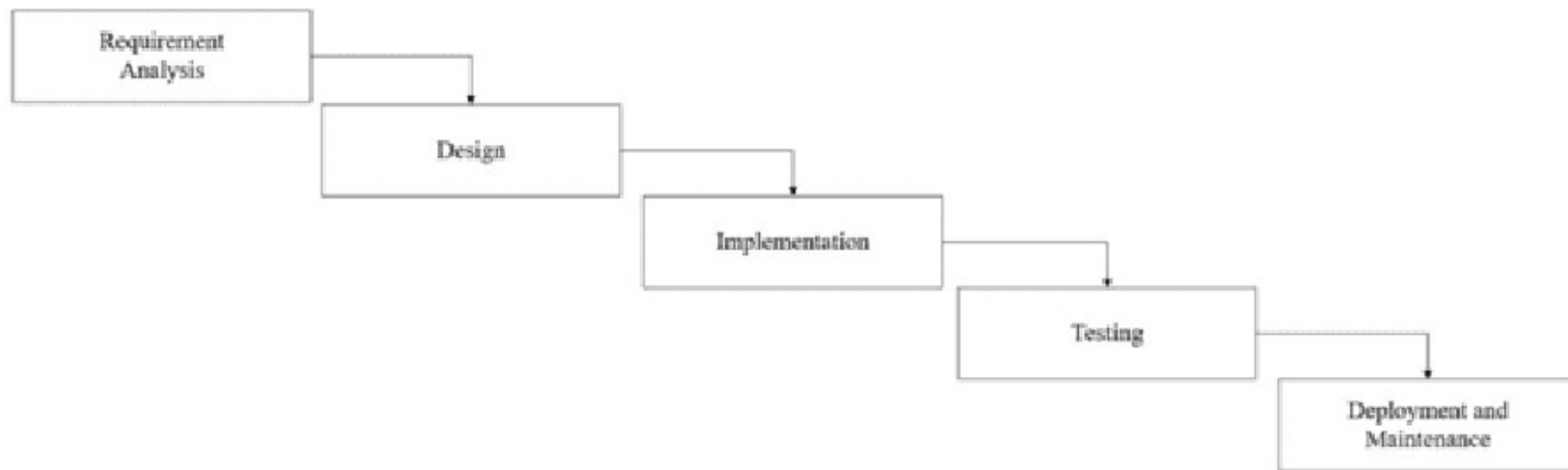
Tahap ini dilakukan dengan mengubah hasil rancangan ke dalam bentuk perangkat lunak melalui penulisan kode program berdasarkan desain yang telah dibuat.

4. Pengujian Sistem (*Testing*)

Tahap ini dilakukan untuk memeriksa fungsi sistem dan memastikan hasil pengembangan telah sesuai dengan kebutuhan yang ditetapkan.

5. Penerapan (*Deployment*) dan Pemeliharaan (*Maintenance*)

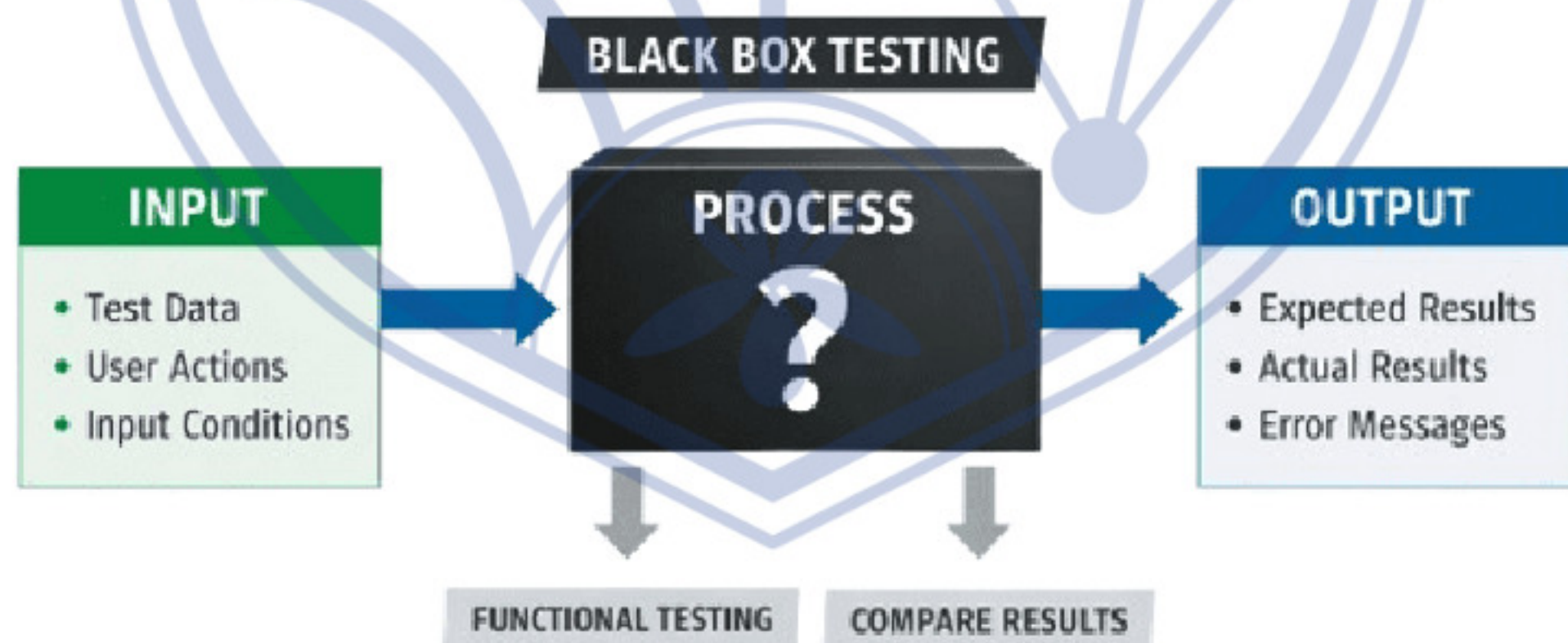
Tahap ini dilakukan setelah sistem selesai dikembangkan dan diuji. Sistem mulai digunakan pada lingkungan operasional. Selanjutnya akan dilakukan pemeliharaan untuk memperbaiki kesalahan, menyesuaikan kebutuhan baru, dan menjaga kinerja sistem agar tetap berjalan dengan baik.



Gambar 2.1 Metode *Waterfall* [18]

2.5 Black Box Testing

Black Box Testing merupakan metode pengujian perangkat lunak yang berfokus pada fungsi sistem tanpa melihat struktur internal program atau kode yang digunakan. Pengujian dilakukan dengan memberikan data masukan (*input*) kemudian mengamati keluaran (*output*) yang dihasilkan sistem. Hasil keluaran tersebut dibandingkan dengan hasil yang diharapkan untuk mengetahui apakah fungsi sistem telah berjalan sesuai kebutuhan. Metode ini digunakan untuk menilai fungsi sistem dari sudut pandang pengguna. Pada *Black Box Testing*, pengujian dilakukan berdasarkan fitur yang tersedia di sistem. Setiap fungsi diuji menggunakan beberapa skenario, seperti data *valid*, data tidak *valid*, data kosong, maupun kondisi lain yang dapat memengaruhi proses sistem. Pengujian ini membantu menemukan kesalahan pada validasi data, proses sistem, maupun hasil keluaran yang ditampilkan [19].



Gambar 2.2 Black Box Testing[19]

Penerapan *Black Box Testing* menggunakan *test case* sebagai acuan pengujian. *Test case* berisi langkah pengujian, data masukan, dan hasil yang diharapkan dari setiap fitur sistem. Penyusunan *test case* membantu proses pengujian agar dilakukan secara terarah dan

teratur. Hasil pengujian dicatat untuk mengetahui apakah fitur berjalan sesuai harapan atau masih perlu diperbaiki [19].

Dalam pelaksanaannya, *Black Box Testing* menggunakan beberapa teknik pengujian untuk menyusun dan menjalankan *test case* secara sistematis, antara lain [20]:

1. *Equivalence Partitioning*

Teknik ini membagi data *input* ke dalam beberapa kelompok yang memiliki karakteristik sama. Pengujian dilakukan dengan memilih perwakilan dari setiap kelompok, sehingga jumlah pengujian menjadi lebih efisien.

2. *Boundary Value Analysis*

Teknik ini berfokus pada pengujian nilai batas minimum, maksimum, dan nilai di sekitar batas input. Teknik ini digunakan karena kesalahan sistem sering terjadi pada batas data.

3. *Decision Table Testing*

Teknik ini digunakan untuk menguji kombinasi beberapa kondisi dan keputusan dalam sistem. Tabel keputusan membantu penguji menyusun skenario pengujian sesuai aturan yang berlaku pada sistem.

4. *Error Guessing*

Teknik ini dilakukan berdasarkan pengalaman dan pemahaman teknis penguji dalam mengidentifikasi bagian sistem yang berpotensi mengalami kesalahan. Pendekatan ini digunakan untuk menemukan kesalahan yang sulit dideteksi melalui teknik pengujian formal.

Berikut contoh penggunaan *Black Box Testing* pada modul registrasi dan *login* yang dapat dilihat pada Tabel 2.1 [21].

Tabel 2.1 Contoh *Test Case* Black Box Testing[21]

Uji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
1	Pengguna mengisi <i>email</i> dan <i>password</i> yang benar	Sistem menampilkan halaman <i>dashboard</i>	<i>Dashboard</i> berhasil ditampilkan	Sesuai
2	Pengguna mengisi <i>email</i> yang salah	Sistem menampilkan pesan <i>error</i>	Pesan <i>error</i> ditampilkan	Sesuai

Uji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
3	Pengguna melakukan registrasi akun baru	Data tersimpan dan sistem kembali ke <i>login</i>	Registrasi berhasil	Sesuai
4	Pengguna menekan <i>login</i> tanpa mengisi data	Sistem menampilkan peringatan	Peringatan tidak muncul	Tidak Sesuai
5	Pengguna memilih menu lupa <i>Password</i>	Sistem mengirim <i>email reset Password</i>	<i>Email</i> tidak terkirim	Tidak Sesuai

2.6 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) merupakan metode pengujian yang melibatkan pengguna akhir untuk menilai kesesuaian sistem dengan kebutuhan pengguna serta kemampuan sistem dalam membantu menyelesaikan permasalahan yang dihadapi pengguna. Pengujian ini dilakukan setelah sistem selesai dibangun dan sebelum digunakan secara penuh. Pengujian ini melibatkan responden sebagai perwakilan pengguna sistem sesuai dengan tempat sistem tersebut digunakan. Pelaksanaan UAT dapat dilakukan dengan cara pengguna mencoba sistem secara langsung, kemudian memberikan penilaian melalui kuesioner menggunakan skala Likert [22].

Penyusunan indikator penilaian pada kuesioner dapat mengacu pada model kualitas perangkat lunak ISO/IEC 25010. Model ini digunakan untuk menilai kualitas perangkat lunak berdasarkan beberapa karakteristik, yaitu *functional suitability* untuk menilai kesesuaian fungsi sistem, *performance efficiency* untuk menilai kinerja sistem, *compatibility* untuk menilai kesesuaian sistem dengan lingkungan tertentu, *usability* untuk menilai kemudahan penggunaan, *reliability* untuk menilai keandalan sistem, *security* untuk menilai keamanan sistem, *maintainability* untuk menilai kemudahan sistem dalam proses pemeliharaan dan pengembangan, dan *portability* untuk menilai kemampuan sistem digunakan pada lingkungan yang berbeda. Dalam penerapannya, karakteristik ISO/IEC 25010 dapat disesuaikan dengan kebutuhan pengujian sistem [23].

2.7 Teknik Pengembangan Sistem

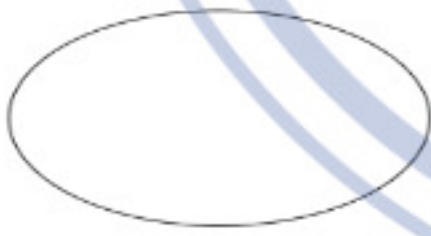
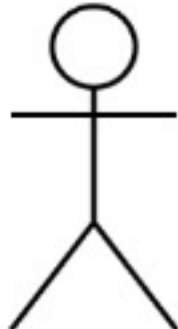
Teknik pengembangan sistem merupakan metode yang digunakan dalam proses analisis dan perancangan sistem informasi agar sistem yang dibangun dapat menyesuaikan kebutuhan pengguna. Dalam penelitian ini, teknik yang digunakan meliputi *use case diagram*, *activity diagram*, analisis PIECES, *flowchart*, dan *fishbone diagram*. Teknik-teknik tersebut digunakan untuk menggambarkan kebutuhan sistem, menjelaskan alur prosesnya, dan mengidentifikasi penyebab permasalahan yang digunakan sebagai dasar pertimbangan dalam merancang sistem [24].

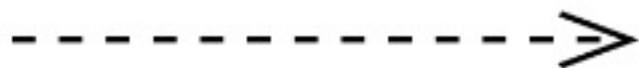
2.7.1 Use Case Diagram

Use case diagram merupakan salah satu diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan interaksi antara pengguna dengan sistem. Diagram ini menjelaskan bagaimana sistem berinteraksi dengan aktor melalui fungsi-fungsi yang disediakan. *Use case diagram* berfokus pada kebutuhan fungsional sistem tanpa membahas proses internal yang terjadi di dalam sistem. Dengan adanya *use case diagram*, batasan sistem dan ruang lingkup interaksi dapat dipahami dengan lebih jelas [25].

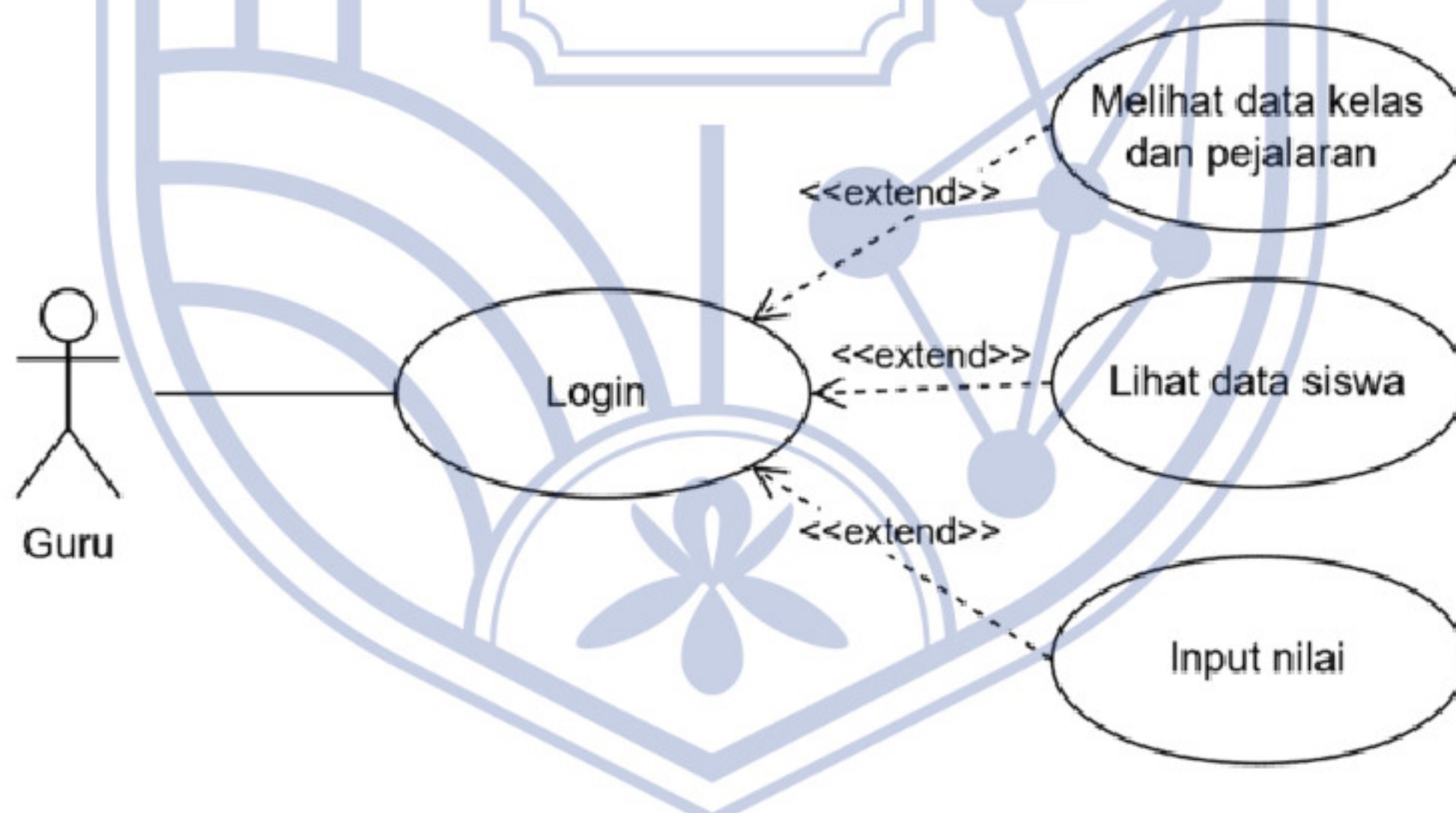
Use case diagram terdiri dari beberapa komponen utama yang digambarkan melalui simbol tertentu. Setiap simbol menunjukkan hubungan antara aktor dan sistem. Simbol-simbol tersebut dapat dilihat pada Tabel 2.2 [25].

Tabel 2.2 Simbol *Use Case Diagram* [25]

Simbol	Keterangan
<i>Use case</i> 	Fungsi atau layanan yang disediakan oleh sistem. Biasanya penamaan <i>use case</i> diawali dengan kata kerja yang menggambarkan aktivitas atau proses yang dilakukan oleh sistem.
<i>Actor</i> 	Entitas yang berinteraksi dengan sistem. Aktor dapat berupa manusia, proses, maupun sistem lain yang berada di luar sistem yang dikembangkan.

Simbol	Keterangan
<i>Association</i> 	Hubungan komunikasi antara aktor dan <i>use case</i> . Relasi ini menunjukkan keterlibatan aktor dalam menjalankan fungsi tertentu di dalam sistem.
<i>Extend</i> 	Relasi tambahan yang menunjukkan bahwa suatu <i>use case</i> dapat memperluas perilaku <i>use case</i> utama dalam kondisi tertentu.
<i>Generalization</i> 	Menggambarkan hubungan umum dan khusus antara dua <i>case</i> atau dua aktor.
<i>Include</i> 	Menunjukkan bahwa suatu <i>use case</i> selalu menyertakan <i>use case</i> lain sebagai bagian dari prosesnya.

Berikut contoh penggambaran *use case diagram* untuk *login guru* [26].



Gambar 2.3 Contoh Use Case Diagram *Login Guru* [26]

1. Aktor Guru

Guru merupakan pengguna yang memiliki hak akses untuk masuk ke dalam sistem informasi akademik sekolah. Setelah *login*, guru dapat mengakses fitur akademik seperti melihat data dan menginput nilai.

2. Use Case Login

Proses autentikasi yang dilakukan oleh guru dengan memasukkan *username* dan *password* agar dapat masuk ke dalam sistem. Proses ini menjadi tahap awal sebelum menggunakan fitur lainnya.

3. Use Case Melihat Data Kelas dan Pelajaran

Setelah berhasil *login*, guru dapat melihat data kelas yang akan diajarkan serta mata pelajaran yang diampu. Informasi ini membantu guru dalam kegiatan pembelajaran.

4. Use Case Lihat Data Siswa

Setelah *login*, guru dapat melihat data siswa sesuai kelas yang diajarkan. Menu ini digunakan untuk mengetahui informasi siswa pada kelas yang diajarkannya.

5. Use Case Input Nilai

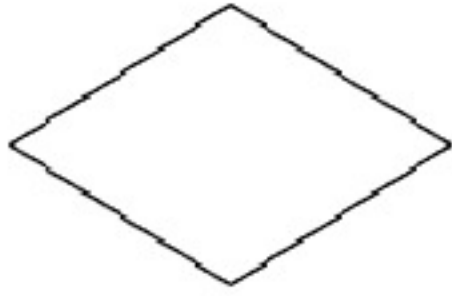
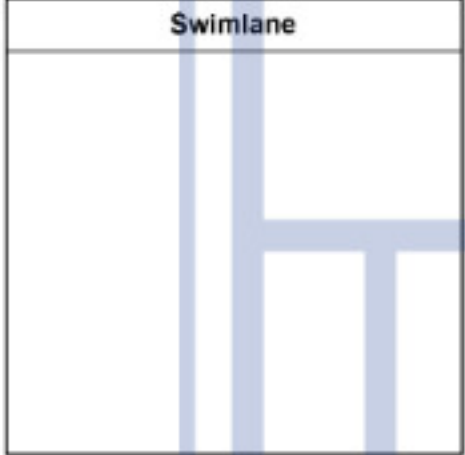
Setelah berhasil masuk ke sistem, guru juga dapat menginput nilai siswa berdasarkan mata pelajaran yang diampu. Proses ini dilakukan melalui fitur penilaian.

2.7.2 Activity Diagram

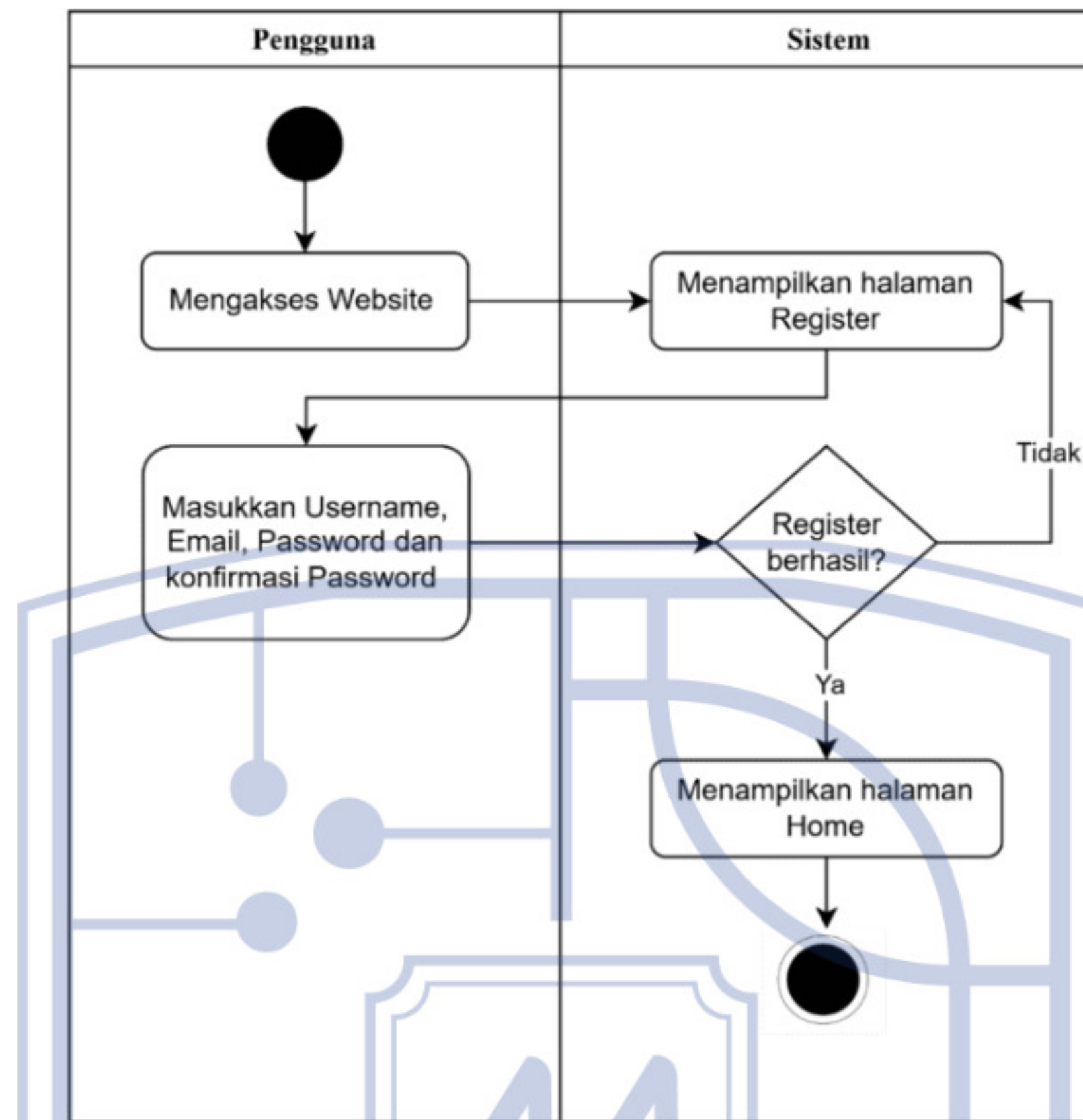
Activity diagram merupakan salah satu diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan alur aktivitas atau urutan proses yang terjadi dalam suatu sistem. Diagram ini menampilkan langkah-langkah kegiatan dari awal sampai akhir proses, termasuk percabangan dan penggabungan alur proses. Dengan diagram ini, alur kerja antara sistem dan pengguna dapat dipahami dengan jelas. Beberapa simbol yang digunakan dalam *activity diagram* untuk menggambarkan alur aktivitas sistem dapat dilihat pada Tabel 2.3 [27].

Tabel 2.3 Simbol-Simbol *Activity Diagram* [27]

Simbol	Keterangan
Status awal 	Simbol ini menunjukkan titik awal (<i>Initial Node</i>) dimulainya aktivitas atau proses dalam sistem.
Aktivitas 	Simbol ini menunjukkan aktivitas atau proses yang terjadi dalam sistem. Biasanya dideskripsikan menggunakan kata kerja.
Percabangan/ <i>Decision</i>	Simbol ini menunjukkan proses pengambilan keputusan yang membagi alur menjadi dua atau lebih pilihan.

Simbol	Keterangan
	
Penggabungan/ <i>Join</i> 	Simbol ini menunjukkan titik penggabungan (<i>join</i>) dimana dua atau lebih alur aktivitas paralel bersatu kembali menjadi satu alur tunggal.
Status akhir 	Simbol ini menunjukkan titik akhir (<i>Activity Final Node</i>) yang menandakan proses dalam <i>activity diagram</i> telah selesai.
<i>Swimlane</i> 	Simbol ini digunakan untuk membagi <i>activity diagram</i> ke dalam beberapa bagian berdasarkan peran atau aktor, sehingga tanggung jawab setiap aktivitas dapat terlihat dengan jelas.

Berikut adalah contoh *activity diagram* yang menggambarkan proses registrasi pengguna pada sistem [27].



Gambar 2.4 Contoh *Activity Diagram* Proses Registrasi [27]

2.7.3 Analisis PIECES

Analisis PIECES merupakan salah satu teknik analisis sistem yang digunakan untuk mengidentifikasi permasalahan dan kebutuhan sistem secara terstruktur. Analisis ini membantu dalam mengevaluasi kondisi sistem yang sedang berjalan dengan meninjau beberapa aspek penting yang memengaruhi kinerja dan kualitas sistem. Metode PIECES sering digunakan pada tahap analisis kebutuhan untuk memperoleh gambaran yang jelas mengenai kelemahan sistem serta peluang perbaikan yang dapat dilakukan dalam pengembangan sistem informasi. Analisis PIECES terdiri dari enam komponen utama, antara lain [28]:

1. *Performance* (Kinerja)

Performance menilai kinerja sistem dalam memproses data dan menghasilkan *output* sesuai kebutuhan pengguna.

2. *Information* (Informasi)

Information menilai kualitas informasi yang dihasilkan sistem berdasarkan ketepatan, kelengkapan, dan keakuratan data.

3. *Economy* (Ekonomi)

Economy berkaitan dengan perbandingan antara biaya yang dikeluarkan dan manfaat yang diperoleh dari penggunaan sistem.

4. *Control* (Pengendalian)

Control menilai pengendalian dan keamanan sistem, termasuk pengaturan akses serta perlindungan data dari kesalahan atau penyalahgunaan.

5. *Efficiency* (Efisiensi)

Efficiency menilai pemanfaatan sumber daya sistem, seperti waktu dan tenaga, dalam mendukung proses operasional.

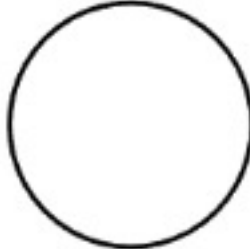
6. *Service* (Pelayanan)

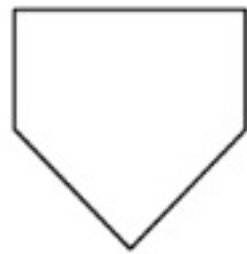
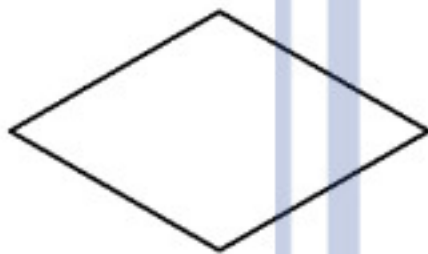
Service menilai kualitas layanan sistem kepada pengguna, termasuk kemudahan penggunaan dan kenyamanan dalam mengakses sistem.

2.7.4 Flowchart

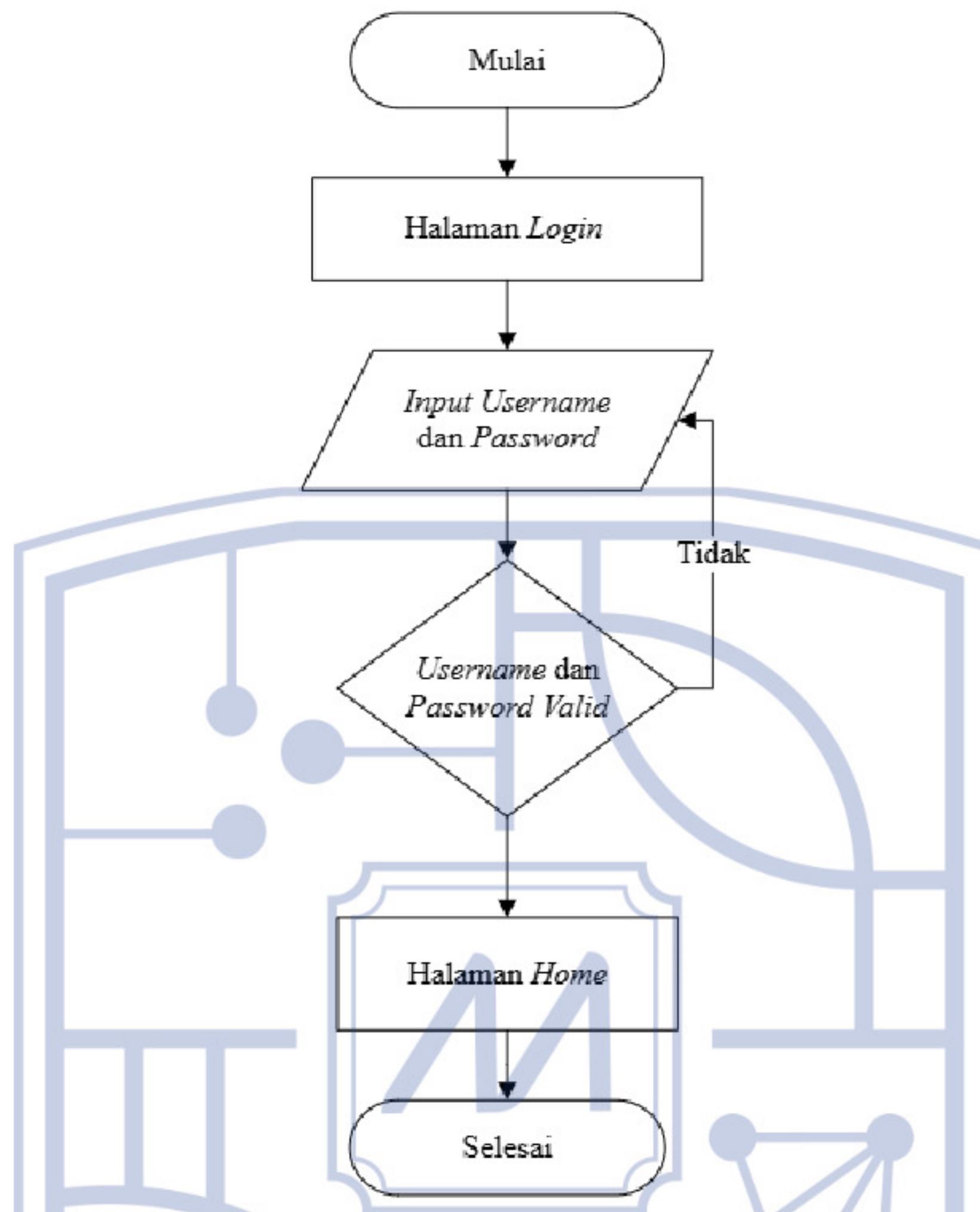
Flowchart adalah diagram yang digunakan untuk menggambarkan urutan proses, logika atau langkah-langkah penyelesaian masalah secara visual. Diagram ini menampilkan alur kerja dalam bentuk simbol sehingga lebih mudah dipahami dibandingkan penjelasan dalam bentuk teks. *Flowchart* digunakan untuk menjelaskan prosedur secara sistematis dan terstruktur agar mudah dipahami dengan jelas oleh pembaca maupun pengembang sistem. Beberapa simbol yang umum digunakan dalam *flowchart* seperti pada Tabel 2.4 [29].

Tabel 2.4 *Flowchart* [29]

Simbol	Arti	Keterangan
	<i>Document</i>	Digunakan untuk menunjukkan masukan dan keluaran yang berasal dari dokumen
	<i>Flow</i>	Berfungsi untuk menghubungkan simbol yang satu dengan yang lainnya, menyatakan arus suatu proses
	<i>On-Page Reference</i>	Digunakan untuk menunjukkan sambungan dari proses yang satu ke proses berikutnya di halaman yang berbeda

Simbol	Arti	Keterangan
	<i>Off-Page Reference</i>	Digunakan untuk memulai atau mengakhiri program
	<i>Terminator</i>	Digunakan untuk menunjukkan pengolahan yang akan dilakukan dalam komputer
	<i>Processing</i>	Digunakan untuk menunjukkan pengolahan yang akan dilakukan dalam komputer
	<i>Decision</i>	Digunakan untuk memilih proses yang akan dilakukan berdasarkan kondisi tertentu
	<i>Input/Output</i>	Digunakan untuk menunjukkan <i>input</i> dan <i>output</i> tanpa melihat jenisnya
	<i>Manual Operation</i>	Digunakan untuk menunjukkan proses pemasukan data yang dilakukan secara manual oleh pengguna melalui perangkat <i>input</i>
	<i>Predefined Process</i>	Digunakan untuk menunjukkan suatu proses yang telah didefinisikan sebelumnya dan dijelaskan lebih rinci pada bagian atau diagram lain
	<i>Preparation</i>	Digunakan untuk menunjukkan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal

Gambar 2.5 merupakan contoh penerapan *flowchart* pada proses *login* pengguna. *Flowchart* ini menggambarkan alur proses autentikasi mulai dari pengguna mengakses halaman *login*, memasukkan *username* dan *Password*, hingga sistem melakukan validasi dan memberikan respons sesuai dengan hasil pemeriksaan data [29].



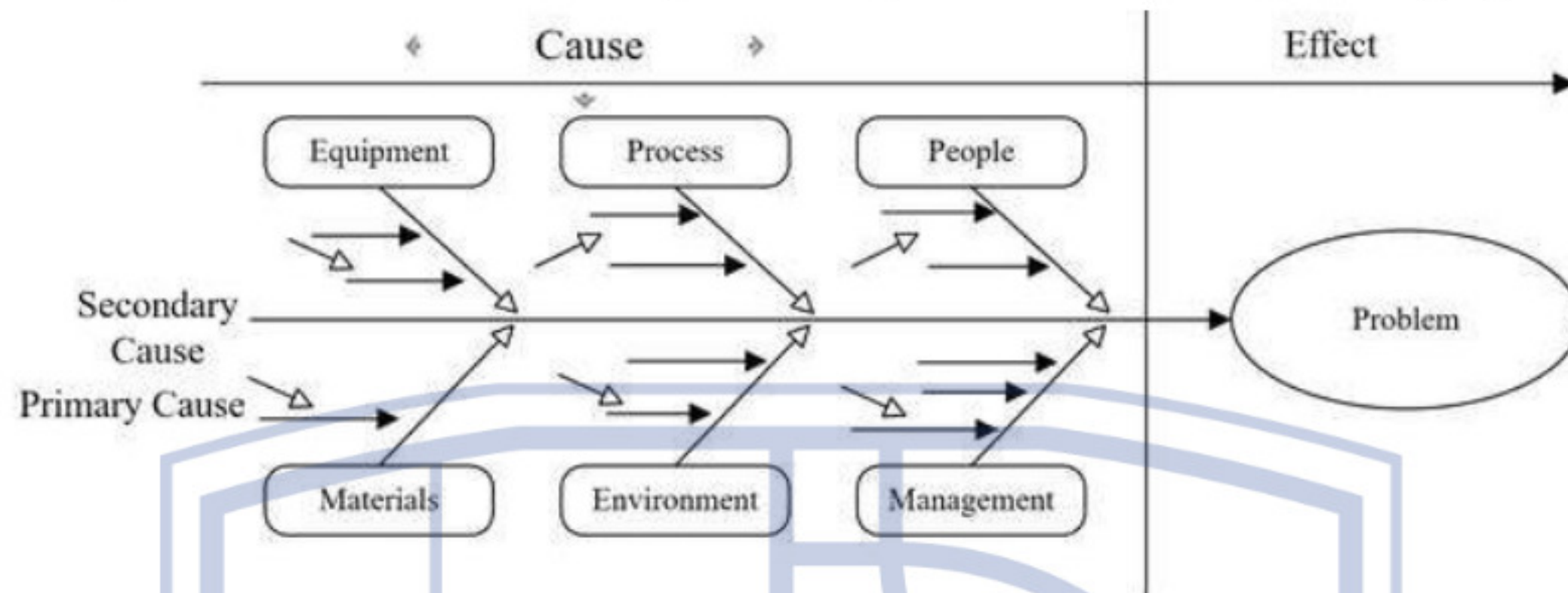
Gambar 2.5 Flowchart Proses Login [29]

2.7.5 Fishbone Diagram

Fishbone diagram merupakan alat bantu visual yang digunakan untuk mengidentifikasi dan menganalisis berbagai faktor penyebab suatu permasalahan. Diagram ini juga dikenal sebagai *Ishikawa Diagram* atau *Cause and Effect Diagram* karena menggambarkan hubungan sebab-akibat antara masalah utama dan faktor penyebabnya. Struktur diagram ini menyerupai tulang ikan, dimana masalah utama ditempatkan pada bagian kepala diagram, sedangkan faktor penyebab disusun dalam bentuk cabang-cabang yang terhubung pada tulang utama. *Fishbone diagram* berfungsi untuk memetakan hubungan antara masalah (*effect*) dan kategori penyebab (*cause*) sehingga proses analisis akar masalah dapat dilakukan secara terstruktur [30].

Komponen utama diagram terdiri dari masalah utama yang diletakkan pada sisi kanan, kategori penyebab utama ditempatkan di cabang diagram, serta penyebab yang lebih spesifik dari setiap kategori. Pengelompokan penyebab dapat disesuaikan dengan kondisi

permasalahan, seperti *Man* (Manusia), *Machine* (Mesin), *Method* (Metode), *Material* (Bahan), *Measurement* (Pengukuran), dan *Environment* (Lingkungan). Pengelompokan ini membantu proses analisis masalah menjadi lebih jelas dan mudah dipahami [30].



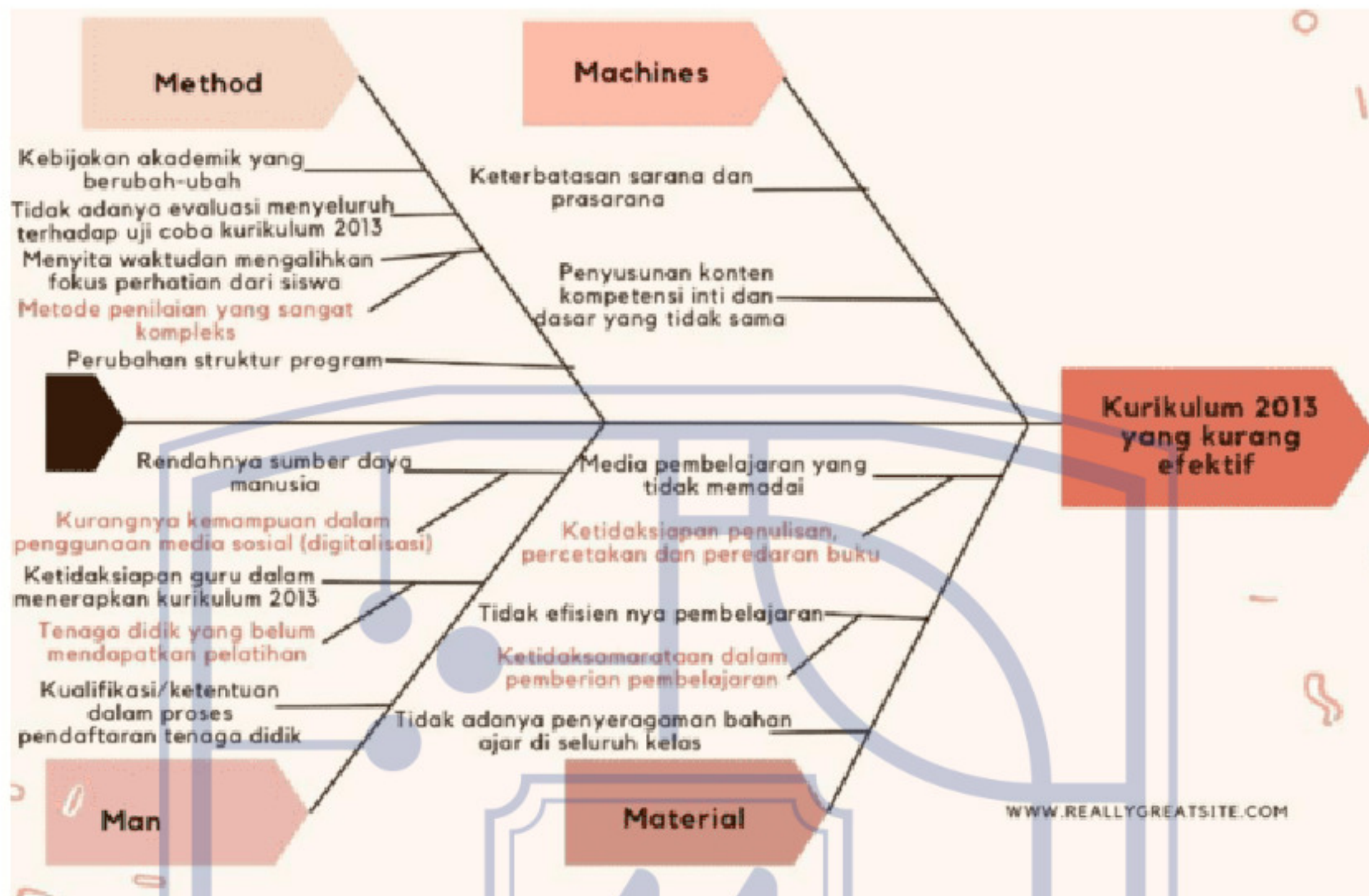
Gambar 2.6 Struktur *Fishbone Diagram* [30]

Proses penyusunan *fishbone diagram* dilakukan melalui beberapa tahapan sebagai berikut [30]:

1. Menentukan permasalahan utama
Permasalahan utama yang akan dianalisis ditetapkan terlebih dahulu dan diletakkan pada bagian kepala *fishbone diagram* sebagai fokus analisis.
2. Menyusun kerangka *fishbone diagram*
Kerangka diagram dibuat menyerupai tulang ikan yang terdiri dari satu garis utama dan beberapa cabang sebagai tempat pengelompokan penyebab masalah.
3. Menentukan kelompok penyebab utama
Kategori penyebab ditetapkan berdasarkan konteks permasalahan yang dianalisis, seperti kategori 6M atau kategori lain sesuai kebutuhan.
4. Mengidentifikasi penyebab spesifik
Penyebab-penyebab yang berkontribusi terhadap masalah dikumpulkan melalui proses *brainstorming* atau diskusi dan ditempatkan pada cabang diagram.
5. Menganalisis akar penyebab masalah
Penyebab yang telah diidentifikasi dievaluasi untuk menentukan faktor yang paling dominan dan berpengaruh terhadap permasalahan utama.
6. Menyusun rencana perbaikan
Langkah perbaikan ditetapkan sebagai tindak lanjut untuk mengatasi penyebab utama yang telah ditentukan.

Berikut contoh penggunaan *fishbone diagram* untuk menganalisis penyebab permasalahan “Kurikulum 2013 yang kurang efektif”. Pada diagram ini, permasalahan

utama ditempatkan pada bagian kepala ikan, sedangkan faktor-faktor penyebabnya digambarkan sebagai tulang-tulang yang mengarah ke masalah utama [30].



Gambar 2.7 Contoh Penggunaan *Fishbone Diagram* [30]

1. *Man*

Faktor manusia berkaitan dengan kesiapan dan kompetensi tenaga pendidik. Contohnya adalah kurangnya pelatihan bagi guru, keterbatasan kemampuan dalam penggunaan media digital, serta kesiapan dalam menerapkan kurikulum 2013.

2. *Method*

Faktor metode berkaitan dengan kebijakan dan proses pelaksanaan kurikulum. Beberapa penyebab yang termasuk dalam kategori ini, seperti perubahan kebijakan akademik yang terus-menerus, kurangnya evaluasi menyeluruh, serta metode penilaian yang kompleks.

3. *Machine*

Kategori ini berhubungan dengan ketersediaan fasilitas pendukung pembelajaran. Permasalahan yang muncul antara lain keterbatasan sarana dan prasarana serta penyusunan konten pembelajaran yang belum seragam.

4. *Material*

Kategori ini mencakup media dan bahan pembelajaran yang digunakan. Beberapa permasalahan yang muncul antara lain media pembelajaran yang kurang memadai, ketidaksiapan percetakan dan distribusi buku, serta tidak adanya penyeragaman bahan ajar di seluruh kelas.

2.8 SMK Swasta Parulian 1 Medan

SMK Swasta Parulian 1 Medan merupakan sekolah menengah kejuruan berstatus swasta dengan akreditasi A yang berlokasi di Jl. Stadion Teladan, Jl. Jati I No. 23, Kelurahan Teladan Barat, Kota Medan, Sumatera Utara. Saat ini, kepemimpinan sekolah dipegang oleh Yenni Hartati Tumanggor, S.Pd. selaku Kepala Sekolah, dengan dibantu oleh Theresia Magdalena, M.Pd. sebagai Wakil Kepala Sekolah. Selain itu, posisi Wakil Kepala Sekolah Bidang Kurikulum dijabat oleh Astipo dan bagian Tata Usaha dikelola oleh Cristela Sitinjak.

Secara operasional, SMK Swasta Parulian 1 Medan menyelenggarakan kegiatan belajar mengajar untuk tiga tingkatan kelas dengan total penggunaan 6 ruang kelas. Sekolah ini menawarkan empat program keahlian, meliputi Otomatis Tata Kelola Perkantoran (OTKP), Akuntansi dan Keuangan Lembaga (AKL), Teknik Jaringan Komputer dan Telekomunikasi (TJKT), dan Pengembangan Perangkat Lunak dan *Game* (PPLG). Berdasarkan data pada Tahun Ajaran 2025/2026, tercatat jumlah keseluruhan peserta didik di sekolah ini adalah sebanyak 145 siswa, yang didukung oleh tenaga pendidik berjumlah 21 orang guru.



Gambar 2.8 SMK Swasta Parulian 1 Medan