

BAB II

KAJIAN LITERATUR

2.1 Konsep Sistem Informasi

Sistem informasi merupakan suatu sistem yang terdiri dari komponen-komponen yang saling terintegrasi dalam mengumpulkan, mengolah, menyimpan, dan mendistribusikan data menjadi informasi yang relevan. Sistem ini tidak hanya berfungsi sebagai alat pengolahan data, namun juga sebagai sarana yang mendukung kelancaran aktivitas operasional bisnis [3]. Melalui sistem informasi, data yang sebelumnya bersifat mentah dapat diolah menjadi informasi yang memiliki makna dan dapat digunakan sebagai dasar dalam pengambilan keputusan[4]. Selain itu, sistem informasi juga berperan dalam meningkatkan kualitas pengelolaan data dengan menyediakan mekanisme pencatatan yang terstruktur dan terdokumentasi dengan baik.

Secara konseptual, aktivitas dasar sistem informasi memiliki empat elemen utama yang terdiri dari [4]:

1. *Input* (masukan) : merupakan tahap awal pengumpulan data dari berbagai sumber. Data dapat berasal dari luar atau dari dalam organisasi. Data dicatat di dokumen dasar. Proses selanjutnya adalah memasukkan data tersebut ke dalam sistem informasi.
2. *Processing* (pengolahan) : Data yang telah dikumpulkan kemudian diolah pada tahap *processing*, pengolahan data untuk menghasilkan informasi yang bermanfaat.
3. *Output* (keluaran) : yaitu informasi yang telah diolah dan memiliki makna bagi pengguna, misalnya dalam bentuk laporan transaksi, data persediaan, maupun ringkasan aktivitas operasional.
4. *Feedback* (umpan balik) : digunakan untuk mengevaluasi dan memperbaiki proses yang sedang berjalan, sehingga sistem dapat terus disesuaikan dengan kebutuhan pengguna dan kondisi operasional yang dinamis.

2.1.1 Sistem Informasi Berbasis Komputer

Aplikasi sistem informasi berbasis komputer merupakan bentuk penerapan nyata dari konsep sistem informasi yang diimplementasikan melalui perangkat lunak (*software*) dan perangkat keras (*hardware*). Sistem ini dirancang untuk membantu proses pengelolaan, pengolahan, dan penyajian data secara terstruktur guna mendukung pengambilan keputusan dalam organisasi. Penerapannya mampu meningkatkan efisiensi pada setiap tahapan

kegiatan operasional serta memperluas kemampuan bisnis. Dalam konteks yang lebih spesifik, hal tersebut dikenal sebagai *Computer-Based Information System* (CBIS). CBIS merupakan sistem yang memanfaatkan teknologi komputer untuk mengotomisasi sebagian atau seluruh proses pengolahan data dengan tujuan meningkatkan efisiensi dan efektivitas pengelolaan informasi serta komunikasi dalam organisasi. Melalui penerapan CBIS, informasi dapat disajikan secara lebih cepat, akurat, dan relevan sehingga dapat memenuhi kebutuhan pengguna maupun organisasi secara keseluruhan[5].

Dalam implementasinya, CBIS telah diterapkan secara luas di berbagai sektor kehidupan, mulai dari bidang bisnis, pendidikan, kesehatan, pemerintahan, hingga industri. Penerapan CBIS terbukti mampu meningkatkan produktivitas organisasi dengan mengotomisasi proses-proses yang sebelumnya dilakukan secara manual, sehingga meminimalkan potensi kesalahan manusia (*human error*) sekaligus mempercepat alur kerja secara keseluruhan. Komponen dasar dari CBIS terdiri atas elemen-elemen yang saling berinteraksi membentuk suatu kesatuan sistem, yaitu sebagai berikut[6]:

1. Perangkat Keras (*Hardware*)

Hardware adalah perangkat fisik komputer seperti prosesor, monitor, *keyboard*, dan printer yang berfungsi menerima, memproses, dan menampilkan data serta informasi.

2. Perangkat Lunak (*Software*)

Software adalah perangkat lunak berupa program atau kumpulan program yang memungkinkan perangkat keras untuk memproses dan mengolah data menjadi informasi.

3. Basis Data (*Database*)

Database adalah kumpulan data yang tersimpan secara terstruktur dalam bentuk dokumen atau tabel yang saling berhubungan, digunakan sebagai sumber data utama dalam sistem.

4. Jaringan (*Network*)

Network adalah sistem penghubung berkabel (*wireline*) maupun nirkabel (*wireless*) yang memungkinkan antar komputer saling berbagi sumber daya dan informasi.

5. Prosedur (*Procedure*)

Procedure adalah serangkaian instruksi dan aturan yang mengatur cara komponen-komponen di atas digunakan bersama untuk mengolah data dan menghasilkan *output* yang diinginkan.

6. Orang (*People*)

People adalah pengguna yang mengoperasikan perangkat keras dan perangkat lunak, berinteraksi dengan sistem, serta memanfaatkan hasil keluaran dari sistem tersebut.

2.1.2 Basis Data

Basis data atau *database* merupakan sekumpulan data yang saling berhubungan dan disusun secara sistematis untuk memenuhi kebutuhan informasi dalam suatu organisasi atau perusahaan. Basis data dirancang agar data dapat disimpan, dikelola, diakses, dan diperbarui secara efisien oleh pengguna tanpa menimbulkan redundansi maupun inkonsistensi data. Dalam pengembangan sistem informasi, basis data berperan sebagai pusat penyimpanan yang mendukung proses operasional, pengambilan keputusan, dan penyajian laporan secara akurat. Penelitian terbaru menunjukkan bahwa perancangan basis data yang baik akan meningkatkan konsistensi data, integritas, serta efisiensi proses bisnis dalam informasi[7].

Untuk mengelola basis data dapat menggunakan *Database Management System* (DBMS), yaitu perangkat lunak yang berfungsi untuk mendefinisikan, membangun, memelihara, dan mengontrol akses data dalam basis data. DBMS memungkinkan pengguna melakukan pengelolaan data melalui beberapa fasilitas utama, yaitu [8]:

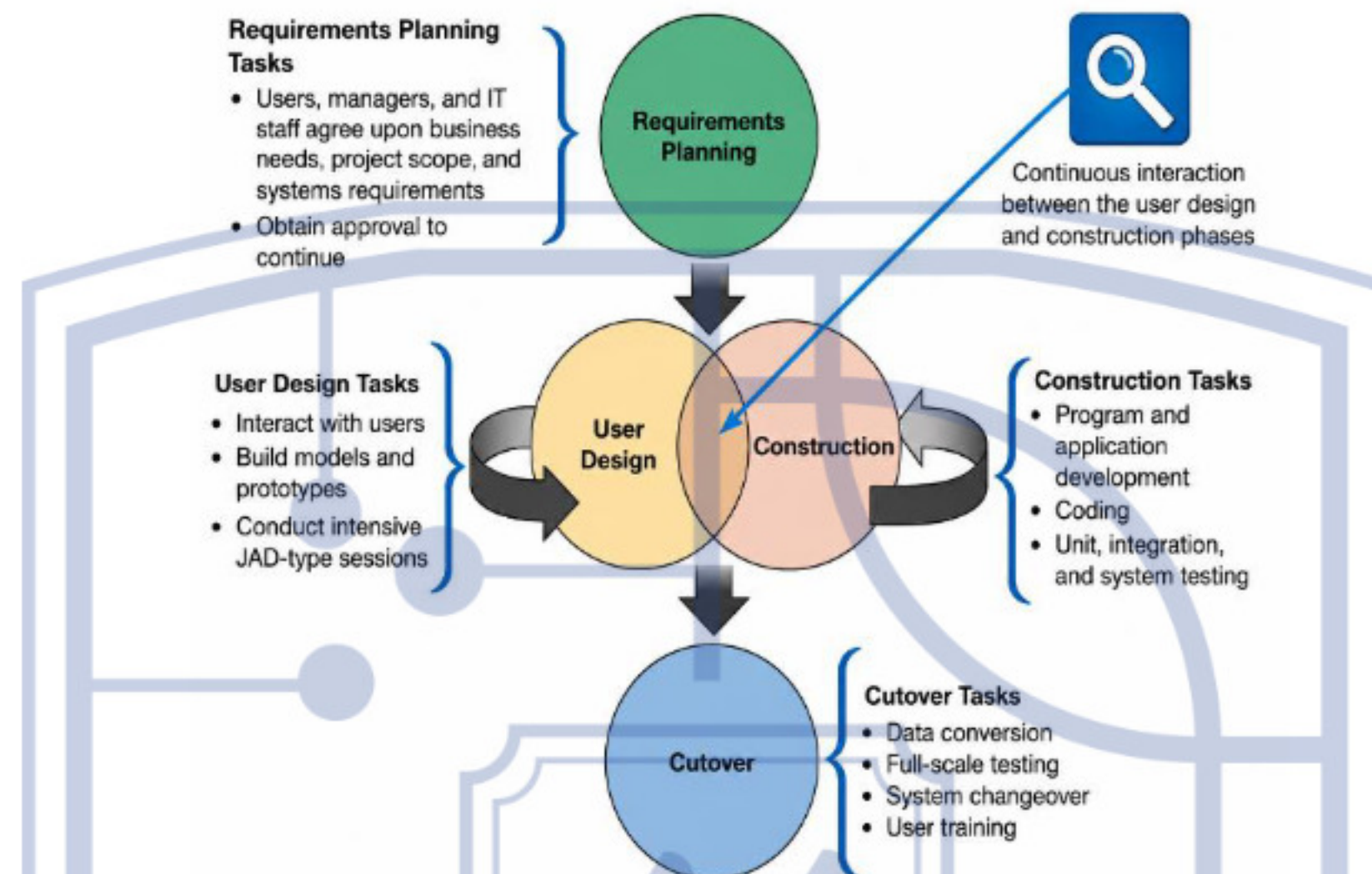
1. *Data Definition Language* (DDL), digunakan untuk mendefinisikan struktur basis data seperti pembuatan tabel, atribut, tipe data, dan relasi.
2. *Data Manipulation Language* (DML), digunakan untuk menambah, mengubah, menghapus, dan mengambil data dari basis data.
3. *Data Control*, digunakan untuk mengatur hak akses pengguna, menjaga keamanan data, serta memastikan integritas data tetap terjaga.

Salah satu DBMS yang banyak digunakan dalam pengembangan sistem informasi adalah *Microsoft SQL Server*. SQL Server merupakan DBMS relasional yang mendukung pengelolaan transaksi, pengelolaan data dalam jumlah besar, serta menyusun laporan secara terintegrasi. SQL Server juga mendukung penggunaan bahasa *Structured Query Server* (SQL) untuk melakukan manipulasi dan pengambilan data secara efisien, seperti menggunakan perintah *select*, *insert*, *update*, *delete*, *join*, dan *aggregate*. Penggunaan *query* yang optimal sangat penting untuk memastikan performa sistem tetap cepat dan akurat saat memproses data transaksi[9].

2.2 Rapid Application Development (RAD)

Rapid Application Development (RAD) merupakan salah satu model pengembangan perangkat lunak yang menekankan kecepatan proses pembangunan sistem, penggunaan

prototype, serta keterlibatan aktif pengguna dalam setiap tahapan pengembangan. Metode ini dirancang untuk menghasilkan perangkat lunak dalam waktu relatif singkat melalui siklus pengembangan yang bersifat iteratif dan *incremental*. Dalam konteks sistem informasi, RAD banyak digunakan karena mampu mempercepat proses implementasi sistem tanpa mengurangi kesesuaian sistem terhadap kebutuhan pengguna[10].



Gambar 2. 1 Model Rapid Application Development

Terdapat empat tahapan dalam RAD (*Rapid Application Development*) yang melibatkan perencanaan persyaratan, desain pengguna, dan konstruksi sistem. Ketiga tahapan tersebut melibatkan kolaborasi aktif antara analis sistem, pengembang, dan pengguna agar proses pengembangan berjalan cepat dan terarah. Berikut tahapan-tahapan yang terkait dengan masing-masing fase [11]:

1. Perencanaan Persyaratan (*Requirements Planning*)

Pada tahap ini, pengguna dan analis bekerja sama untuk merencanakan tujuan dari aplikasi atau sistem yang akan dikembangkan serta menentukan kebutuhan informasi yang harus dipenuhi. Proses ini fokus pada penyelesaian permasalahan yang dihadapi perusahaan dengan melakukan analisis fungsional dan non-fungsional. Fase awal ini juga melibatkan pendokumentasian yang mendalam mengenai kebutuhan operasional pengguna akhir.

2. Desain Pengguna (*User Design*)

Tahap ini fokus pada perancangan dan penyempurnaan sistem yang dilakukan dalam bentuk perencanaan prototipe dan menerima umpan balik. Analis dan pengembang bekerja sama untuk membangun serta menampilkan representasi visual dengan prototyping dari desain dan alur kerja kepada pengguna. Sifat berulang dari metode RAD pada fase ini memfasilitasi siklus umpan balik yang konstan antara

pengembang dan pengguna akhir. Durasi tahapan ini bervariasi tergantung pada kompleksitas aplikasi yang dikembangkan. Selama sesi ini, pengguna memberikan umpan balik terhadap prototipe yang telah dibuat, sementara analis menyesuaikan dan menyempurnakan modul berdasarkan masukan tersebut untuk memastikan prototipe sesuai dengan ekspektasi pengguna.

3. Konstruksi Sistem (*Construction*)

Pada tahap konstruksi sistem fokus pada aspek-aspek yang disepakati dan sistem dikembangkan serta dievaluasi. Fase ini melibatkan pengembangan aplikasi secara aktual dengan memanfaatkan umpan balik yang diterima selama fase desain pengguna untuk melakukan penyempurnaan secara berulang. Sistem baru atau bagian dari sistem yang telah disempurnakan akan diuji coba sebelum akhirnya diperkenalkan dan diterapkan dalam organisasi. Kemajuan yang terstruktur dalam fase ini menunjukkan efisiensi metode RAD dalam mengakomodasi kebutuhan proyek yang dinamis.

Jika dibandingkan dengan metode pengembangan lainnya, *Rapid Application Development* (RAD) memiliki sejumlah keunggulan yang menjadikannya relevan untuk digunakan dalam penelitian ini. Dibandingkan dengan model *Waterfall*, RAD menawarkan tingkat fleksibilitas yang lebih tinggi karena memungkinkan perubahan kebutuhan pengguna selama proses pengembangan berlangsung tanpa harus mengulang seluruh tahapan dari awal. Hal ini sejalan dengan penelitian yang menjelaskan bahwa model *Waterfall* cenderung bersifat linear, sedangkan RAD lebih adaptif karena menggunakan pendekatan interaktif dan *prototype* yang dapat disesuaikan secara cepat [12] tersebut, penggunaan metode RAD dalam penelitian ini dinilai sebagai pilihan yang tepat karena mampu mempercepat proses pengembangan aplikasi, memudahkan penyesuaian terhadap kebutuhan pengguna, serta menghasilkan sistem yang lebih sesuai dengan proses bisnis *laundry* yang berjalan.

2.3 Teknik Pengembangan Sistem

2.3.1 Model *Use Case*

Use case merupakan salah satu teknik pemodelan dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan interaksi antara aktor (pengguna atau sistem eksternal) dengan sistem yang akan dibangun. Model ini berfungsi untuk menjelaskan kebutuhan fungsional sistem berdasarkan sudut pandang pengguna, sehingga dapat membantu proses analisis kebutuhan secara lebih terstruktur. *Use case* dapat diartikan sebagai ringkasan dari sebuah sistem yang isinya adalah apa yang dilakukan *user* terhadap sistem, bukan apa yang sistem lakukan. Untuk menggambarkan suatu sistem yang sama bisa

jadi mengalami perbedaan pandangan antara satu orang dengan orang yang lainnya[8]. Adapun bentuk visual dari Model *Use Case* yaitu *Use Case Diagram*, digunakan untuk menggambarkan hubungan antara aktor dengan fungsi-fungsi yang tersedia dalam proses analisis dan pengembangan sistem, karena mampu membantu seluruh pihak untuk memahami mengenai kebutuhan serta fitur yang disediakan oleh sistem.

Simbol-simbol yang digunakan dalam *Use Case Diagram* sebagai berikut [13]:

Tabel 2. 1 Simbol Diagram *Use Case*

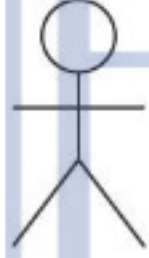
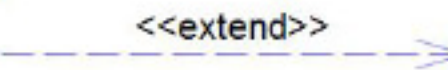
No.	Simbol	Nama	Keterangan
1		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
2		<i>Actor</i> (Aktor)	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi itu sendiri.
3		<i>Association</i> (Asosiasi)	Komunikasi antar aktor dan <i>Use Case</i> yang berpartisipasi pada <i>Use Case</i> atau <i>Use Case</i> memiliki interaksi dengan aktor.
4		<i>Generalization</i> (Generalisasi)	Hubungan generalisasi dan spesialisasi (umum- khusus) antar dua buah <i>Use Case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya.
5		<i>Include</i> (Termasuk)	Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dimana <i>Use Case</i> yang ditambahkan memerlukan <i>Use Case</i> ini untuk menjalankan fungsinya.
6		<i>Extends</i> (Perluasan)	Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> , dimana <i>Use Case</i> yang ditambahkan dapat berdiri sendiri.

Diagram *Use Case* tidak hanya berfungsi sebagai representasi visual yang menggambarkan hubungan antara aktor dan sistem, tetapi juga menjadi landasan dalam

penyusunan deskripsi atau narasi *Use Case*. Setelah hubungan antar elemen pada diagram dipahami, deskripsi *Use Case* digunakan untuk menjelaskan secara lebih rinci proses interaksi yang terjadi dalam operasional sistem. Melalui deskripsi tersebut, alur penggunaan sistem, skenario proses, serta peran dan tanggung jawab masing-masing aktor dapat dijelaskan secara sistematis. Deskripsi atau narasi *Use Case* merupakan bagian penting dalam pemodelan sistem karena memuat penjabaran detail mengenai mekanisme interaksi antara aktor dan sistem. Narasi ini membantu pengembang dalam memahami kebutuhan fungsional sistem secara lebih jelas, sehingga proses pengembangan sistem dapat dilakukan sesuai dengan tujuan dan kebutuhan pengguna.

Elemen-elemen yang digunakan untuk mendeskripsikan *use case* terdiri seperti berikut [14]:

No	Elemen	Keterangan	Contoh
1	Nama <i>Use Case</i>	Penamaan <i>Use Case</i> menggunakan frasa kata kerja yang ringkas dan mencerminkan tujuan fungsional yang ingin dicapai oleh aktor terhadap sistem.	Kelola Data Pengguna, Cetak Laporan Penjualan
2	Aktor	Pihak eksternal yang berinteraksi langsung dengan sistem dan memiliki peran dalam menginisiasi atau menerima hasil dari <i>Use Case</i> yang bersangkutan.	Administrator, Pengguna
3	Deskripsi Singkat	Ringkasan naratif yang menjelaskan tujuan utama dan ruang lingkup <i>Use Case</i> secara umum, tanpa merinci langkah-langkah teknis pelaksanaannya.	<i>Use case</i> ini memungkinkan kasir untuk memproses transaksi pembayaran dan mencetak struk pembelian kepada pelanggan.
4	Prasyarat (<i>Preconditions</i>)	Kondisi atau keadaan awal yang wajib terpenuhi terlebih dahulu sebelum <i>Use Case</i> dapat dimulai dan dijalankan oleh aktor.	Sistem telah aktif dan aktor telah berhasil terautentikasi sesuai hak aksesnya.

5	Langkah- langkah Utama (<i>Main Flow</i>)	Rangkaian langkah-langkah interaksi antara aktor dan sistem yang berlangsung secara berurutan pada kondisi ideal tanpa terjadi gangguan atau kesalahan.	1. Aktor memilih menu yang dituju. 2. Sistem menampilkan formulir isian. 3. Aktor melengkapi data dan mengkonfirmasi. 4. Sistem menyimpan dan menampilkan notifikasi berhasil.
6	Alur Alternatif (<i>Alternative Flow</i>)	Skenario penanganan yang dijalankan sistem ketika terjadi kondisi pengecualian, kesalahan <i>input</i> , atau situasi di luar alur normal yang telah ditetapkan.	Apabila data yang dimasukkan tidak valid, sistem memberikan peringatan dan mengarahkan aktor untuk memperbaiki isian sebelum melanjutkan proses.
7	<i>Post condition</i>	Keadaan sistem yang diharapkan setelah seluruh rangkaian <i>Use Case</i> selesai dieksekusi, baik melalui alur normal maupun alur alternatif.	Data tersimpan dengan baik pada basis data dan sistem kembali ke halaman utama yang siap menerima perintah selanjutnya.

2.3.2 PIECES

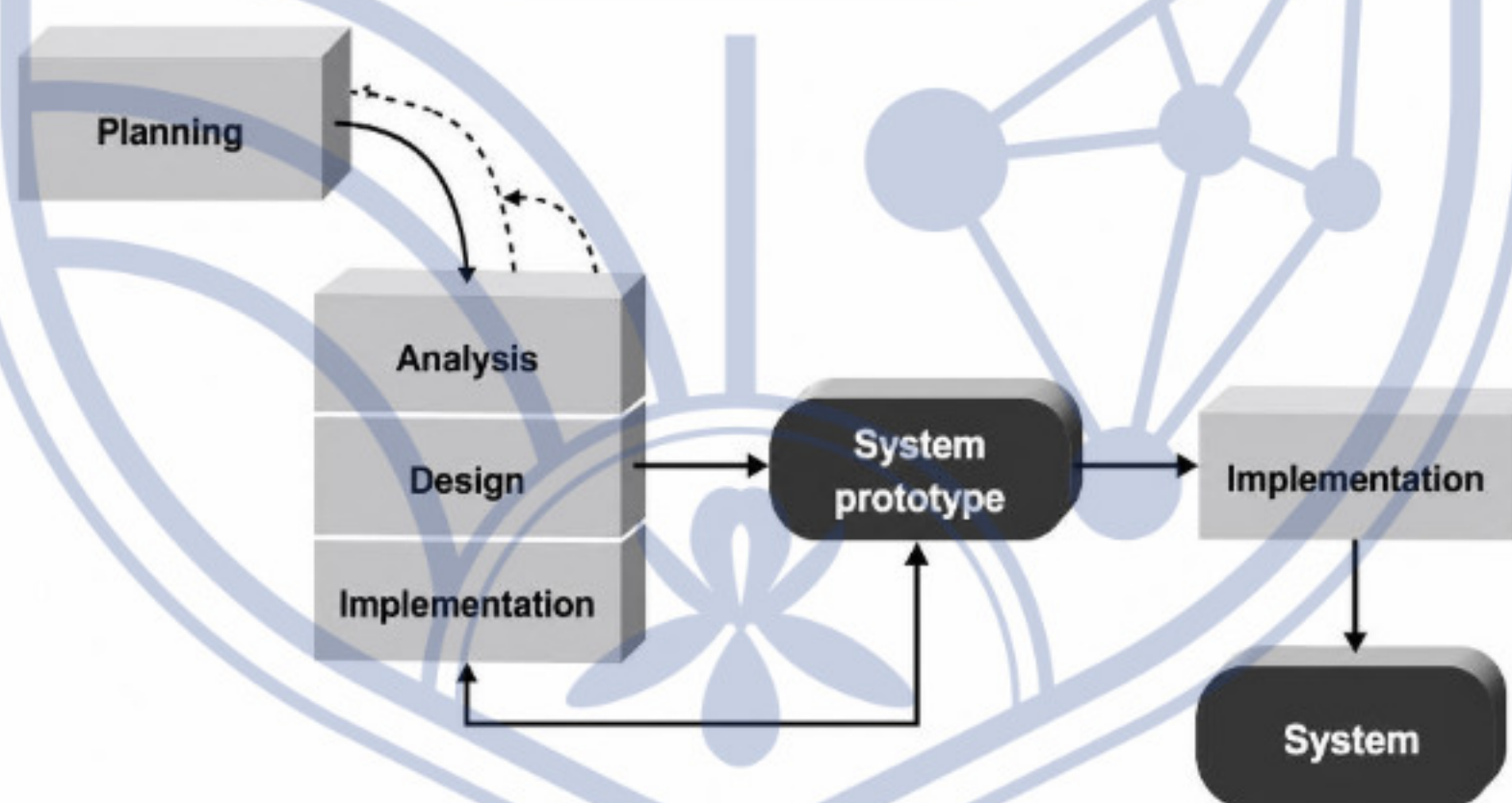
PIECES merupakan suatu kerangka kerja yang digunakan untuk menganalisis dan mengevaluasi sistem informasi. Analisis perancangan sistem ini digunakan untuk mengambil keputusan apabila sistem mempunyai masalah atau sudah tidak berfungsi secara baik, dan hasilnya digunakan sebagai dasar untuk memperbaiki sistem. Metode ini membantu dalam mengidentifikasi masalah, peluang, dan arahan (*directives*) dalam analisis definisi ruang lingkup (*scope definition analysis*) serta perancangan sistem. Metode ini memiliki enam variabel penilaian dalam melakukan evaluasi, yaitu [15]:

1. *Performance* (Kinerja): Mengukur seberapa cepat dan efisien sistem bekerja.
2. *Information* (Informasi): Menilai dari kualitas serta kejelasan informasi yang dihasilkan sistem.
3. *Economy* (Ekonomi): Menentukan kelayakan biaya dan manfaat finansial sistem.

4. *Control* (Pengendalian): Memeriksa mekanisme pengamanan dan pengawasan sistem.
5. *Efficiency* (Efisiensi): Mengevaluasi seberapa efektif sistem mencapai tujuannya dengan sumber daya yang ada.
6. *Service* (Layanan): Menilai kualitas dukungan dan pelayanan yang diberikan oleh sistem.

2.3.3 Prototyping

Prototyping merupakan metode dalam pengembangan sistem yang digunakan untuk membuat model awal (*prototype*) dari sistem yang akan dibangun, sehingga dapat diuji dan dievaluasi sebelum sistem final dikembangkan. Metode ini menekankan pada interaksi antar pengguna dan pengembang melalui proses yang bersifat berulang (*interaktif*), sehingga kebutuhan pengguna dapat dipahami secara lebih jelas. *Prototyping* sistem melibatkan proses analisis, desain, dan implementasi secara bersamaan untuk menghasilkan versi awal sistem. Versi awal ini dikembangkan secara sederhana sebagai representasi awal dari sistem yang akan dibangun, kemudian diuji langsung oleh pengguna untuk memperoleh umpan balik. Hasil evaluasi tersebut digunakan sebagai dasar dalam melakukan perbaikan dan penyempurnaan sistem secara berulang hingga kebutuhan pengguna dapat terpenuhi secara optimal[14].



Gambar 2. 2 Sistem *Prototyping*

Pendekatan *prototyping* sistem memiliki keunggulan dalam mempercepat proses pengembangan sistem karena pengguna dapat melakukan evaluasi mulai dari tahap awal. Selain itu, keterlibatan pengguna secara langsung dalam proses pengembangan dapat meningkatkan kesesuaian sistem dengan kebutuhan yang diharapkan. Namun, pendekat ini juga memiliki kelemahan antara lain kurangnya dokumentasi yang terstruktur serta potensi

keterbatasan desain akibat pemahaman kebutuhan yang belum sepenuhnya jelas pada tahap awal.

2.4 Black Box Testing

Black Box Testing atau yang dikenal juga sebagai pengujian berbasis spesifikasi fungsional, merupakan metode pengujian perangkat lunak yang fokus pada verifikasi fungsionalitas sistem berdasarkan perilaku masukan (*input*) dan keluaran (*output*). Metode ini memungkinkan penguji untuk menyalakan perangkat lunak tanpa perlu mengetahui, memeriksa, atau memahami struktur logika internal dan kode program dari sistem tersebut. Tujuan utama dari pengujian ini adalah untuk memastikan bahwa seluruh pengguna antarmuka dan sistem fungsionalitas berjalan sesuai dengan spesifikasi kebutuhan yang telah ditentukan pada tahap perancangan. Selain itu, metode ini sangat berguna untuk mengidentifikasi kesalahan dari perspektif pengguna akhir (*end-user*), di mana penguji berperan murni sebagai pengguna yang berinteraksi langsung dengan antarmuka aplikasi[16]. Secara umum, proses evaluasi dalam pengujian Black Box difokuskan pada upaya untuk menemukan beberapa kategori kesalahan pada sistem, antara lain:

1. Fungsi-fungsi yang tidak benar atau hilang sehingga sistem tidak berjalan sebagaimana mestinya.
2. Kesalahan pada antarmuka pengguna (*User Interface*).
3. Kesalahan pada struktur data atau proses akses ke basis data eksternal.
4. Kesalahan kinerja saat sistem menerima masukan operasional.
5. Kesalahan inisialisasi dan penguatan pada aplikasi.

Untuk merancang skenario pengujian (*test case*) yang terstruktur dan efektif, metode Black Box Testing memiliki beberapa pendekatan teknik pelaksanaan. Dalam pengembangan sistem informasi yang memproses kelancaran operasional transaksi dan pengelolaan data dasar, kombinasi teknik yang paling umum dan relevan digunakan meliputi[17]:

1. Partisi Ekuivalensi (*Equivalence Partitioning*)

Teknik ini membagi domain atau jarak masukan data menjadi dua bagian utama, yaitu kelompok data yang valid dan data yang tidak valid. Pengelompokan ini didasarkan pada evaluasi ekuivalensi (kesetaraan) dari suatu tipe atau kelas kondisi masukan, seperti nilai numerik, rentang nilai, atau kondisi boolean. Penguji kemudian merancang skenario pengujian dengan mengambil nilai perwakilan dari setiap kelompok. Teknik ini sangat berguna untuk memverifikasi apakah sistem keluaran telah sesuai dengan skenario uji,

mengurangi sekaligus jumlah pengujian yang harus dilakukan secara efisien tanpa mengurangi efektivitas hasil uji.

2. Analisis Nilai Batas (*Boundary Value Analysis*)

Teknik *Boundary Value Analysis* (BVA) merupakan pelengkap sekaligus perluasan dari teknik Equivalence Partitioning. Pengujian ini secara spesifik dikhususkan pada batas-batas ekstrem dari suatu rentang data masukan, di mana nilai-nilai ekstrem seperti batas minimum dan maksimum yang dipilih untuk diuji. BVA dinilai sangat efektif untuk mengidentifikasi potensi masalah yang sering kali luput, karena kerentanan atau kesalahan logika (*bug*) pada sistem umumnya terjadi pada batasan nilai input. Melalui teknik ini, sistem diuji untuk memastikan respon dan validasi yang tepat terhadap nilai masukan batas bawah, batas atas, hingga nilai yang berada di luar batas yang.

Pendekatan *Black Box Testing* melalui kombinasi teknik-teknik di atas sangat tepat diterapkan untuk mengukur kualitas fungsionalitas sistem dari sudut pandang interaksi pengguna akhir. Validitas pengujian dijamin melalui pemetaan yang konsisten antara spesifikasi perancangan dengan keluaran yang dihasilkan; mencakup pemeriksaan respon antarmuka, validasi masukan data operasional, keakuratan kalkulasi transaksi, hingga integritas data yang tersimpan pada basis data. Dengan demikian, metode ini memastikan perangkat lunak yang dikembangkan benar-benar andal, meminimalkan kendala operasional, dan siap diimplementasikan secara langsung.

2.5 Penjualan

Penjualan adalah kegiatan perusahaan dalam memindahkan hak kepemilikan barang atau jasa kepada pelanggan melalui transaksi tunai maupun kredit. Dalam proses bisnis modern, penjualan tidak hanya mencakup transaksi barang keluar, tetapi juga pengelolaan data pelanggan, pembuatan nota, retur penjualan, serta penyusunan laporan penjualan. Sistem informasi penjualan membantu perusahaan memproses transaksi secara cepat, meminimalkan kesalahan perhitungan, dan menghasilkan informasi penjualan yang akurat untuk mendukung pengambilan keputusan[18]. Dalam transaksi penjualan terdapat dua jenis transaksi penjualan, yaitu [19]:

1. Penjualan Tunai

Penjualan tunai merupakan transaksi penjualan yang mengharuskan pembeli melakukan pembayaran secara penuh terlebih dahulu sebelum barang atau jasa diserahkan oleh penjual. Setelah pembayaran diterima, barang atau jasa kemudian diberikan kepada pembeli dan transaksi tersebut dicatat sebagai penjualan oleh bagian

yang berwenang. Sistem ini menekankan adanya penerimaan kas secara langsung pada saat transaksi berlangsung sehingga perusahaan dapat segera memperoleh dana dari hasil penjualan.

2. Penjualan Kredit (cicilan)

Penjualan kredit adalah transaksi penjualan di mana barang atau jasa diserahkan terlebih dahulu kepada pembeli, sementara pembayarannya dilakukan dalam jangka waktu tertentu sesuai kesepakatan yang telah ditetapkan. Dalam mekanisme ini, penjual memiliki hak tagih kepada pembeli yang dicatat sebagai piutang sampai pembayaran dilunasi. Penjualan kredit umumnya dilakukan untuk memberikan kemudahan kepada pelanggan melalui sistem pembayaran bertahap atau angsuran, sehingga dapat meningkatkan daya beli serta memperluas peluang penjualan perusahaan.

2.5.1 Penjualan Jasa

Penjualan jasa merupakan aktivitas perusahaan dalam menawarkan layanan kepada pelanggan yang tidak menghasilkan produk berwujud, melainkan berupa kinerja, keahlian, atau manfaat tertentu yang dapat dirasakan oleh pelanggan. Dalam proses ini, nilai utama yang diberikan terletak pada kualitas pelayanan, kepuasan, serta pengalaman yang diterima pelanggan setelah jasa tersebut digunakan.

Karakteristik utama penjualan jasa meliputi sifat tidak berwujud (*intangible*), tidak dapat disimpan (*perishable*), serta proses produksi dan konsumsi yang terjadi secara bersamaan (*inseparability*). Oleh karena itu, pengelolaan penjualan jasa memerlukan sistem yang mampu mengelola data pelanggan, transaksi layanan, serta laporan secara akurat agar dapat meningkatkan efisiensi dan kualitas pelayanan[20].

2.5.2 Penjualan Produk

Penjualan produk adalah kegiatan perusahaan dalam menjual barang berwujud (*tangible*) kepada pelanggan, yang ditandai dengan adanya perpindahan hak kepemilikan barang setelah transaksi dilakukan. Produk yang dijual dapat berupa barang jadi maupun setengah jadi yang memiliki nilai ekonomi. Dalam sistem informasi penjualan produk, proses yang dilakukan meliputi pengelolaan data barang, pengendalian persediaan (*inventory*), pencatatan transaksi penjualan, hingga penyusunan laporan penjualan[21]. Sistem ini berperan penting dalam meningkatkan ketepatan pencatatan, mengurangi kesalahan manusia, serta mendukung pengambilan keputusan manajemen.

2.5.3 Loyalitas Pelanggan

Loyalitas pelanggan merupakan komitmen pelanggan untuk tetap menggunakan produk atau jasa dari suatu perusahaan secara berulang dalam jangka waktu yang panjang. Pelanggan yang loyal tidak hanya melakukan pembelian ulang, tetapi juga cenderung merekomendasikan layanan kepada orang lain serta memiliki tingkat kepuasan yang lebih tinggi terhadap perusahaan.

Dalam dunia usaha, salah satu strategi yang umum diterapkan untuk meningkatkan loyalitas pelanggan adalah melalui program *membership*. Program *membership* memberikan berbagai keuntungan kepada pelanggan, seperti potongan harga, pemberian poin, promo khusus, ataupun fasilitas pelayanan tertentu. Penerapan program *membership* bertujuan mendorong pelanggan melakukan transaksi secara berulang sehingga dapat meningkatkan loyalitas pelanggan[22].

Pada usaha laundry, penerapan sistem *membership* dapat berupa pemberian diskon khusus kepada pelanggan yang telah terdaftar sebagai anggota. Integrasi fitur *membership* dalam sistem informasi membantu pengelolaan data pelanggan, pencatatan status keanggotaan, serta penerapan diskon secara otomatis pada saat transaksi sehingga pelayanan menjadi lebih konsisten dan efisien. Penerapan sistem seperti ini juga dapat mendukung peningkatan loyalitas pelanggan melalui kemudahan pengelolaan data pelanggan dan pemberian manfaat yang mendorong pelanggan untuk terus menggunakan layanan yang sama.

2.6 Pembelian

Pembelian merupakan suatu kegiatan yang dilakukan perusahaan untuk memperoleh barang atau bahan yang dibutuhkan dalam menjalankan operasionalnya, baik berupa bahan operasional, barang dagangan, maupun perlengkapan lainnya. Proses pembelian melibatkan interaksi dengan pihak luar, yaitu pemasok (*supplier*), serta mencakup serangkaian aktivitas mulai dari permintaan barang hingga pembayaran atas barang yang telah diterima. Dalam sistem informasi akuntansi, pembelian tidak hanya dipandang sebagai transaksi pertukaran antara uang dengan barang, tetapi juga sebagai suatu siklus yang terstruktur dan terkoordinasi. Siklus ini meliputi kegiatan seperti pengajuan permintaan pembelian, pemilihan pemasok, pembuatan pesanan (*purchase order*), penerimaan barang, hingga pencatatan dan pembayaran kewajiban kepada pemasok[23].

Fungsi-fungsi yang terkait dalam sistem informasi akuntansi pembelian adalah sebagai berikut[24]:

2. Fungsi Gudang

Fungsi gudang memiliki tanggung jawab dalam membuat dan mengajukan surat permintaan pembelian jika persediaan di gudang mencapai jumlah minimum. Permintaan pembelian yang telah diajukan harus sesuai dengan posisi persediaan yang ada di gudang, untuk memudahkan penyimpanan oleh fungsi penyimpanan.

3. Fungsi Pembelian

Fungsi pembelian memiliki tanggung jawab untuk mendapatkan informasi terbaru mengenai harga barang dari pemasok yang dipilih. Setelah mendapatkan pemasok sesuai dengan kriteria dan harga barang yang diinginkan, fungsi pembelian akan membuat surat order pembelian yang kemudian dikirim ke pemasok yang telah dipilih.

4. Fungsi Penerimaan

Fungsi penerimaan barang dagang memiliki wewenang dalam menerima barang dan melakukan pemeriksaan terhadap barang-barang yang telah diterima dari pemasok seperti jenis, jumlah maupun kualitas barang. Guna untuk menentukan apakah barang yang tersebut sudah sesuai dengan formulir surat order pembelian yang telah diajukan.

5. Fungsi Akuntansi

Pada sistem akuntansi pembelian, fungsi akuntansi bertanggung jawab dalam fungsi pencatatan persediaan dan fungsi pencatatan utang. Fungsi pencatatan persediaan bertanggung jawab dalam mencatat harga pokok persediaan ke dalam kartu utang sedangkan fungsi pencatatan utang bertanggung jawab dalam mencatat transaksi pembelian ke dalam register kas keluar kemudian mengarsip dokumen bukti kas keluar lalu di catat ke dalam buku pembantu utang.

2.7 Persediaan

Persediaan merupakan aset yang dimiliki perusahaan untuk dijual dalam aktivitas operasional sehari-hari atau digunakan dalam proses produksi guna menghasilkan barang yang siap dipasarkan. Persediaan meliputi bahan habis pakai, barang dalam proses, serta barang jadi. Oleh karena itu, pengelolaan persediaan perlu dilakukan secara efektif agar dapat mencegah terjadinya kekurangan maupun kelebihan stok yang berpotensi mengganggu kelancaran operasional perusahaan. Dalam hal ini, sistem informasi akuntansi memiliki peran penting dalam mengelola persediaan sehingga informasi yang dihasilkan lebih akurat dan dapat digunakan sebagai dasar dalam pengambilan keputusan manajemen[23].

Secara umum, terdapat dua jenis sistem agar pencatatan persediaan tetap akurat, seperti yang di bawah ini [25]:

1. Sistem periodik adalah metode pencatatan persediaan di mana pencatatan mutasi persediaan tidak dilakukan secara terus-menerus, melainkan pada periode tertentu. Dalam sistem ini, nilai persediaan akhir hanya dapat diketahui setelah dilakukan perhitungan fisik (*stock opname*) pada akhir periode.

Contoh perhitungan pada sistem ini, pemakaian harian yang terjadi selama bulan Mei tidak dicatat pada saat itu. Perhitungan sebagai berikut :

- a. Persediaan Awal (1 Mei): Rp200.000 (10 botol)
- b. Pembelian (5 Mei): Rp400.000 (20 botol)
- c. Barang Tersedia Digunakan: $\text{Rp}200.000 + \text{Rp}400.000 = \text{Rp}600.000$ (30 botol)
- d. Nilai Persediaan Akhir (31 Mei): Setelah dihitung secara fisik di gudang, ternyata tersisa 15 botol. Nilainya adalah $15 \text{ botol} \times \text{Rp}20.000 = \text{Rp}300.000$.

Perhitungan Beban Pemakaian (Akhir Periode):

- a. Barang Tersedia Digunakan – Persediaan Akhir Fisik
- b. $\text{Rp}600.000 - \text{Rp}300.000 = \text{Rp}300.000$

Kesimpulan dari sistem periodik nilai pemakaian deterjen sebesar Rp300.000 dan sisa stok 15 botol baru bisa diketahui secara pasti pada tanggal 31 Mei setelah menghitung sisa stok fisik pada akhir periode.

2. Sistem perpetual adalah metode pencatatan persediaan di mana setiap transaksi yang berkaitan dengan persediaan dicatat secara langsung dan terus-menerus, sehingga jumlah persediaan dapat diketahui setiap saat tanpa harus menunggu akhir periode.

Contoh perhitungan pada sistem ini, setiap kali ada barang masuk (dibeli) atau barang keluar (dipakai/dijual), akan langsung dilakukan pembaruan stok. Perhitungan sebagai berikut :

- a. Saldo Awal(1 Mei): Sistem mencatat stok 10 botol senilai Rp200.000.
- b. Pembelian(5 Mei): Saat data pembelian dimasukkan ke sistem, saldo langsung bertambah otomatis.
- c. Saldo di sistem: $10 \text{ botol} + 20 \text{ botol} = 30 \text{ botol}$ (Rp600.000).
- d. Pemakaian(10 Mei s/d 28 Mei): Setiap petugas mengambil deterjen (total 15 botol), sistem mencatatnya sebagai pengurang stok senilai Rp300.000 ($15 \times \text{Rp}20.000$).
- e. Saldo di sistem bergerak seketika: $30 \text{ botol} - 15 \text{ botol} = 15 \text{ botol}$.

Kesimpulan pada sistem perpetual tidak perlu menunggu akhir bulan kapan pun pemilik usaha membuka aplikasi, sistem sudah menunjukkan sisa persediaan aktual adalah 15 botol senilai Rp300.000.

