BAB II
KAJIAN LITERATUR

2.1 Konsep Sistem Informasi

2.1.1 Sistem

Berbagai definisi tentang sistem yang dikutip dari beberapa sumber yang memberikan penjelasan tentang sistem [5].

1. Sistem adalah suatu jaringan prosedur yang dibuat menurut pola yang terpadu untuk melaksanakan kegiatan pokok perusahaan.
2. Secara sederhana suatu sistem dapat di artikan suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu.

Ada 3 elemen yang membentuk sebuah sistem yaitu[5]:

1. *Input*

   Segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan untuk diproses.
2. Proses

   Bagian yang melakukan perubahan dari *input* menjadi *output* yang berguna, misalnya berupda informasi dan produk, tetapi juga bisa berupa hal-hal yang tidak berguna, misalnya sisa pembuangan atau limbah.
3. *Output*

   Hasil dari pemrosesan, misalnya berupa suatu informasi, saran, cetakan laporan, dan lain-lain.

Berdasarkan pengertian diatas dapat disimpulkan sistem adalah cara yang dilakukan untuk mencapai tujuan yang telah dibuat mulai dari meng-*input* sesuatu memprosesnya kemudian menghasilkan *output* [5].

Model umum sebuah sistem adalah *input*, proses, dan *output*. Hal ini merupakan konsep sebuah sistem yang sangat sederhana sebab sebuah sistem dapat mempunyai beberapa masukan dan keluaran. Selain itu, sebuah sistem mempunyai karakteristik atau sifat-sifat tertentu yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. Adapun karakeristik yang dimaksud adalah sebagai berikut [6]:

1. Komponen Sistem (*Components*)

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem, Setiap subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar atau sering disebut "supra sistem".
2. Batasan Sistem (*Boundary*)

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan.

3. Lingkungan Luar Sistem (*Environtment*)

   Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut lingkungan luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Dengan demikian, lingkungan luar tersebut harus tetap dijaga dan dipelihara, Lingkungan luar yang merugikan harus dikendalikan. Karena apabila tidak dilakukan, hal tersebut berpotensi mengganggu kelangsungan hidupsistem tersebut.

4. Penghubung Sistem (*interface*)

   Media yang menghubungkan sistem dengan subsistem lain disebut penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lain. Bentuk keluaran dari satu subsistem akan menjadi masukan untuk subsistem lain melalui penghubung tersebut. Dengan demikian, dapat terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan Sistem (*Input*)

   Energi yang dimasukkan ke dalam sistem disebut masukkan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Contoh, di dalam suatu unit sistem komputer, "program" adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan "data" adalah signal *input* untuk diolah menjadi infomasi.

6. Keluaran Sistem (*Output*)

   Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi sub sistem yang lain seperti sistem informasi. Keluaran yang dihasilkan adalah informasi. Informasi ini dapat digunakan sebagai masukkan untuk pengambilan keputusan atau hal-hal lain yang menjadi *input* bagi sub sistem lain.

7. Pengolah Sistem (Proses)

   Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran, contohnya adalah sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

8. Sasaran Sistem (*Objective*)

   Suatu sistem memiliki tujuan dan sasaran yang pasti. Apabila sistem tidak memiliki sasaran maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

2.1.2 Informasi

Berbagai definisi tentang informasi yang dikutip dari beberapa sumber yang memberikan penjelasan lebih lanjut tentang informasi [5].

1. Informasi adalah data yang telah diorganisasi dan telah memiliki kegunaan dan manfaat.
2. Pengertian teknologi informasi dan komunikasi informasi adalah segala sesuatu yang mempunyai arti dan nilai bagi penerima informasi.

Sumber informasi adalah data. Data merupakan kenayataan yang menggambarkan suatu kejadian serta merupakan suatu bentuk yang masih mentah yang belum dapat bercerita banyak sehingga perlu diolah lebih lanjur melalui suatu model untung menghasilkan informasi [6].

2.1.3 Sistem Informasi

Berbagai definisi tentang sistem informasi yang dikutip dari beberapa sumber yang memberikan penjelasan lebih lanjut tentang sistem informasi [5].

1. Sistem informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi, dan didistribusikan kepada pemakai.
2. Sistem informasi yaitu suatu sistem yang menyediakan informasi untuk manajemen dalam mengambil keputusan dan juga untuk menjalankan operasional perusahaan, di mana sistem tersebut merupakan kombinasi dari orang-orang, teknologi informasi dan prosedur-prosedur yang tergorganisasi. Biasanya suatu perusahan atau badan usaha menyediakan semacam informasi yang berguna bagi manajemen.
3. Sistem informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi, dan didistribusikan kepada pemakai.

Fungsi sistem informasi sebagai berikut [7]:

1. Untuk meningkatkan aksesiblitas data yang ada secara efektif dan efisien kepada pengguna, tanpa dengan perantara sistem informasi.
2. Memperbaiki produktivitas aplikasi pengembangan dan pemeliharaan sistem.
3. Menjamin tersedianya kualitas dan keterampilan dalam memanfaatkan sistem informasi secara kritis.
4. Mengidentifikasi kebutuhan mengenai keterampilan pendukung sistem informasi.
5. Mengantisipasi dan memahami akan konsekuensi ekonomi.
6. Menetapkan investasi yang akan diarahkan pada sistem informasi.
7. Mengembangkan proses perencanaan yang efektif.

Komponen-komponen dari sistem informasi adalah sebagai berikut [7]:

1. Komponen *input*, adalah data yang masuk ke dalam sistem informasi

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

2. Komponen model, adalah kombinasi prosedur, logika dan model matematika yang memproses data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.
3. Komponen *output*, adalah hasil informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.
4. Komponen teknologi, adalah alat dalam sistem informasi, teknologi digunakan dalam menerima *input* menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan *output* dan memantau pengendalian sistem.
5. Komponen basis data, adalah kumpulan data yang saling berhubungan yang tersimpan di dalam computer dengan menggunakan *software database*.
6. Komponen kontrol, adalah komponen yang mengendalikan gangguan terhadap sistem informasi.

Ciri-ciri sistem informasi [7]:

1. Baru, adalah informasi yang didapat sama sekali barudan segar bagi penerima
2. Tambahan, adalah informasi dapat diperbarui atau memberikan tambahan terhadap informasi yang sebelumnya telah ada.
3. Kolektif, adalah informasi yang dapat menjadi suatu koreksi dari informasi yang salah sebelumnya.
4. Penegas, adalah informasi yang dapat mempertegas informasi yang telah ada

2.1.3.1 Sistem Informasi Pemasaran

Sistem Informasi Pemasaran adalah alat yang memposting informasi yang terkait dengan catatan keputusan pemasaran, analisis dan keputusan distribusi. SIP membantu bisnis memahami pasar, perilaku konsumen, dan tren industri. Menggunakan teknologi informasi memungkinkan bisnis untuk mengoptimalkan strategi pemasaran mereka dan meningkatkan efektivitas kampanye mereka [8].

2.1.3.2 Sistem Informasi Penjualan

Sistem Informasi Penjualan digunakan untuk mendukung proses penjualan dengan menyediakan data untuk mengelola hubungan pelanggan dan meningkatkan efisiensi penjualan. SIPA memungkinkan untuk mengejar kegiatan penjualan, menganalisis kinerja tim penjualan, dan mengidentifikasi peluang pasar baru [9].

2.2 Hubungan antara *Fashion* dan Tas

2.2.1 Tas

Tas adalah wadah yang digunakan untuk membawa barang-barang, yang telah menjadi bagian integral dari kehidupan manusia selama berabad-abad. Selain fungsi utamanya sebagai alat untuk menyimpan

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

dan membawa barang, tas juga berkembang menjadi aksesori *fashion* yang mencerminkan kepribadian dan gaya hidup seseorang. Dalam industri *fashion*, tas telah berkembang menjadi lebih dari sekadar aksesori fungsional, menjadi simbol status, tren, dan ekspresi diri [10].

2.2.2 *Fashion*

Fashion, atau mode, merujuk pada gaya berpakaian, aksesori, musik, seni, dan elemen budaya lainnya yang populer pada waktu tertentu. Secara etimologis, kata "*fashion*" berasal dari bahasa Latin "*factio*", yang berarti "melakukan". Dalam perkembangannya, istilah ini diserap ke dalam bahasa Inggris menjadi "*fashion*", yang secara sederhana diartikan sebagai gaya pakaian yang populer dalam suatu budaya. *Fashion* tidak hanya mencerminkan tren estetika, tetapi juga berfungsi sebagai alat komunikasi identitas sosial. Pilihan *fashion* seseorang dapat mencerminkan status sosial, kepribadian, dan afiliasi kelompok tertentu. Sebagai contoh, penelitian menunjukkan bahwa *fashion* dan gaya hidup berperan sebagai identitas dan komunikasi dalam masyarakat modern [11].

2.2.3 Tas *Fashion*

Tas memainkan peran penting dalam industri *fashion* sebagai aksesori yang tidak hanya fungsional tetapi juga estetis. Desain, bahan, dan warna tas sering kali mengikuti tren *fashion* yang berkembang, menjadikannya komponen vital dalam industri *fashion*. Pengembangan desain tas yang inovatif dapat meningkatkan daya tarik dan nilai jual produk.

Selain itu, tas juga dapat berfungsi sebagai media ekspresi seni dan budaya. Misalnya, penerapan teknik *ecoprint* pada media kulit dalam pembuatan tas *fashion* menunjukkan bagaimana tas dapat menjadi kanvas untuk menampilkan karya seni yang unik dan bernilai estetika tinggi. Dengan demikian, tas tidak hanya berfungsi sebagai alat untuk membawa barang, tetapi juga sebagai elemen penting dalam *fashion* yang mencerminkan identitas, status sosial, dan ekspresi diri seseorang [12].

2.3 Jenis-Jenis Cat untuk Melukis pada Tas

Pemilihan jenis cat yang tepat sangat penting dalam proses melukis pada tas, karena mempengaruhi hasil akhir, daya tahan, dan estetika produk. Berikut adalah beberapa jenis cat yang umum digunakan:

1. Cat Air

Cat air memiliki karakteristik transparan, yang memungkinkan lapisan warna di bawahnya tetap terlihat, sehingga menghasilkan efek yang lembut dan *natural* pada lukisan. Biasanya, cat air digunakan pada media kertas berkualitas tinggi, seperti kertas *aquarel*, yang mampu menyerap pigmen dengan baik. Teknik melukis dengan cat air sering dimanfaatkan untuk menggambarkan pemandangan dan ilustrasi yang membutuhkan

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

efek transparansi dan gradasi warna yang halus. Selain mudah digunakan dan dibersihkan, cat air memiliki daya tarik karena memberikan nuansa warna yang lembut dan alami. Namun, penggunaan cat air memiliki beberapa kelemahan, terutama jika diaplikasikan pada media seperti kain atau kulit. Daya lekatnya yang rendah membuatnya kurang cocok untuk permukaan selain kertas. Selain itu, hasil lukisan dengan cat air rentan terhadap air setelah kering, sehingga membutuhkan perlindungan tambahan untuk menjaga keawetannya [13].

2. Cat Minyak

   Cat minyak memiliki karakteristik kekentalan yang tinggi, menghasilkan warna yang kaya dan tajam pada lukisan. Media yang paling sering digunakan untuk cat minyak adalah kanvas atau permukaan lain yang telah dipersiapkan khusus, seperti papan dengan dasar *gesso*. Salah satu keunggulan cat minyak adalah waktu pengeringannya yang lama, memungkinkan seniman untuk bekerja lebih detail dan melakukan *blending* warna secara maksimal. Namun, waktu pengeringan yang lama ini juga bisa menjadi kendala, terutama dalam proyek yang memerlukan hasil cepat. Selain itu, cat minyak mengandung bahan kimia seperti terpenting yang memerlukan penanganan dan ventilasi khusus saat digunakan [14].

3. Cat Akrilik

   Cat akrilik memiliki fleksibilitas yang tinggi karena dapat digunakan pada berbagai media seperti kertas, kanvas, kain, kayu, dan bahkan kulit. Kecepatan pengeringan cat akrilik menjadi salah satu kelebihannya, memungkinkan seniman untuk mempercepat proses kerja dan menambahkan lapisan tanpa harus menunggu terlalu lama. Setelah kering, cat akrilik memiliki daya tahan yang baik karena bersifat tahan air dan tidak mudah luntur, menjadikannya pilihan yang populer untuk melukis pada media yang rentan terkena air, seperti tas kanvas. Kelebihan lain dari cat akrilik adalah kemudahan penggunaannya, karena dapat dibersihkan dengan air sebelum mengering dan tidak memerlukan persiapan khusus pada media. Namun, karena sifatnya yang cepat kering, diperlukan teknik khusus untuk melakukan blending warna dengan baik. Selain itu, cat akrilik yang sudah kering sulit untuk dihapus atau diperbaiki, sehingga membutuhkan perencanaan yang matang sebelum diaplikasikan [15].

   Cat akrilik menjadi pilihan utama dalam melukis tas *fashion* karena kemampuannya menempel dengan baik pada berbagai jenis bahan tas, seperti kanvas dan kulit. Selain itu, daya tahannya terhadap air dan sinar matahari membuatnya ideal untuk produk *fashion* yang sering digunakan di luar ruangan. Proses pengeringannya yang cepat juga memungkinkan produksi dalam skala yang lebih besar dengan waktu yang efisien.

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

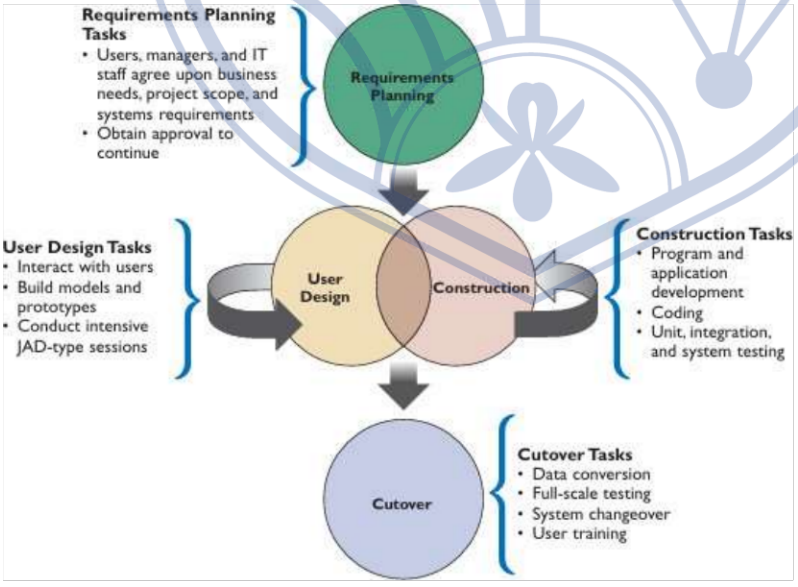
2.4 Teknik Melukis pada Tas

Teknik melukis pada tas merupakan salah satu bentuk penerapan seni lukis pada media alternatif, yaitu kain atau bahan sintetis yang digunakan dalam produk *fashion*. Proses ini memerlukan pemahaman mengenai karakteristik media, jenis cat, serta teknik pewarnaan agar menghasilkan karya yang tidak hanya estetis, tetapi juga tahan lama terhadap penggunaan sehari-hari [16].

Salah satu teknik yang umum digunakan adalah freehand *painting*, yaitu teknik melukis langsung tanpa pola atau cetakan. Teknik ini memberikan kebebasan berekspresi yang tinggi, tetapi membutuhkan keterampilan kontrol kuas yang baik. Selain itu, teknik sketsa awal menggunakan pensil atau kapur kain juga diterapkan untuk menjaga akurasi bentuk dan komposisi sebelum proses pewarnaan. Dalam pewarnaan, pelukis tas umumnya menggunakan cat akrilik khusus tekstil karena sifatnya yang fleksibel, tahan air, dan memiliki daya lekat yang kuat terhadap serat kain. Teknik pelapisan akhir atau *finishing* juga dilakukan, biasanya dengan *varnish* atau *sealant*, untuk melindungi hasil lukisan dari goresan maupun kelembaban. Dengan penerapan teknik yang tepat serta pemilihan bahan yang sesuai, lukisan pada tas tidak hanya menjadi nilai tambah dari segi estetika, tetapi juga meningkatkan nilai jual dan daya tahan produk dalam jangka panjang [16].

2.5 *Rapid Application Development* (RAD)

Rapid Application Development (RAD) adalah model proses pengembangan perangkat lunak yang bersifat *incremental* terutama untuk waktu pengerjaan yang pendek [17]. Tahapan RAD dapat dilihat pada gambar 2.1.



© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

Tahapan dari model RAD dapat dijabarkan sebagai berikut [18]:

1. Perencanaan Kebutuhan (*Requirements Planning*)

   Fase perencanaan kebutuhan menggabungkan elemen dari fase perencanaan sistem dan fase analisis sistem dari SDLC. *User*, manajer dan anggota staf IT berdiskusi dan menyetujui kebutuhan bisnis, ruang lingkup proyek, batasan dan kebutuhan sistem. Fase perencanaan kebutuhan ini berakhir ketika tim setuju pada hal-hal penting dan memperoleh otorisasi manajemen untuk lanjut ke tahapan berikutnya.

2. Perancangan Pemakai (*User Design*)

   Selama fase perancangan pemakai, *user* berinteraksi dengan analis sistem dan mengembangkan model dan *prototype* yang merepresentasikan semua proses, *output* dan *input* sistem. Perancangan pemakai adalah sebuah proses yang berkesinambungan dan interaktif yang memungkinkan *user* untuk memahami, memodifikasi dan menyetujui model kerja dari sistem yang sesuai dengan kebutuhan *user*.

Gambar 2.1 Empat Fase dari Model RAD

3. Konstruksi (*Construction*)

   Fase konstruksi fokus pada tugas pengembangan program dan aplikasi sama seperti pada SDLC. Namun, pada RAD, *user* terus berpartisipasi dan tetap dapat memberikan perubahan atau pengembangan terhadap tampilan atau laporan yang dibuat.

4. Pemutusan (*Cutover*)

   Fase pemutusan menggabungkan tugas terakhir dari fase implementasi pada SDLC, mencakup konversi data, pengujian, pengubahan ke sistem baru, dan pelatihan pemakai. Dibandingkan dengan metode tradisional, keseluruhan proses telah dikompresi, sehingga sistem baru dibuat dan diterapkan dengan waktu lebih cepat.

Adapun beberapa kelebihan dari metode RAD adalah [18]:

1. Lebih efektif dalam menghasilkan sistem yang memenuhi kebutuhan langsung dari pelanggan.
2. Cocok untuk proyek yang memerlukan waktu yang singkat.
3. Model RAD mengikuti tahap pengembangan sistem seperti pada umumnya, tetapi mempunyai kemampuan untuk menggunakan kembali komponen yang ada sehingga pengembang tidak perlu membuatnya dari awal lagi sehingga waktu pengembangan menjadi lebih singkat dan efisien.

Sementara itu, kelemahan dari metode RAD adalah [18]:

1. Model RAD menuntut pengembangan dan pelanggan memiliki komitmen di dalam aktivitas *rapid-fire* yang diperlukan untuk melengkapi sebuah sistem, di dalam kerangka waktu yang sangat diperpendek. Jika komitmen tersebut tidak ada, proyek RAD akan gagal.
2. Tidak semua aplikasi sesuai untuk RAD, bila sistem tidak dapat dimodulkan dengan teratur, pembangunan komponen penting pada RAD akan menjadi sangat bermasalah.

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

3. RAD tidak cocok digunakan untuk sistem yang mempunyai risiko teknik yang tinggi.
4. Membutuhkan tenaga kerja yang banyak untuk menyelesaikan sebuah proyek dalam skala besar.

Pada tahapan *User Design* dan *Construction*, pengguna terlibat secara langsung dalam proses pengembangan sistem. Meskipun tidak dijelaskan secara rinci berapa kali iterasi dilakukan, namun dijelaskan bahwa tahapan tersebut bersifat interaktif dan iteratif. Artinya, perancangan prototipe dilakukan secara berulang-ulang berdasarkan masukan pengguna hingga diperoleh desain akhir yang sesuai dengan kebutuhan sistem [19].

2.6 *Unified Modeling Language*

UML adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML mampu merepresentasikan rancangan sistem informasi yang akan dibuat sehingga menjadi sebuah sistem informasi yang siap digunakan oleh pengguna [18].

Tujuan dari *Unified Modeling Language* (UML) sebagai berikut [18]:

1. Memodelkan suatu sistem (bukan hanya perangkat lunak) yang menggunakan konsep berorientasi objek.
2. Menciptakan suatu bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin.
3. Memberikan bahasa yang bebas dari berbagai bahasa pemrograman.

Diagram UML yang digunakan dalam penelitian ini mencakup:

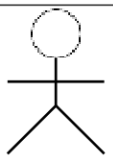
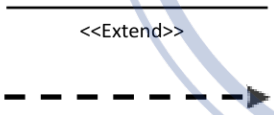
2.6.1 *Use Case* Diagram

Use case diagram merupakan alat visual dalam *Unified Modeling Language* (UML) yang digunakan selama tahap *requirements modeling* untuk merepresentasikan interaksi antara pengguna (disebut aktor) dan sistem informasi. Diagram ini membantu analis sistem dan pengguna untuk mendokumentasikan kebutuhan sistem serta memodelkan fungsi-fungsi sistem dari sudut pandang pengguna. Dalam *use case* diagram, setiap aktor memiliki peran spesifik yang menunjukkan bagaimana ia berinteraksi dengan sistem. Diagram ini menggambarkan skenario penggunaan sistem secara umum melalui hubungan antara aktor dan use case yang relevan [18].

Tabel 2.1 Notasi pada *Use Case* Diagram

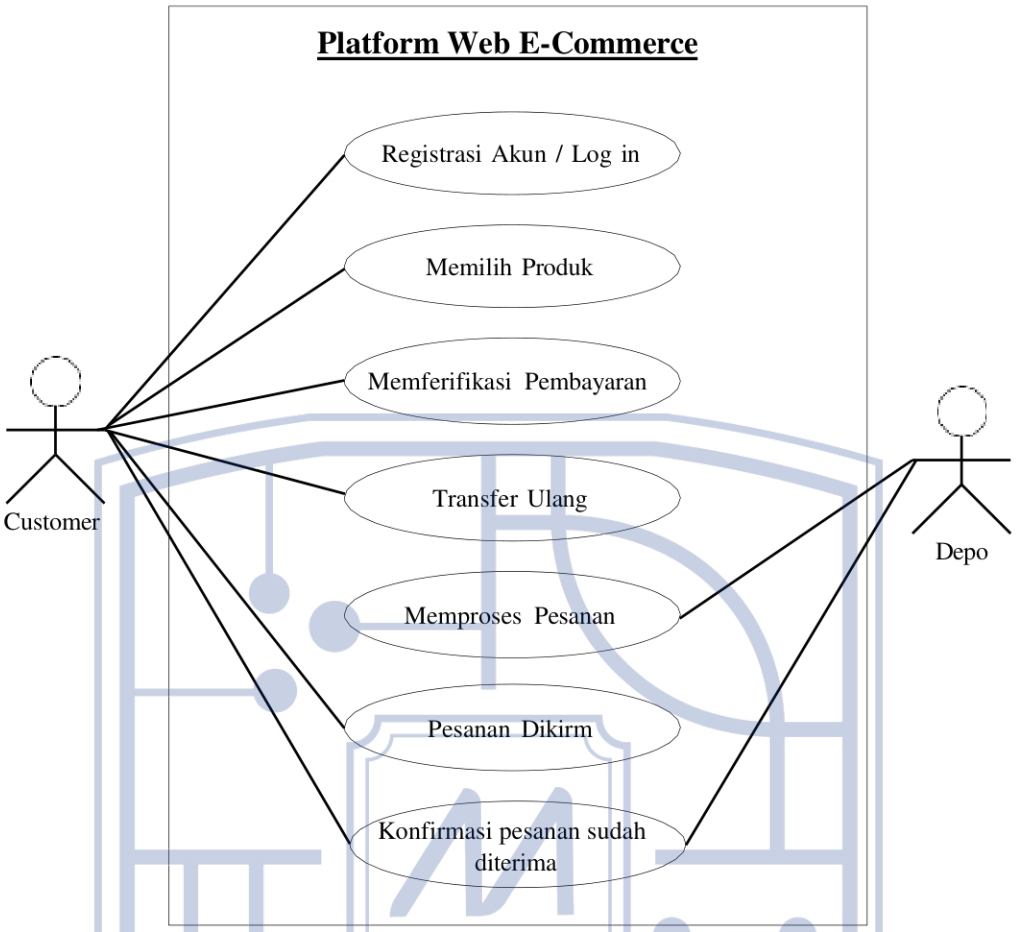
Simbol	**Penjelasan**
	Aktor

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat sendiri, jadi simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang: biasanya dinyatakan menggunakan kata benda diawal *frase* nama aktor.
	Asosisasi Komunikasi antar aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor.
	Generalization Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
	Use Case Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama *Use Case*
	*Extend* Menggambarkan hubungan antar *use case* dimana bahwa sebuah *use case* merupakan fungionalitas *use case* lainnya apabila kondisi tertentu terpenuhi.

Contoh *use case* diagram dapat dilihat pada gambar berikut:

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.



Gambar 2.2 Contoh *Use Case* Diagram

Use case scenario merupakan penjelasan secara tekstual dari sekumpulan skenario interaksi. Setiap skenario mendeskripsikan urutan aksi/langkah yang dilakukan aktor ketika berinteraksi dengan sistem, baik yang berhasil maupun gagal. *Use case scenario* dijelaskan secara tekstual dalam beberapa format tergantung kebutuhannya, yaitu singkat (*brief*), informal (*casual*), atau lengkap (*fully dressed*), yang bisa dijelaskan dalam bentuk tabel dengan 1 kolom atau 2 kolom. Pada format singkat, penjelasan diberikan cukup 1 paragraf yang mengacu hanya pada skenario yang berhasil. Pada format informal, penjelasan diberikan dalam beberapa paragraf yang mencakup semua skenario, baik yang berhasil maupun gagal. Sedangkan, pada format lengkap, penjelasan dibuat secara detil disertai dengan bagianbagian pendukung yang penting [18]. Format terakhir ini yang banyak digunakan di dalam praktik. Bagian-bagian penting tersebut adalah [18]:

a. aktor primer (*primary actor*), yaitu aktor yang menginisiasi layanan sistem untuk mencapai tujuan dari aktor tersebut. Jumlah aktor primer dimungkinkan lebih dari 1.
b. prakondisi (*preconditions*), yaitu kondisi spesifik yang harus terpenuhi sebelum sebuah *Use case* bisa diinisiasi atau dieksekusi oleh aktor primer. Jumlah prakondisi bisa lebih dari 1 keadaan.
c. alur utama (*main or basic flow*), yaitu jalur interaksi yang mengarahkan pada skenario yang berhasil sehingga tujuan aktor bisa terpenuhi. Jalur ini hanya terdiri dari 1 jalur saja.

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

d. alur alternatif (*alternative flows*), yaitu jalur alternatif dari interaksi yang terjadi antar aktor dengan sistem yang mencakup pencabangan (pilihan) maupun skenario yang gagal sehingga tujuan aktor tidak terpenuhi. Jalur ini bisa terdiri dari lebih dari 1 jalur kemungkinan.
e. kondisi akhir (*postconditions*), yaitu kondisi spesifik yang harus terjadi ketika *Use case* berhasil dijalankan atau dieksekusi secara lengkap, sebagai representasi dari tujuan yang ingin dicapai oleh aktor primer. Jumlah kondisi akhir bisa lebih dari 1 keadaan
f. asumsi (*assumptions*), yaitu kondisi yang dianggap benar selama *use case* berlangsung, meskipun tidak dijamin oleh sistem. Asumsi ini memberi konteks tambahan yang dapat memengaruhi jalannya skenario.

Beberapa kesalahan di dalam penulisan *Use case scenario* akan dijelaskan secara singkat berdasarkan klasifikasinya [20].

a. Fokus pada diagram

Penjelasan secara tekstual dalam *use case scenario* adalah hal utama dalam pemodelan *use case*, dan bukan pada *use case* diagram. Kesalahan yang sering dilakukan adalah lebih banyak menghabiskan waktu untuk membuat *use case* diagram dibandingkan menuliskan *use case scenario*. Rekomendasi perbaikannya adalah kembali fokus pada penulisan *use case scenario* dengan detil dan akurat dalam porsi yang lebih dibandingkan pembuatan *use case* diagram.

b. Penamaan *use case* yang salah

Penamaan *use case* diawali tidak dengan menggunakan kata kerja dan lebih merepresentasikan sesuatu berdasarkan sudut pandang sistem dan bukan aktor. Sebagai contoh, nama *Use Case Pemrosesan Transfer Dana* adalah contoh yang salah karena diawali dengan kata benda (yaitu pemrosesan) dan merepresentasikan apa yang dilakukan oleh sistem (yaitu memproses). Rekomendasi perbaikannya adalah menggunakan nama yang diawali dengan kata kerja dan merepresentasikan tujuan yang ingin dicapai oleh aktor, sehingga menjadi *Transfer Dana*.

c. Penjelasan yang terlalu umum

Dalam menjelaskan interaksi yang terjadi antara aktor dengan sistem pada alur utama dan/atau alur alternatif, informasi yang diberikan bersifat umum sebagaimana diilustrasikan dalam contoh potongan interaksi berikut.

1. Sistem menampilkan *form* isian data pribadi.
2. Aktor mengisi *form* data pribadi.

Pada alur tersebut, informasi tentang data pribadi yang harus ditampilkan oleh sistem dan diisi oleh aktor tidak dijelaskan secara spesifik, terdiri dari elemen data apa saja. Deskripsi yang dituliskan dalam *use case scenario* seperti ini akan menyulitkan pada tahapan pemodelan berikutnya, karena penjelasan ini akan menjadi salah satu bahan dalam mengidentifikasi objek-objek yang akan terlibat dalam

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

pendefinisian diagram urutan (*sequence* diagram) maupun diagram kelas (*class* diagram). Rekomendasi perbaikannya adalah menuliskan ulang dalam bentuk yang lebih spesifik sebagaimana contoh potongan interaksi berikut.

1. Sistem menampilkan *form* isian data pribadi yang terdiri dari nama, alamat, tempat lahir, dan tanggal lahir.
2. Aktor mengisi *form* data pribadi secara lengkap.

d. Identifikasi aktor yang salah

Kesalahan terjadi karena identifikasi aktor yang tidak cermat. Hal ini bisa berkaitan dengan kondisi akhir yang tidak konsisten (misalnya untuk kasus *login*) atau konteks sistem yang tidak jelas (pada level model aplikasi atau model bisnis). Untuk kasus *login*, kondisi akhir yang harus terpenuhi adalah aktor sudah bisa dikenali sistem sebagai aktor yang valid, sehingga siapapun aktor (baik yang nantinya valid atau tidak) bisa menginisiasi *login*. Jadi, bukan aktor yang valid saja yang bertindak sebagai aktor primer untuk kasus *login*. Sebagai contoh, aktor *Teller* tidak tepat sebagai aktor primer untuk *Use Case Login* karena *Teller* adalah aktor yang baru teridentifikasi secara valid pada kondisi akhir setelah *Use Case Login* dijalankan secara lengkap. Sehingga rekomendasinya, aktor primer yang diidentifikasi adalah Pengguna, sebagai aktor umum yang belum dikenali oleh sistem. Sedangkan untuk konteks sistem yang tidak jelas, kesalahan terjadi karena level abstraksinya yang tidak sama, apakah aktor pada level model aplikasi atau model bisnis. Jika pada level model aplikasi, maka peran yang berinteraksi langsung dengan sistem yang harus menjadi aktor primer. Rekomendasinya adalah perlu mengidentifikasi aktor secara tepat sesuai dengan konteksnya.

e. Identifikasi prakondisi dan kondisi akhir yang tidak tepat

Prakondisi dan kondisi akhir yang harus dipenuhi tidak dijelaskan secara tepat dan spesifik. Keduanya harus menyatakan sebuah keadaan objek dari domain masalahnya sistem yang bisa diobservasi, bukan menunjukkan sebuah proses atau aksi, baik sebagai syarat sebuah *Use case* bisa dieksekusi (prakondisi) atau sebagai keadaan yang harus terpenuhi ketika *Use case* telah dieksekusi secara lengkap (kondisi akhir). Sebagai contoh, untuk kasus *login* maka prakondisinya adalah halaman *Login* sudah terbuka atau tersedia, dan bukan data Teller sudah ada di basis data. Sedangkan kondisi akhirnya adalah aktor sudah teridentifikasi sebagai Teller. Kedua kondisi sebaiknya dijelaskan dalam bentuk kalimat pasif. Rekomendasi yang perlu dilakukan adalah mendefinisikan prakondisi dan kondisi akhir untuk setiap *Use case* secara spesifik dengan menggunakan kalimat pasif.

f. Tidak ada penjelasan aliran utama dan/atau aliran alternatif

Aliran utama dan/atau aliran alternatif tidak dijelaskan sama sekali di dalam *Use case scenario*, sehingga interaksi yang terjadi antara aktor dengan sistem pada sebuah Use case tidak bisa tergambarkan secara

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

baik. Penjelasan *use case* hanya diberikan untuk hal-hal yang berkaitan dengan aktor primer, deskripsi *use case*, prakondisi, dan kondisi akhir. Rekomendasi perbaikannya adalah kedua aliran harus diberikan penjelasan yang baik, paling tidak aliran utama harus ada dalam sebuah *use case* karena hal itu menggambarkan bagaimana interaksi dasar yang harus terjadi antara aktor dengan sistem sehingga kondisi akhir terpenuhi. Aliran alternatif, baik yang merupakan pencabangan atau skenario gagal, dalam sebuah *use case* mungkin tidak ada sama sekali.

g. Interaksi hanya dari sisi aktor

Use case scenario dijelaskan hanya berdasarkan interaksi yang diinisiasi oleh aktor primer saja. Respon yang diberikan oleh sistem tidak pernah dijelaskan. Penjelasan yang seperti ini tidak lengkap karena tidak mampu menggambarkan bagaimana sistem merespon terhadap apa yang dilakukan oleh aktor, sebagaimana contoh potongan interaksi berikut.

1. Sistem menampilkan form isian data pribadi yang terdiri dari nama, alamat, tempat lahir, dan tanggal lahir.
2. Sistem menyimpan informasi data pribadi yang telah diisi oleh aktor.

Rekomendasi perbaikannya adalah dengan menjelaskan secara lengkap gambaran interaksi antara aktor dengan sistem yang dilihat dari dua sisi, sebagaimana contoh potongan interaksi berikut.

1. Sistem menampilkan form isian data pribadi yang terdiri dari nama, alamat, tempat dan tanggal lahir.
2. Aktor mengisi form secara lengkap kemudian memerintahkan sistem untuk memproses.
3. Sistem melakukan validasi dan menyimpan data.

h. Aliran utama dan/atau aliran alternatif tidak jelas

Kesalahan ini terjadi ketika penjelasan interaksi pada aliran utama dicampuradukkan dengan penjelasan untuk aliran alternatif, sebagaimana contoh potongan interaksi berikut.

Sistem melakukan validasi dan menyimpan data.

1. Jika validasi salah, maka sistem menampilkan kembali form isian data pribadi berikut pesan kesalahan “Data tidak valid”.
2. Jika data tidak berhasil disimpan karena permasalahan di basis data, maka sistem menampilkan pesan “Data gagal disimpan”.
3. Jika data berhasil disimpan maka sistem menampilkan pesan “Data berhasil disimpan”.

Rekomendasi perbaikannya adalah penjelasan aliran utama harus terpisah dengan aliran alternatif. Pada aliran utama tidak boleh ada penjelasan yang mengandung pilihan dan/atau kegagalan, sebagaimana contoh potongan interaksi berikut.

1. Sistem melakukan validasi dan menyimpan data.
2. Sistem menampilkan pesan “Data berhasil disimpan”.

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

Pilihan dan/atau kegagalan harus dijelaskan pada bagian aliran alternatif secara terstruktur dengan mengacu pada urutan interaksi yang berpotensi terjadinya kegagalan dan/atau pilihan dalam aliran utama, sebagaimana contoh berikut.

Kegagalan validasi dan penyimpanan:

1. Jika validasi salah, maka sistem menampilkan kembali form isian data pribadi berikut pesan kesalahan "Data tidak valid".
2. Jika data tidak berhasil disimpan karena permasalahan di basis data, maka sistem menampilkan pesan "Data gagal disimpan".

i. Penjelasan yang bersifat *white-box*

Kesalahan ini terjadi jika penjelasan dalam aliran utama dan/atau aliran alternatif diberikan terlalu teknis yang berkaitan dengan bagaimana sistem bekerja secara internal, sebagaimana contoh potongan interaksi berikut.

Sistem melakukan validasi dan menyimpan data ke basis data pada Tabel DataPribadi dengan menggunakan stored procedure sp_DataPribadi.

Rekomendasi perbaikannya adalah dengan memberikan penjelasan yang bersifat *black-box*, yaitu menjelaskan apa (responsibilities) yang harus dilakukan oleh sistem dan bukan bagaimana sistem melakukannya, sebagaimana contoh potongan interaksi berikut.

Sistem melakukan validasi dan menyimpan data.

j. Hanya menjelaskan aliran utama/normal

Kesalahan ini terjadi jika dalam *Use case scenario* hanya menjelaskan aliran utama/normal saja tanpa aliran alternatif, baik yang berupa pencabangan (kondisi/opsi/pilihan) maupun yang berupa kegagalan. Rekomendasi perbaikannya adalah setiap *Use case scenario* harus mengandung berbagai skenario kegagalan dan/atau skenario yang mungkin sebagai pilihan yang harus diputuskan oleh aktor.

k. Penyebutan aktor yang inkonsisten dengan yang ada di *use case* diagram

Kesalahan ini terjadi jika aktor (primer/sekunder) yang disebutkan dalam *Use case scenario* tidak sesuai dengan sebutan aktor yang digunakan dalam *use case* diagram. Sebagai contoh, aktor yang disebutkan dalam *Use case scenario*adalah Pengguna, tetapi yang digambarkan dalam *Use case* diagram adalah aktor *User*. Rekomendasi perbaikannya adalah dengan menyesuaikan penyebutan semua nama aktor yang ada di *Use case scenario* dan *Use case* diagram.

2.6.2 *Activity* Diagram

Activity diagram merupakan diagram dalam UML yang digunakan untuk memodelkan urutan aktivitas dalam suatu proses sistem. Diagram ini menggambarkan bagaimana aliran kerja atau langkah-langkah

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

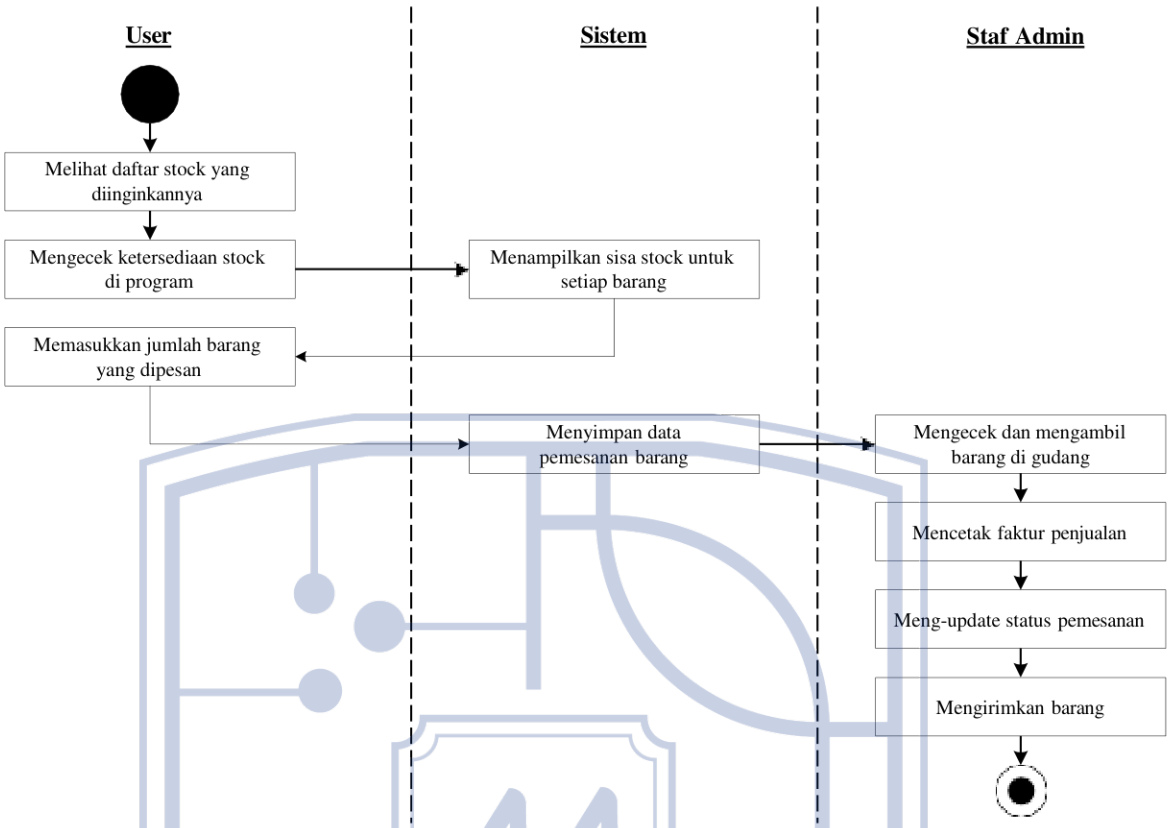
operasional dilakukan, baik oleh pengguna maupun sistem, termasuk percabangan dan proses paralel yang mungkin terjadi selama eksekusi suatu fungsi [18].

Tabel 2.2 Notasi pada *Activity Diagram*

Simbol	**Penjelasan**
	Status Awal Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas Aktivitas biasanya diawali dengan kata kerja.
	Penggabungan Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
	Percabangan Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
	Status akhir Sebuah diagram aktivitas memiliki sebuah status akhir.

Contoh *activity* diagram dapat dilihat pada gambar berikut:

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.



Gambar 2.3 Contoh *Activity* Diagram

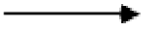
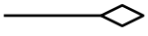
2.6.3 *Class* Diagram

***Class* diagram** adalah representasi visual yang menunjukkan struktur sistem dengan mendefinisikan kelas-kelas yang mewakili objek-objek penting dalam sistem, beserta atribut dan operasi (metode) yang dimilikinya. Diagram ini menggambarkan bagaimana objek-objek saling berhubungan satu sama lain, serta membantu dalam memodelkan sistem dari perspektif objek. *Class* diagram digunakan sebagai jembatan antara desain sistem dan implementasi perangkat lunak dalam pemrograman berbasis objek [18].

Tabel 2.3 Notasi pada *Class* Diagram

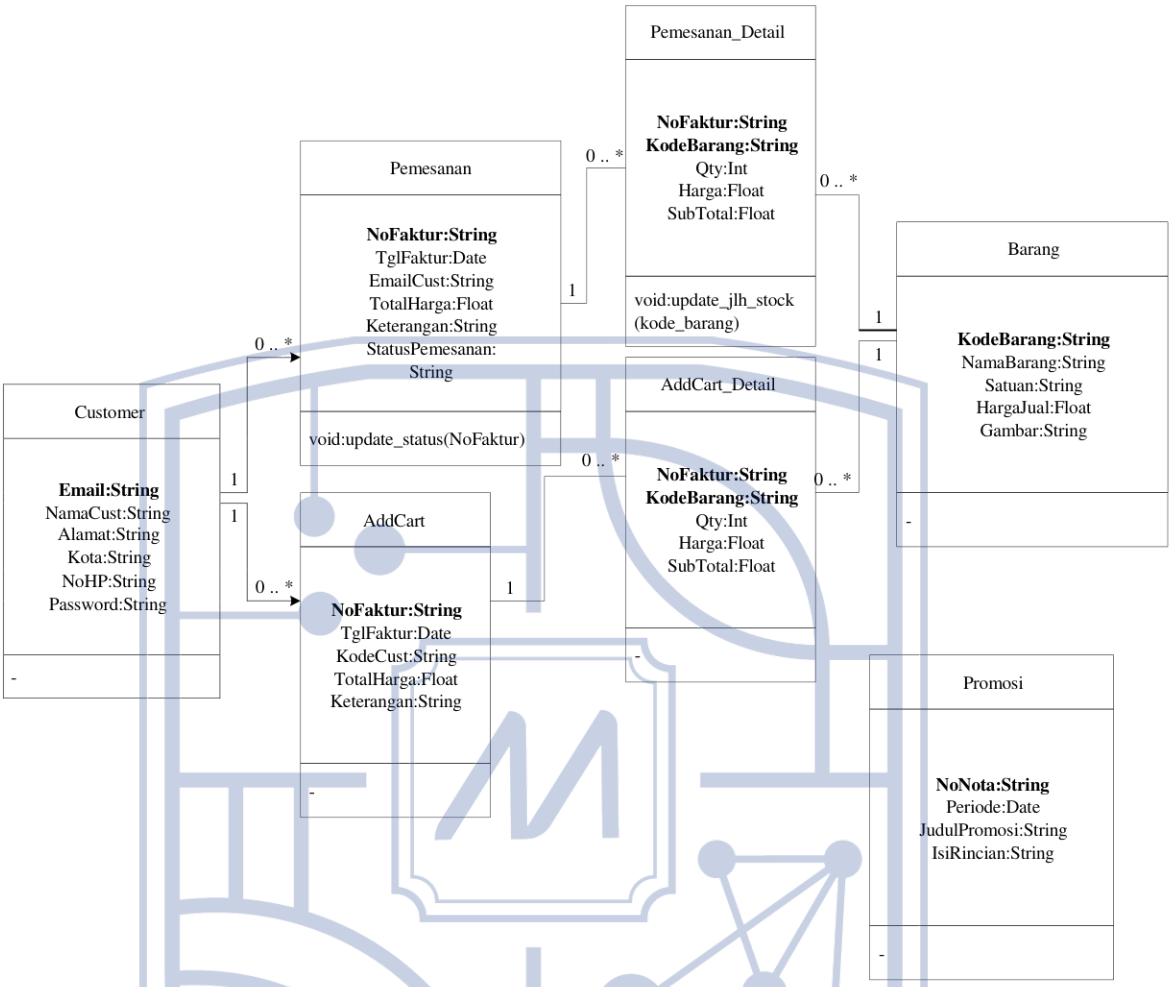
Simbol	Keterangan
CLASS ← 1 ← 2 ← 3	*Class* Kelas pada struktur sistem. 1. Nama Kelas, pada suatu kelas haruslah disertai dengan nama.

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

	2. Atribut, merupakan perlengkapan pada suatu kelas. Nilai diproses sesuai dengan atributnya. 3. Operasi, merupakan proses ekseskusi suatu kelas baik itu pada kelasnya sendiri maupun kelas lainnya.
	Asosiasi Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*.
	Asosiasi berarah Relasi antar kelas dengan makna satu kelas digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity*.
	Generalisasi Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
	Kebergantungan Relasi antar kelas dengan makna kebergantungan antarkelas.
	Agregasi Relasi antarkelas dengan makna semua bagian (*whole-part*)

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

Contoh *class* diagram dapat dilihat pada gambar berikut:



Gambar 2.4 Contoh Class Diagram

2.7 Persyaratan Sistem

Dalam pengembangan sistem informasi, identifikasi kebutuhan sistem sangat penting untuk memastikan sistem yang dibangun sesuai dengan harapan pengguna. Kebutuhan sistem umumnya dibagi menjadi dua kategori [21]:

1. Persyaratan Fungsional : Persyaratan ini mencakup fungsi-fungsi utama yang harus dilakukan oleh sistem, seperti proses input, pengolahan data, dan output yang dihasilkan.
2. Persyaratan Non-Fungsional : Persyaratan ini berkaitan dengan kualitas sistem, termasuk aspek keamanan, keandalan, efisiensi, dan kemudahan penggunaan.

Untuk membantu dalam analisis kebutuhan sistem, salah satu metode yang dapat digunakan adalah kerangka PIECES. PIECES merupakan akronim dari [21]:

1. Performance (Kinerja): Seberapa baik sistem menjalankan tugasnya.
2. Information (Informasi): Kualitas dan kuantitas informasi yang dihasilkan oleh sistem.

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

3. Economy (Ekonomi): Biaya yang terkait dengan operasional sistem.
4. Control (Kontrol): Kemampuan sistem dalam mengendalikan dan mengamankan data.
5. Efficiency (Efisiensi): Penggunaan sumber daya secara optimal.
6. Service (Pelayanan): Tingkat layanan yang diberikan oleh sistem kepada pengguna.

2.8 *Database*

Sebuah basis data (*database*) adalah sebuah kumpulan data yang saling berhubungan secara logis, dan merupakan sebuah penjelasan dari data tersebut, yang didesain untuk menemukan data yang dibutuhkan oleh sebuah organisasi[21] .

Di dalam basis data, semua data diintegrasikan dengan menghindari duplikasi data. Basis data dapat digunakan oleh banyak departemen dan pemakai. Basis data tidak hanya memegang data operasional organisasi, tetapi juga penjelasan mengenai data tersebut. Karena alasan tersebut basis data dapat juga dideskripsikan sebagai kumpulan data yang saling terintegrasi. Basis data juga merupakan sekumpulan elemen data terintegrasi yang secara logika saling berhubungan. Basis data mengonsolidasikan berbagai catatan yang terlebih dahulu disimpan dalam file-file terpisah ke dalam satu gabungan umum elemen data yang menyediakan data untuk banyak aplikasi. Elemen data mendeskripsikan entitas-entitas dan hubungan antara entitas-entitas tersebut [21] .

Dalam implementasinya, untuk memudahkan dalam mengakses data, data disusun dalam suatu struktur logis yang menjelaskan bahwa [22] :

1. Kumpulan tabel menyusun basis data.
2. Tabel tersusun atas sejumlah *record*.
3. Sebuah *record* mengandung sejumlah *field*.
4. Sebuah *field* disimpan dalam bentuk kumpulan bit.

Berikut merupakan pengertian dari istilah di atas [22]:

1. *Field* menyatakan data terkecil yang memiliki makna. Istilah lain untuk *field* yaitu elemen data, kolom item, dan atribut. Contoh *field* yaitu nama, alamat, telepon, dan jenis kelamin.
2. *Record* menyatakan kumpulan dari sejumlah elemen data yang saling terkait. Sebagai contoh, nama, alamat, bit dan jenis kelamin dari seseorang menyusun sebuah *record*. Istilah lain yang juga menyatakan *record* yaitu tupel dan baris.
3. Tabel menghimpun sejumlah *record*. Sebagai contoh, data pribadi dari semua pegawai disimpan dalam sebuah tabel.
4. Basis data (*database*), adalah sekumpulan data terintegrasi, yang diorganisasi untuk memenuhi kebutuhan para pemakai di dalam suatu organisasi. Sebagai contoh, basis data akademik

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

Dalam pengembangan suatu sistem informasi, basis datamemiliki peran yangpenting dan sangat diperlukan. Berikut merupakan alasan perlunya basis data [22]:

1. Salah satu komponen penting dalam suatu sistem informasiadalah basis data, karena basis data merupakan dasar dalammenyediakan informasi.
2. Informasi dikatakan lebih bernilai jika memiliki manfaat yang efektif dibandingkan dengan biaya dalam mendapatkannya. Dalam hal ini, basis data akan menentukan kualitas informasi.
3. Data akan dapat saling berelasi dengan mengimplementasikan basis data.
4. Basis data mengurangi duplikasi data (*data redudancy*). Basis data dapat mengurangi pemborosan tempat simpanan luar.

Basis data dirancang sedemikian rupa untuk selanjutnya diintegrasikan dengan sistem informasi. Orang yang berkepentingan dengan basis data meliputi [22]:

1. Pemakai akhir dan vendor DBMS (*Data Base Management System*).
2. Programmer aplikasi basis data.
3. Administrator Basis Data (*Database Administrator*).

Tujuan dalam merancang basis data adalah [22]:

1. Kecepatan dan kemudahan (*Speed*)

   Pemanfaatan basis data memungkinkan untuk dapat menyimpan data atau melakukan perubahan/manipulasi terhadap data atau menampilkan kembali data tersebut dengan cepat dan mudah.
2. Efisiensi ruang penyimpanan (*Space*)

   Penggunaan ruang penyimpanan di dalam basis data dilakukan untuk mengurangi jumlah *redudancy* (pengulangan) data, baik dengan melakukan penerapan sejumlah pengkodeanatau dengan membuat relasi-relasi (dalam bentuk *file*) antarkelompok data yang saling berhubungan.
3. Keakuratan (*Accuracy*)

   Pemanfaatan pengkodean atau pembentukan relasi antar databersama dengan penerapan aturan/batasan tipe data, domain data, keunikan data, dan sebagainya, yang diterapkan dalambasis data, sangat berguna untuk menentukan ketidakakuratanpemasukan atau penyimpanan data.
4. Ketersediaan (*Availability*)

   Pertumbuhan data (baik dari jumlah maupun jenisnya) sejalan dengan waktu akan semakin membutuhkan ruang penyimpanan yang besar. Data yang sudah jarang atau bahkan tidak pernahlagi digunakan dapat diatur untuk dilepaskan dari sistem basisdata dengan cara penghapusan atau dengan memindahkannya ke media penyimpanan.
5. Kelengkapan (*Completeness*)

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

Lengkap atau tidaknya data yang dikelola bersifat relatif, baik terhadap kebutuhan pemakai maupun terhadap waktu. Dalam sebuah basis data, struktur dari basis data harus disimpan. Untuk mengakomodasi kebutuhan kelengkapan data yangsemakin berkembang, tidak cukup hanya menambah *record-record* data, tetapi juga harus melakukan penambahan struktur dalam basis data.

6. Keamanan (*Security*)

   Sistem keamanan digunakan untuk dapat menentukan siapa saja yang boleh menggunakan basis data dan menentukan jenisoperasi apa saja yang boleh dilakukan.

7. Kebersamaan pemakai (*Shareability*)

   Pemakai basis data sering kali tidak terbatas hanya pada satupemakaian saja atau oleh satu sistem aplikasi saja. Basis data yang dikelola oleh sistem (aplikasi) yang mendukung lingkungan*multiuser*, akan dapat memenuhi kebutuhan ini, tetapi denganmenjaga/menghindari terhadap munculnya persolan baruseperti inkonsistensi data (karena data yang sama diubah oleh banyak pemakai pada saat bersamaan).

Database Management System (DBMS) adalah perangkatlunak yang didesain untuk menangani dalam hal pemeliharaan dan utilitas kumpulan data dalam jumlah besar. DBMS dapat menjadi alternatif penggunaan secara khusus untuk aplikasi, misalnya penyimpanan data dalam *field* dan menulis kode aplikasi yang spesifik untuk pengaturannya[22].

Fungsi-fungsi yang disediakan DBMS antara lain [22]:

1. Penyimpanan, pengambilan dan perubahan data. SebuahDBMS harus menyediakan kemampuan menyimpan,mengambil dan merubah data dalam basis data.
2. Katalog yang dapat diakses pemakai. Menyediakan sebuahkatalog yang berisi deskripsi item data yang disimpan dandiakses oleh pemakai.
3. Mendukung transaksi. Menyediakan mekanisme yang akan menjamin semua perubahan yang berhubungan dengan transaksi yang sudah ada atau yang akan dibuat.
4. Melayani *Control concurrency*. Sebuah DBMS harus menyediakan mekanisme yang menjamin basis data ter-*update* secara benar pada saat beberapa pemakai melakukan perubahan terhadap basis data yang sama secara bersamaan.
5. Melayani *recovery*. Menyediakan mekanisme untuk mengembalikan basis data ke keadaan sebelum terjadinya kerusakan pada basis data tersebut.
6. Melayani autorisasi. Sebuah DBMS harus menyediakan mekanisme untuk menjamin bahwa hanya pemakai yang berwenang saja yang dapat mengakses basis data.
7. Mendukung komunikasi data. Sebuah DBMS harus mampu terintegrasi dengan *software* komunikasi.
8. Melayani *integrity*. Sebuah DBMS bertujuan untuk menjamin semua data dalam basis data dan setiap terjadi perubahandata harus sesuai dengan aturan yang berlaku.

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

9. Melayani data *independence*. Sebuah DBMS harus mencakup fasilitas untuk mendukung kemandirian program dari struktur basis data yang sesungguhnya.
10. Melayani *utility*. Sebuah DBMS sebaiknya menyediakan kumpulan layanan *utility*.

2.9 *Website*

Secara etimologi, *website* adalah kumpulan dari halaman-halaman situs yang terangkum dalam sebuah domain atau sub domain yang tempatnya berada dalam *World Wide Web* (WWW) di dalam internet. *Website* adalah keseluruhan halamanhalaman *web* yang terdapat dari sebuah domain yang mengandung informasi[23].

Sebuah halaman *web* biasanya berupa dokumen yang ditulis dalam format *HTML* (*Hyper Text Markup Language*) yang selalu bisa diakses melalui *HTTP*, yaitu sebuah protokol yang menyampaikan informasi dari *server website* untuk ditampilkan kepada para pemakai melalui *web browser*. Semua publikasi dari *website* tersebut dapat membentuk sebuah jaringan informasi yang sangat besar [24].

World Wide Web (WWW atau *web*) merupakan sistem informasi terdistribusi yang berbasis *hypertext*. *World Wide Web* merupakan jaringan dokumentasi yang sangat besar yang saling berhubungan satu dengan yang lain. Satu set protokol yang mendefinisikan bagaimana sistem bekerja dan mentransfer data, dan sebuah *software* yang membuatnya bekerja dengan mulus. *Web* menggunakan teknik *hypertext* dan multimedia yang membuat internet mudah digunakan dijelajahi dan dikontribusikan. *Web* merupakan system *hypermedia* yang berarea luas yang ditujukan untuk akses secara universal. Salah satu kuncinya adalah kemudahan tempat seseorang atau perusahaan dapat menjadi bagian dari *web* berkontribusi pada *web* [24].

Sebuah situs *web* (sering pula disingkat menjadi situs saja, *website* atau site) adalah sebutan bagi sekelompok halaman *web* (*web* page), yang umumnya merupakan bagian dari suatu nama domain (domain name) atau subdomain di World Wide *Web* (WWW) di Internet. Sebuah *webpage* adalah dokumen yang ditulis dalam format HTML (*Hyper Text Markup Language*), yang hampir selalu bisa diakses melalui HTTP, yaitu protokol yang menyampaikan informasi dari *serverwebsite* untuk ditampilkan kepada para pemakai melalui *webbrowser* baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*). Bersifat statis apabila isi informasi *website* tetap, jarang berubah, dan isi informasinya searah hanya dari pemilik *website*. Bersifat dinamis apabila isi informasi *website* selalu berubah-ubah, dan isi informasinya interaktif dua arah berasal dari pemilik serta pengguna *website*. Contoh *website* statis adalah berisi profil perusahaan, sedangkan *website* dinamis adalah seperti *Friendster, Multiply*, dll. Dalam sisi pengembangannya, *website* statis hanya bisa diupdate oleh pemiliknya saja, sedangkan *website* dinamis bisa diupdate oleh pengguna maupun pemilik. Halaman-halaman sebuah situs *web* diakses dari sebuah URL yang

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

menjadi “akar” (*root*), yang disebut *homepage* (halaman induk; sering diterjemahkan menjadi “beranda”, “halaman muka”), dan biasanya disimpan dalam server yang sama. Tidak semua situs *web* dapat diakses dengan gratis. Beberapa situs *web* memerlukan pembayaran agar dapat menjadi *customer*, misalnya situs-situs yang menampilkan pornografi, situs-situs berita, layanan surat elektronik (*e-mail*), dan lain-lain. *Website* ini dibuka melalui sebuah program penjelajah (*Browser*) yang berada di sebuah komputer. Program penjelajah yang bisa digunakan dalam komputer diantaranya: *IE (Internet Explorer), Mozilla, Firefox, Netscape, Opera dan yang terbaru adalah Google Chrome* [24].

2.8.1 Unsur-Unsur *Website*

Untuk menyediakan sebuah *website*, maka harus tersedia unsur-unsur penunjangnya, adalah sebagai berikut [24] :

1. Nama domain (Domain name/URL – *Uniform Resource Locator*)

   Nama domain atau biasa disebut dengan Domain Name atau URL adalah alamat unik di dunia internet yang digunakan untuk mengidentifikasi sebuah *website*, atau dengan kata lain domain name adalah alamat yang digunakan untuk menemukan sebuah *website* pada dunia internet. Contoh : http://www.baliorange.net. Nama domain diperjualbelikan secara bebas di internet dengan status sewa tahunan. Setelah Nama Domain itu terbeli di salah satu penyedia jasa pendaftaran, maka pengguna disediakan sebuah kontrol panel untuk administrasinya. Jika pengguna lupa/tidak memperpanjang masa sewanya, maka nama domain itu akan di lepas lagi ketersediaannya untuk umum. Nama domain sendiri mempunyai identifikasi ekstensi/akhiran sesuai dengan kepentingan dan lokasi keberadaan *website* tersebut. Contoh nama domain ber-ekstensi internasional adalah com, net, org, info, biz, name, ws. Contoh nama domain ber-ekstensi lokasi Negara Indonesia adalah :

   a. .co.id : Untuk Badan Usaha yang mempunyai badan hukum sah.
   b. .ac.id : Untuk Lembaga Pendidikan.
   c. .go.id : Khusus untuk Lembaga Pemerintahan Republik Indonesia.
   d. .mil.id : Khusus untuk Lembaga Militer Republik Indonesia.
   e. .or.id : Untuk segala macam organisasi yand tidak termasuk dalam kategori “ac.id”,”co.id”,”go.id”,”mil.id” dan lain lain.
   f. .war.net.id : untuk industri warung internet di Indonesia.
   g. .sch.id : khusus untuk Lembaga Pendidikan yang menyelenggarakan pendidikan seperti SD, SMP dan atau SMU.
   h. .*web*.id : Ditujukan bagi badan usaha, organisasi ataupun perseorangan yang melakukan kegiatannya di World Wide *Web*.

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

2. Rumah tempat *website* (*Webhosting*)

   WebHosting dapat diartikan sebagai ruangan yang terdapat dalam harddisk tempat menyimpan berbagai data, *file-file*, gambar, video, data email, statistik, *database* dan lain sebagainya yang akan ditampilkan di *website*. Besarnya data yang bisa dimasukkan tergantung dari besarnya *webhosting* yang disewa/dipunyai, semakinbesar *webhosting* semakin besar pula data yang dapat dimasukkan dan ditampilkan dalam *website*. *WebHosting* juga diperoleh dengan menyewa. Pengguna akan memperoleh kontrol panel yang terproteksi dengan *username* dan *password* untuk administrasi *website*nya. Besarnya *hosting* ditentukan ruangan *harddisk* dengan ukuran MB (Mega Byte) atau GB (Giga Byte). Lama penyewaan *webhosting* rata-rata dihitung per tahun. Penyewaan *hosting* dilakukan dari perusahaan-perusahaan penyewa *webhosting* yang banyak dijumpai baik di Indonesia maupun Luar Negeri. Lokasi peletakan pusat data (*datacenter*) *webhosting* bermacam-macam. Ada yang di Jakarta, Singapore, Inggris, Amerika, dll dengan harga sewa bervariasi.

3. Bahasa Program (*Scripts Program*).

   Adalah bahasa yang digunakan untuk menerjemahkan setiap perintah dalam *website* yang pada saat diakses. Jenis bahasa program sangat menentukan statis, dinamis atau interaktifnya sebuah *website*. Semakin banyak ragam bahasa program yang digunakan maka akan terlihat *website* semakin dinamis, dan interaktif serta terlihat bagus. Beragam bahasa program saat ini telah hadir untuk mendukung kualitas *website*. Jenis jenis bahasa program yang banyak dipakai para desainer *website* antara lain HTML, ASP, PHP, JSP, Java Scripts, Java applets, XML, Ajax dsb. Bahasa dasar yang dipakai setiap situs adalah HTML sedangkan PHP, ASP, JSP dan lainnya merupakan bahasa pendukung yang bertindak sebagai pengatur dinamis, dan interaktifnya situs. Bahasa program ASP, PHP, JSP atau lainnya bisa dibuat sendiri. Bahasa program ini biasanya digunakan untuk membangun portal berita, artikel, forum diskusi, buku tamu, anggota organisasi, *email*, *mailing list* dan lain sebagainya yang memerlukan update setiap saat.

4. Desain *website*.

   Setelah melakukan penyewaan *domain name* dan *webhosting* serta penguasaan bahasa program (*scripts program*), unsur *website* yang penting dan utama adalah desain. Desain *website* menentukan kualitas dan keindahan sebuah *website*. Desain sangat berpengaruh kepada penilaian pengunjung akan bagus tidaknya sebuah *website*. Untuk membuat *website* biasanya dapat dilakukan sendiri atau menyewa jasa *websitedesigner*. Saat ini sangat banyak jasa *webdesigner*, terutama di kota- kota besar. Perlu diketahui bahwa kualitas situs sangat ditentukan oleh kualitas *designer*. Semakin banyak penguasaan *webdesigner* tentang beragam program/software pendukung pembuatan situs maka akan dihasilkan situs yang semakin berkualitas, demikian pula sebaliknya. Jasa *webdesigner* ini yang umumnya memerlukan biaya yang

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

tertinggi dari seluruh biaya pembangunan situs dan semuanya itu tergantung kualitas *designer*. Program-program desain *website* salah satunya adalah Macromedia Firework, Adobe Photoshop, Adobe Dreamweaver, Microsoft Frontpage, dll.

5. Program transfer data ke pusat data.

   Para *webdesigner* mengerjakan *website* dikomputernya sendiri. Berbagai bahasa program, data informasi teks, gambar, video, dan suara telah menjadi *file-file* pendukung adanya *website*. *File* tersebut bisa dibuka menggunakan program penjelajah (browser) sehingga terlihatlah sebuah *website* utuh di dalam komputer sendiri (offline). Tetapi *file-file* tersebut perlu untuk diletakkan dirumah *hosting* versi *online* agar terakses ke seluruh dunia. Pengguna akan diberikan akses FTP (*FileTransfer Protocol*) setelah memesan sebuah *webhosting* untuk memindahkan *file-filewebsite* ke pusat data *webhosting*. Untuk dapat menggunakan FTP diperlukan sebuah program FTP, misalnya WS FTP, Smart FTP, Cute FTP, dll. Program FTP ini banyak ditemui di internet dengan status penggunaan gratis maupun harus membayar. Para *webdesigner* pun dapat menggunakan fasilitas FTP yang terintegrasi dengan program pembuat *website*, misal Adobe Dreamweaver.

6. Publikasi *website*.

   Keberadaan *website* tidak ada gunanya dibangun tanpa dikunjungi atau dikenal oleh masyarakat atau pengunjung internet. Karena efektif tidaknya situs sangat tergantung dari besarnya pengunjung dan komentar yang masuk. Untuk mengenalkan situs kepada masyarakat memerlukan apa yang disebut publikasi atau promosi. Publikasi situs di masyarakat dapat dilakukan dengan berbagai cara seperti dengan pamlet-pamlet, selebaran, baliho, kartu nama dan lain sebagainya tapi cara ini bisa dikatakan masih kurang efektif dan sangat terbatas. Cara yang biasanya dilakukan dan paling efektif dengan tak terbatas ruang atau waktu adalah publikasi langsung di internet melalui search engine-search engine (mesin pencari, spt : *Yahoo, Google, MSN*, Search Indonesia, dsb). Cara publikasi di search engine ada yang gratis dan ada pula yang membayar. Yang gratis biasanya terbatas dan cukup lama untuk bisa masuk dan dikenali di search engine terkenal seperti Yahoo atau Google. Cara efektif publikasi adalah dengan membayar, walaupun harus sedikit mengeluarkan akan tetapi situs cepat masuk ke search engine dan dikenal oleh pengunjung.

7. Pemeliharaan *Website*.

   Untuk mendukung kelanjutan dari situs diperlukan pemeliharaan setiap waktu sesuai yang diinginkan seperti penambahan informasi, berita, artikel, link, gambar atau lain sebagainya. Tanpa pemeliharaan yang baik situs akan terkesan membosankan atau monoton juga akan segera ditinggal pengunjung. Pemeliharaan situs dapat dilakukan per periode tertentu seperti tiap hari, tiap minggu atau tiap bulan sekali secara rutin atau secara periodik saja tergantung kebutuhan (tidak rutin). Pemeliharaan rutin

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

biasanya dipakai oleh situs-situs berita, penyedia artikel, organisasi atau lembaga pemerintah. Sedangkan pemeliharaan periodik bisanya untuk situs-situs pribadi, penjualan/e-commerce, dan lain sebagainya.

8. Perpanjangan Masa Sewa Domain Name dan *WebHosting*

   Perlu dipahami bahwa domain name dan *webhosting* berstatus sewa. Selama kedua hal itu dibayarkan masa sewa perpanjangannya, maka Anda berhak untuk memilikinya dan mempergunakannya. Banyak terjadi kasus kelupaan dalam memperpanjang masa sewanya, atau sulit untuk menghubungi pihak ketiga (*webdesigner*) sebagai perantara pendaftaran awal, maka akan berakibat fatal. Anda akan kehilangan domain name sebagai identitas dalam dunia internet.

2.8.2 Fungsi *Website*

Website mempunyai fungsi yang bermacam-macam, tergantung dari tujuan dan jenis *website* yang dibangun, tetapi secara garis besar dapat berfungsi sebagai [24]:

1. Media Promosi

   Sebagai media promosi dapat dibedakan menjadi media promosi utama, misalnya *website* yang berfungsi sebagai search engine atau toko *Online*, atau sebagai penunjang promosi utama, namun *website* dapat berisi informasi yang lebih lengkap daripada media promosi *offline* seperti koran atau majalah

2. Media Pemasaran

   Pada toko *online* atau system afiliasi, *website* merupakan media pemasaran yang cukup baik, karena dibandingkan dengan toko sebagaimana di dunia nyata, untuk membangun toko *online* diperlukan modal yangr relatif lebih kecil, dan dapat beroperasi 24 jam walaupun pemilik *website* tersebut sedang istirahat atau sedang tidak ditempat, serta dapat diakses darimana saja.

3. Media Informasi

   Website portal dan radio atau tv *online* menyediakan informasi yang bersifat global karena dapat diakses dari mana saja selama dapat terhubung ke internet, sehingga dapat menjangkau lebih luas daripada media informasi konvensional seperti koran, majalah, radio atau televisi yang bersifat lokal.

4. Media Pendidikan

   Ada komunitas yang membangun *website* khusus berisi informasi atau artikel yang sarat dengan informasi ilmiah misalnya wikipedia.

5. Media Komunikasi

   Sekarang banyak terdapat *website* yang dibangun khusus untuk berkomunikasi seperti forum yang dapat memberikan fasilitas bagi para anggotanya untuk saling berbagi informasi atau membantu pemecahan masalah tertentu.

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

2.10 *User Interface* (UI)

User Interface (UI) adalah tempat dimana manusia dan mesin melakukan interaksi, seperti antarmuka pengguna dengan software atau hardware komputer. Ini mencakup semua yang pengguna lihat, dengar, dan sentuh pada layar atau perangkat fisik, seperti tombol, menu, ikon, dan pengaturan. *User Interface* merupakan proses pembuatan tampilan pada sebuah software yang berfokus pada design dan tampilan yang dibuat harus intuitif dan interaktif. Untuk mendapatkan *User* Experience (UX) yang baik, design *User Interface* (UI) harus mencakup keterlibatan visual dan interaksi yang mudah dipahami dan digunakan[25].

Tampilan visual dari sebuah aplikasi, *website*, *hardware*, atau *software* yang menentukan respon pengguna ketika berinteraksi dengan *system* tersebut. *User Interface* juga termasuk kedalam HCI (*Human Computer Interaction*). HCI merupakan segala aspek interaksi antara pengguna dengan komputer, tidak hanya hardware. Semuanya yang terlihat dilayar, dokumen yang dapat diedit mengunakan mouse dan keyboard merupakan bagian dari *user interface* [25].

User Interface digunakan untuk menghubungkan informasi dari pengguna ke *system*, sehingga komputer dapat berfungsi. Maka dari itu, *user interface* dapat disebut sebagai mekanisme integrasi antar *hardware* dan *software* yang membuat pengalaman berkomputer [25].

Pedoman dalam merancang *User interface* menurut para ahli adalah sebagai berikut [6] :

1. *Affordance and visibility*

   Affordance merupakan fungsi elemen dalam antarmuka design yang membuat user mudah berinteraksi dengan sistem. *Visibility* merupakan sejauh mana pengguna dapat mengakses elemen – elemen antarmuka dan opsi yang relevan. *Visiblity* juga dapat diartikan sebagai prinsip dasar dimana apabila sebuah elemen dapat terlihat dengan jelas, maka *user* juga akan memiliki perhatian lebih serta mengetahui apa fungsi dari elemen tersebut.

2. *Consistency*

   Consistency merupakan konsistensi dalam antarmuka design yang merujuk pada penggunaan elemen design yang seragam pada sebuah sistem. Konsistensi sebuah system dapat ditentukan pada pengunaan warna, bentuk ikon, susunan menu, font tulisan. Alur sistem (*flow*) juga merupakan bagian dari *consistency* yang dimana *user* dapat mengerti letak menu, alur sistem, serta fungsi jika dilihat secara sekilas. Contoh *Consistency* adalah jika suatu tombol memiliki fungsi pada suatu halaman, maka tombol tersebut juga harus memiliki fungsi yang sama pada halaman lain.

3. *Shortcut*

   Shortcut digunakan untuk mengurangi jumlah interaksi antara *user* dan *system* agar dapat meringankan tugas *user*. *User* yang sering menggunakan suatu aplikasi berulang kali untuk jangka waktu yang lama menginginkan *shortcut* untuk penggunaan fungsi yang digunakan berulang kali. Agar dapat mengurangi

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

jumlah *action* yang dilakukan maka *shortcut* akan sangat diperlukan untuk menyelesaikan sebuah pekerjaan. Contoh *shortcut* adalah seperti Ctrl + C untuk *copy* dan Ctrl + V untuk *paste*.

4. *Feedback*

Feedback menentukan seberapa responsif sebuah sistem yang dapat dilihat dari proses yang sedang berjalan pada sistem dan pemahaman *user* terhadap sistem yang berjalan tersebut. Apabila *user interface* yang dibuat memiliki *feedback* yang kurang, maka *user* akan kebingungan seperti apakah sistem dapat berjalan dengan baik dan perintah telah terproses. *Feedback* yang baik adalah *feedback* yang intuitif dan dapat memberi tahu *user* apa yang terjadi pada *interface* dan bagaimana cara *interface* tersebut harus digunakan.

Rangkaian tindakan harus disusun berdasarkan urutan awal, tengah, dan akhir. Dialog yang bersifat informatif memberikan *user* rasa puas ketika mereka menyelesaikan serangkaian interaksi. Dialog tersebut berisi apakah interaksi yang dilakukan user sudah benar atau belum dan apa tindakan selanjutnya. Langkah-langkah dalam memandu user melalui interface berguna untuk mencegah kebingungan dan mudah dipahami. Dialog ini memberi tahu bahwa interaksi telah benar dilakukan. Contohnya adalah ketika *user* memberi barang di e-commerce, *user* akan dipandu melalui prosedur dan *feedback* yang diberikan setelah setiap prosedur dilakukan akan memastikan bahwa user berada di langkah yang tepat. Notifikasi yang diberikan seperti pesan pemberitahuan akan mengarahkan mereka ke tujuan yang benar.

5. *Error Handling*

Error Handling sering diabaikan ketika melakukan *design*. *User* dapat merasa frustasi ketika melakukan kesalahan pada interaksi yang dilakukan. Design *User Interface* yang baik harus memiliki antisipasi pada kesalahan umum dan membantu user untuk menghindarinya. Ketika terjadi kesalahan, system harus memberikan informasi mengenai kesalahan tersebut secara spesifik dengan memberikan informasi mengenai apa yang salah dan bagaimana cara memperbaiki kesalahan tersebut. Contoh *Error Handling* adalah ketika *user* ingin melakukan *login* ke sebuah *website* dan *user* memasukkan *username* dan *password* yang salah, maka akan muncul sebuah pemberitahuan bahwa username dan *password* yang dimasukkan salah dan biasanya visual pada elemen tersebut akan berubah agar *user* dapat mengetahui dimana salah mereka.

6. *Easy reversal of actions*

Sistem yang dibuat harus dirancang agar *user* dapat menjelajahi opsi, dan melakukan tindakan yang dapat dibatalkan, atau dikembalikan ke tindakan sebelumnya tanpa adanya kesulitan. Ini merupakan sebuah cara agar user dapat mempelajari dan mengerti bagaimana sistem bekerja. Dengan demikian, maka *user* dapat mengetahui bahwa mereka telah melakukan kesalahan dan dapat mengulang kembali ke tindakan

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

sebelumnya. Perancang *User Interface* harus memastikan bahwa tombol atau menu batal dalam semua dialog sehingga memungkinkan user untuk kembali ke tindakan sebelumnya. Pada *easy reversal of actions,* tindakan yang tersedia untuk user sebaiknya dapat dibalik sebanyak mungkin. Dengan melakukan ini, kita dapat membantu user mengurangi beban mereka ketika terjadi kesalahan atau aktivitas yang dapat dibatalkan. Ini juga mendorong user untuk melakukan eksplorasi opsi yang tidak familiar. Contohnya adalah ketika *user* tidak sengaja menghapus sebuah paragraf pada Microsoft word, maka user dapat melakukan *undo* (CTRL + Z).

7. *Reduce short-term memory load*

User interface yang baik harus menghindari untuk mengharuskan *user* mengingat apapun dari satu kotak dialog ke kotak dialog yang lain selama *user* berinteraksi dengan sistem. Maka *interface* yang dibuat seharusnya membantu *user* melacak dimana terakhir mereka berada seperti memberikan bantuan visual.

Terdapat dua jenis *user interface*, yaitu Command Line *Interface* (CLI) dan Graphical *User Interface* (GUI) [26].

1. *Command Line Interface* (CLI)

   Command Line Interface (CLI) merupakan *User Interface* yang memungkinkan *user* berinteraksi dengan *system* menggunakan teks atau perintah baris. Dalam CLI, *user* memasukkan perintah teks dan menerima output teks sebagai respons. CLI lebih sering digunakan oleh pengguna yang lebih ahli dalam lingkungan teknis atau administrator *system* untuk melakukan tugas - tugas khusus. Dalam lingkungan teknis, CLI memberikan interaksi yang lebih effisien dibandingkan dengan GUI dan memungkinkan *user* untuk membuat *workflow* yang fleksibel atau custom melalui *scripting*.

2. Graphical *User Interface* (GUI)

   Graphical *User Interface* (GUI) merupakan *User Interface* yang menggunakan elemen visual seperti ikon, tombol, dan menu agar *user* dapat berinteraksi dengan *system*. Dalam GUI, *user* dapat berinteraksi dengan *system* melalui perangkat masukan seperti mouse dan keyboard. GUI dirancang untuk memberikan *user* experience yang lebih intuitif dan visual.

   Menurut ahli, GUI merupakan jenis dari computer human *interface* dalam sebuah computer. GUI menyelesaikan masalah layar hitam yang mengganggu *user* awam komputer. Komputer tidak memberikan indikasi mengenai apa yang harus dilakukan oleh *user* pada Langkah selanjutnya untuk mengatasi masalah layer hitam tersebut.GUI merupakan percobaan untuk mengatasi masalah tersebut. GUI yang baik menghilangkan hambatan komunikasi dengan *system* computer dan memungkinkan pengguna untuk langsung bekerja pada masalah yang dihadapi.

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

2.11 User Experience (UX)

User Experience merupakan persepsi dan respons seseorang yang dihasilkan dari penggunaan sebuah produk, *system*, atau jasa. *User* experience adalah konsep yang berfokus pada cara *user* berinteraksi dengan produk yang digunakan. Proses desainnya bertujuan untuk menemukan cara yang paling mudah dan efisien bagi pengguna dalam memakai produk tersebut. *User* Experience mencakup emosi, keyakinan, preferensi, persepsi, respon fisik dan psikologis, dan perilaku *user* yang terjadi sebelum, selama, dan setelah pengunaan [25].

User Experience yang baik diperlukan untuk memastikan kepuasan dan kenyamanan *user*. Berikut merupakan beberapa hal yang membuat *user* experience menjadi baik yaitu [27] :

1. Design yang dibuat disesuaikan dengan kebutuhan dan keinginan *user*
2. Design yang dibuat harus konsisten
3. Design yang dibuat harus sederhana
4. Design yang dibuat harus responsif dan cepat
5. Design yang dibuat harus memudahkan *user* untuk mengakses fitur
6. Design yang dibuat harus menyediakan *user* control
7. Design yang dibuat memiliki *feedback* yang jelas

Ada beberapa fungsi dari *User* Experience yaitu mempermudah *user* dalam menggunakan sebuah aplikasi, menarik minat para *user* agar menggunakan aplikasi secara terus menerus, membangun UI yang bagus, menerapkan UX pada produk digital untuk bisa bersaing dalam bisnis [28].

Untuk membuat sebuah produk yang menyelesaikan masalah pengguna, suatu *interface* tentu harus memiliki UX design yang baik. Berikut merupakan beberapa karakteristik UX yang perlu diperhatikan agar sebuah *interface* memiliki UX design yang baik, yaitu [28] :

1. *Useful*

   Produk yang baik adalah produk yang menyelesaikan masalah *user*. Untuk mengetahui apakah suatu produk sudah berguna atau tidak, designer UI/UX harus mengetahui apakah produk tersebut menyelesaikan masalah *user*, design yang dibuat membantu *user* mencapai tujuan yang diinginkan, design memberikan value pada UX.

2. *Equitable*

   Prorduk yang dibuat harus dapat digunakan oleh individu dengan kemampuan dan latar belakang yang beragam. Contohnya adalah ketika seorang tunarungu membutuhkan caption ketika melihat video.

3. *Usable*

   Produk yang dibuat harus mudah untuk digunakan, agar *user* dapat merasa puas ketika menggunakan produk. Salah satu caranya adalah dengan membuat design, bentuk, dan struktur yang jelas dan efisien.

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

Agar dapat mengetahui apakah sebuah produk usable, maka design yang dibuat harus memudahkan *user* untuk menemukan fitur yang diinginkan, menyelesaikan sebuah interaksi dengan mudah, dan *user* dapat dengan mudah memahami kegunaan dari sebuah fitur.

2.12 *Heuristic Evaluation*

Heuristic Evaluation adalah metode analisis penggunaan yang informal dimana sejumlah evaluator diberikan *interface* design dan diminta untuk memberikan komentar terhadap *interface* tersebut. Metode heuristic ini dikembangkan untuk mengidentifikasi masalah potensial dalam design yang dapat mempengaruhi *user* experience. *Heuristic Evaluation* adalah cara yang efektif untuk mengidentifikasi masalah *design* dengan cepat dan murah sebelum dilakukan *user testing* yang lebih luas [29].

Tujuan utama dari *heuristic evaluation* adalah untuk mendeteksi masalah – masalah pada design dengan efisien. Hasil dari *heuristic evaluation* dapat digunakan oleh tim developer untuk melakukan perbaikan design yang cepat, mengoptimalkan *user interface*, dan meningkatkan keseluruhan kualitas penggunaan *system* [29].

Berikut merupakan sepuluh *Usability Heuristic* yang digunakan untuk merancang *interface* [29]:

1. *Visibility of system status*

   Sistem yang dibuat harus selalu memberikan informasi kepada *user* mengenai apa yang sedang terjadi, melalui feedback yang sesuai pada waktu yang tepat.

2. *Match between system and the real world*

   Design yang dibuat harus dimengerti oleh pengguna. Gunakan kata-kata, frasa, dan konsep yang familiar bagi pengguna. Ikuti konversi dunia nyata, sehingga informasi muncul dalam urutan yang alami dan logis. Hal seperti ini sangat krusial terutama untuk elemen tindakan seperti tombol. Tombol yang dibuat harus selalu jelas kegunaannya agar *user* tidak salah sangka dalam penggunaan elemen tersebut.

3. *User control and freedom*

   User sering sekali dengan tidak sengaja memilih sebuah tindakan. Penting bagi *user* untuk memiliki option undo untuk kembali ke keadaan sebelumnya.

4. *Consistency and standards*

   User tidak seharusnya merasa bingung apakah kata – kata, situasi, atau tindakan yang berbeda memiliki arti yang sama.

5. *Error prevention*

   Error message yang baik sangatlah penting, tapi design yang baik dibuat agar mencegah masalah muncul pada tahapan awal. Baik menghilangkan kondisi yang rentan terhadap kesalahan, atau memeriksa dan menawarkan opsi konfirmasi kepada *user* sebelum mereka melanjutkan ke tindakan tersebut.

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

6. *Recognition rather than recall*

   Meminimalisir beban ingatan dengan membuat elemen, tindakan, dan opsi terlihat. *User* seharusnya tidak perlu untuk mengingat informasi dari satu *interface* ke *interface* lain. Informasi yang diperlukan untuk menggunakan design harus terlihat atau mudah digunakan ketika diperlukan.

7. *Flexibility and efficiency of use*

   *Shortcut*s dapat mempercepat interaksi bagi *user* ahli sehingga design dapat digunakan dengan baik oleh *user* berpengalaman dan tidak berpengalaman. Izinkan *user* untuk menyesuaikan tindakan yang sering dilakukan.

8. *Aesthetic and minimalist design*

   Interface tidak boleh berisi tentang informasi yang tidak relevan atau jarang dibutuhkan. Setiap unit informasi tambahan dalam *interface* bersaing dengan unit informasi yang relevan dan mengurangi visibilitas relatifnya.

9. *Help users recognize, diagnose, and recover from errors*

   Error message harus disampaikan dalam bahasa yang sederhana (tanpa error code), dengan tepat menunjukkan masalah, dan memberikan solusi.

10. *Help and documentation*

    System yang baik tidak memerlukan penjelasan tambahan. Namun, dokumentasi dapat membantu pengguna untuk memahami cara menyelesaikan tugas mereka.

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.