

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Interaksi Manusia dan Komputer**

Interaksi Manusia dan Komputer (IMK) merupakan sebuah ilmu yang mengarah pada kegunaan atau manfaat dari suatu *software*. IMK juga merupakan sebuah ilmu yang digunakan untuk mengetahui kualitas dari suatu hubungan antara manusia dengan teknologi. Sebuah sistem yang menggunakan IMK di dalamnya haruslah didesain untuk menjadi sebuah sistem yang efektif, efisien, menarik, bisa mengatasi masalah, dan mudah untuk dipelajari. Selain itu, sistem yang digunakan juga harus didesain secara simpel dan memiliki sistem navigasi yang mudah untuk dioperasikan. IMK memiliki 6 (enam) aspek yang diperhatikan, yaitu *usability*, *understandability*, *learnability*, *operability*, *attractiveness*, dan *compliance*. Untuk membangun sebuah sistem informasi yang digunakan untuk keperluan bisnis ataupun pribadi, IMK merupakan salah satu ilmu yang wajib untuk diperhatikan agar sistem yang dibangun dapat berjalan sesuai dengan kebutuhan pengguna. IMK merupakan disiplin area yang bersangkutan dengan teori, desain, implementasi, dan evaluasi yang digunakan oleh manusia untuk melakukan interaksi dengan komputer. IMK memiliki beberapa elemen di dalamnya, yaitu [2]:

1. *Usability goal*, dimana elemen IMK ini bertujuan untuk mengetahui seberapa efektif dan efisien sebuah sistem terhadap pengguna yang menggunakannya.
2. *User's experience*, dimana elemen IMK ini bertujuan untuk mengetahui seberapa sesuai sistem yang digunakan dengan kebutuhan yang dibutuhkan.
3. *User model profile*, dimana elemen IMK ini bertujuan untuk mengetahui seberapa besar pengguna mengenali sistem yang baru berdasarkan pengalaman mereka menggunakan sistem yang lama

IMK merupakan bidang antar disiplin ilmu dan masing-masing disiplin ilmu memberikan penekanan pada aspek yang berbeda, seperti misalnya [3]:

1. Psikologi dan ilmu kognitif: membahas tentang teori proses kognitif dan analisis empiris tentang perilaku pengguna.
2. Ergonomik: berhubungan dengan aspek fisik pengguna untuk mendapatkan lingkungan kerja yang nyaman.

3. Sosiologi: membahas tentang interaksi antara teknologi, kerja, dan organisasi.
4. Ilmu komputer: membahas tentang perancangan aplikasi dan rekayasa antarmuka untuk dapat digunakan oleh manusia dengan mudah.
5. Desain grafis: merancang grafik agar dapat digunakan sebagai sarana percakapan yang efektif antara manusia dengan komputer.
6. Dan lain-lain.

Berikut ini merupakan beberapa jenis paradigma IMK [3]:

1. *Time sharing*, yaitu satu komputer yang dapat digunakan oleh banyak pengguna (*multiple user*).
2. *Video Display Unit (VDU)*, yaitu dapat memvisualisasikan dan memanipulasi informasi yang sama dalam representasi yang berbeda serta mampu memvisualisasikan abstraksi data.
3. *Programming toolkits* (alat bantu pemrograman komputer), yang memungkinkan *programmer* meningkatkan kinerja dan produktifitasnya.
4. Komputer pribadi (*personal computing*), yaitu komputer yang digunakan untuk pribadi (pengguna tunggal).
5. Sistem *windows* dan *interface* WIMP (*Windows, Icon, Menu, Pointer*), yaitu sistem *windows* yang memungkinkan pengguna untuk berinteraksi dengan komputer dalam beberapa aktivitas yang berbeda.
6. Metapora (*methapor*), yaitu sistem yang digunakan untuk mengajari konsep baru dimana konsep tersebut sudah dipahami sebelumnya.
7. Manipulasi langsung (*direct manipulation*), yaitu sesuatu yang memungkinkan pengguna untuk mengubah keadaan internal dengan cepat

## 2.2 User Interface (UI)

*User Interface (UI)* merupakan sebuah bentuk tampilan yang berhubungan langsung dengan pengguna dan *user interface* berfungsi untuk menghubungkan antara pengguna dengan sistem operasi sehingga komputer dapat dioperasikan. UI berfokus pada peningkatan produktivitas pengguna dengan menyediakan prosedur entri data yang disederhanakan, tampilan yang mudah dipahami, dan umpan balik informatif yang cepat untuk meningkatkan perasaan kompetensi, penguasaan, dan kontrol terhadap sistem [4].

Berikut ini adalah 8 (delapan) aturan desain antarmuka pengguna, yaitu [4]:

1. Berjuang untuk konsistensi (*strive for consistency*), urutan tindakan yang konsisten harus diperlukan dalam situasi serupa; terminologi identik harus digunakan dalam petunjuk, menu, dan layar bantuan; serta warna yang konsisten, tata letak, kapitalisasi, *font*, dan sebagainya harus digunakan di seluruh desain. Pengecualian seperti konfirmasi yang diperlukan dari perintah *delete* atau tidak ada kata sandi yang digabung harus dapat dipahami dan dibatasi jumlahnya.
2. Mencari kegunaan universal (*seek universal usability*). Mengenali kebutuhan beragam pengguna dan desain untuk plastisitas, memudahkan transformasi konten. Perbedaan pemula hingga ahli, rentang usia, cacat, variasi internasional, dan keragaman teknologi masing-masing memperkaya spektrum persyaratan yang memandu desain. Menambahkan fitur untuk pemula, seperti penjelasan, pakar untuk ahli, seperti pintasan dan tempo yang lebih cepat, memperkaya desain antarmuka, dan meningkatkan kualitas yang dirasakan.
3. Menawarkan umpan balik informatif (*offer informative feedback*). Untuk setiap tindakan pengguna, harus ada umpan balik antarmuka. Untuk tindakan yang sering dan ringan, responnya bisa sederhana, sedangkan untuk tindakan yang jarang dan besar, responnya harus lebih substansial. Presentasi visual dari objek yang menarik memberikan lingkungan yang nyaman untuk menunjukkan perubahan eksplisitas.
4. Dialog desain untuk menghasilkan penutupan (*design dialogs to yield closure*). Urutan tindakan bisa diatur menjadi kelompok-kelompok dengan awal, tengah, dan akhir. Umpan balik informatif pada saat penyelesaian sekelompok tindakan memberi kepuasan pada pemenuhan, rasa lega, sinyal untuk menjatuhkan rencana darurat dari pikiran, dan indikator untuk mempersiapkan kelompok tindakan berikutnya. Misalnya situs *web e-commerce* memindahkan pengguna dari memilih produk ke kasir, diakhiri dengan halaman konfirmasi yang jelas yang menyelesaikan transaksi.
5. Mencegah kesalahan (*prevent errors*). Sebisa mungkin, pengguna desain antarmuka tidak dapat membuat kesalahan serius, misalnya item menu abu-abu yang tidak sesuai dan tidak memungkinkan karakter alfabet dalam bidang entri numerik. Jika pengguna membuat kesalahan, maka antarmuka harus menawarkan instruksi yang sederhana, konstruktif, dan spesifik untuk pemulihan. Misalnya

pengguna tidak perlu mengetik ulang seluruh nama-alamat formulir jika mereka memasukkan kode pos yang tidak valid, tetapi harus dipandu untuk memperbaiki hanya bagian yang salah. Tindakan yang salah harus meninggalkan keadaan antarmuka tidak berubah atau antarmuka harus memberikan instruksi tentang pemulihan.

6. Memungkinkan pembalikan aksi yang mudah (*permit easy reversal of actions*). Sebisa mungkin tindakan harus bisa dikembalikan. Fitur ini mengurangi kecemasan, karena pengguna tahu bahwa kesalahan dapat diurungkan dan mendorong eksplorasi opsi yang tidak dikenal. Unit reversibilitas dapat berupa tindakan tunggal, tugas entri data, atau grup tindakan lengkap, seperti blok nama-alamat.
7. Menjaga pengguna tetap terkendali (*keep users in control*). Pengguna berpengalaman sangat menginginkan pengertian bahwa mereka bertanggung jawab atas antarmuka dan antarmuka menanggapi tindakan mereka. Mereka tidak ingin kejutan atau perubahan dalam perilaku yang dikenal dan mereka terganggu oleh urutan entri data yang membosankan, kesulitan dalam memperoleh informasi yang diperlukan, dan ketidakmampuan untuk menghasilkan hal yang diinginkan.
8. Mengurangi beban memori jangka pendek (*reduce short-term memory load*). Kapasitas manusia yang terbatas untuk pemrosesan informasi dalam memori jangka pendek (aturan dasarnya adalah bahwa orang dapat mengingat tujuh plus atau minus untuk bongkahan informasi) mengharuskan perancang menghindari antarmuka dimana pengguna harus mengingat informasi dari satu tampilan dan kemudian menggunakan informasinya di layar lain. Dengan begitu, ponsel tidak perlu lagi memasukkan kembali nomor telepon, lokasi situs *web* harus tetap terlihat, dan bentuknya harus dipadatkan agar sesuai dengan satu layar.

Pada dasarnya UI terbagi menjadi 2 (dua) jenis, yaitu *Command Line Interface* (CLI) dan *Graphical User Interface* (GUI) [5]:

1. *Command Line Interface* (CLI) merupakan tipe antarmuka dimana pengguna berinteraksi dengan sistem operasi melalui teks-terminal. Pengguna menjalankan perintah dan program di sistem operasi tersebut dengan cara mengetikkan baris-baris tertentu.

2. *Graphical User Interface* (GPU) merupakan tipe antarmuka yang digunakan oleh pengguna untuk berinteraksi dengan sistem operasi melalui gambar-gambar grafik, ikon, menu, dan menggunakan perangkat penunjuk (*pointing device*) seperti *mouse* atau *track ball*. Elemen-elemen utama dari GUI bisa diringkas dalam konsep WIMP (*window, icon, menu, pointing device*).

Beberapa pembagian bentuk dari GUI sebagai berikut [5]:

1. *Head Up Display* (HUD)

HUD adalah suatu tampilan yang transparan dimana menampilkan data tanpa mengharuskan pengguna untuk melihat ke arah yang lain dari sudut pandang biasanya. Asal nama dari alat ini yaitu pengguna dapat melihat informasi dengan kepala yang terangkat (*head up*) dan melihat ke arah depan daripada melihat ke arah bawah bagian instrumen. Walaupun HUD dibuat untuk kepentingan penerbangan militer, sekarang HUD telah digunakan pada penerbangan sipil, kendaraan bermotor, dan aplikasi lainnya.

2. *Tangible User Interface* (TUI)

TUI merupakan suatu antarmuka yang memungkinkan seseorang bisa berinteraksi dengan suatu informasi digital lewat lingkungan fisik. Salah satu perintis TUI ialah Hiroshi Ishii, seorang profesor di Laboratorium Media MIT yang memimpin Tangible Media Group. Pandangan yang istimewa untuk *tangible* UI disebut *tangible bits*, yaitu memberikan bentuk fisik kepada informasi digital sehingga membuat *bit* dapat dimanipulasi dan diamati secara langsung.

3. *Computer Vision*

*Computer Vision* yaitu suatu ilmu pengetahuan dan teknologi dari mesin yang melihat. *Computer Vision* dimanfaatkan juga untuk membangun teori kecerdasan buatan yang membutuhkan informasi dari citra (gambar) yang ditangkap dalam berbagai bentuk, seperti urutan video, pandangan dari kamera yang diambil dari berbagai sudut, dan data multi dimensi yang didapatkan dari hasil pemindaian (*scan*) medis. *Computer vision* juga berusaha untuk mengintegrasikan model dan teori untuk pembangunan sistem visi komputer.

4. *Browsing Audio Data*

*Browsing Audio Data* merupakan metode *browsing* jaringan yang digunakan untuk *browsing* video/audio data yang ditangkap oleh sebuah IP kamera.

### 2.3 User Experience (UX)

UX adalah pengalaman suatu produk atau jasa yang dibuat untuk seseorang yang menggunakan di dunia nyata. Ketika produk atau jasa dikembangkan, orang-orang memberikan banyak perhatian pada produk atau jasa yang dikembangkan, orang-orang memberikan banyak perhatian pada produk atau jasa tersebut. Pengertian pengalaman pengguna atau *user experience* sering diartikan sebagai suatu pencapaian suatu produk atau jasa yang dianggap berhasil atau gagal oleh penggunanya [6].

UX dalam hal pembuatan atau penggunaan suatu produk dapat diartikan pengalaman yang diberikan oleh produk kepada orang yang menggunakan produk tersebut di dalam dunia nyata. Desain dengan menggunakan UX akan memastikan bahwa suatu produk memiliki penampilan yang menarik bagi pengguna dan produk dapat menjalankan fungsi sesuai konteks tujuan pengguna menggunakan produk. Suatu produk yang menggunakan UX sebagai dasar pembuatan produk akan mempertimbangkan tampilan sensor antarmuka yang akan menarik perhatian pengguna, desain antarmuka, navigasi, informasi, dan struktur navigasi, serta arsitektur informasi yang jelas. Selain itu, pembuatan produk juga menerapkan klasifikasi konten yang dapat dipahami pengguna. UX pada suatu mesin pencari akan berfokus kepada bagaimana pengguna dapat dengan mudah dan cepat mendapatkan informasi yang dicari, tentu saja pembuatannya juga harus memperhitungkan setiap desain dan fitur yang akan diberikan sehingga pencarian sebaiknya memiliki navigasi yang mudah dimengerti [7].

UX terdiri dari 4 (empat) elemen yang saling tergantung, yaitu [8]:

1. *Branding*, mencakup semua hal yang terkait dengan estetika dan desain yang ada di dalam *website*.
2. *Usability*, secara umum mensyaratkan kemudahan pengguna terhadap komponen dan fitur yang ada pada sebuah situs.
3. *Functionality*, mencakup semua hal teknik dan proses yang melatarbelakanginya dan aplikasinya.
4. *Content*, mengacu pada konten yang sebenarnya dari sebuah situs (teks, multimedia, dan gambar) serta strukturnya atau arsitektur informasinya.

## 2.4 Usability

*Usability* berasal dari kata *usable* yang secara umum berarti dapat digunakan dengan baik. Sesuatu dapat dikatakan berguna dengan baik apabila kegagalan dalam penggunaannya dapat dihilangkan atau diminimalkan serta memberi manfaat dan kepuasan kepada pengguna. Dalam interaksi antara manusia dengan komputer, usability atau juga disebut “ketergunaan” berkaitan dengan kemudahan dan keterbacaan informasi sekaligus pengalaman navigasi yang *user-friendly*. Pembahasan mengenai *interface* (antarmuka) yang *user-friendly* biasanya digunakan untuk halaman *website* atau perangkat lunak (*software*) agar dapat digunakan secara lebih efisien, mudah, dan memberikan pengalaman yang menyenangkan [9].

Terdapat 5 (lima) unsur yang menjadi pokok usability, yaitu: 1) Kegunaan; 2) Efisiensi; 3) Efektivitas; 4) Kepuasan; dan 5) Aksesibilitas. Dalam perkembangan teknologi media baru berbasis internet, halaman *web* menjadi sentral. Di ruang virtual inilah para pengguna internet berselancar dan mendapatkan pengalaman berinteraksi dengan perangkat teknologi tersebut. Halaman *web* bisa sangat variatif menampilkan informasi sesuai layanan yang diberikan [9].

*Usability* atau “ketergunaan” adalah tingkat kualitas dari sistem yang mudah dipelajari, mudah digunakan, dan mendorong pengguna untuk menggunakan sistem sebagai alat bantu positif dalam menyelesaikan tugas. Dalam konteks ini, yang dimaksud sebagai sistem adalah perangkat lunak. *Usability* dapat juga diartikan sebagai suatu ukuran, dimana pengguna dapat mengakses fungsionalitas dari sebuah sistem dengan efektif, efisien, dan memuaskan dalam mencapai tujuan tertentu. Terdapat banyak definisi *usability* menurut beberapa referensi, baik itu perorangan maupun lembaga [9].

*Usability* memiliki 5 (lima) kriteria komponen, yaitu [9]:

1. *Learnability*, berkaitan dengan seberapa mudah suatu aplikasi atau *website* digunakan. Kemudahan tersebut diukur dari pemakaian fungsi-fungsi dan fitur yang tersedia.
2. *Efficiency*, berkaitan dengan kecepatan dalam pengerjaan “tugas” dalam *website* atau aplikasi perangkat lunak tertentu.

3. *Memorability*, berkaitan dengan kemampuan pengguna mempertahankan pengetahuannya setelah jangka waktu tertentu. Kemampuan tersebut diarahkan oleh tata letak desain *interface* yang relatif tetap.
4. *Errors*, berkaitan dengan kesalahan-kesalahan yang dibuat oleh yang dilakukan oleh pengguna selama berinteraksi dengan *website* atau aplikasi tertentu.
5. *Satisfaction*, berkaitan dengan kepuasan pengguna setelah menggunakan *website* atau aplikasi. Pengukuran terhadap kepuasan juga meliputi aspek manfaat yang didapat dari pengguna selama menggunakan perangkat tertentu.

## 2.5 System Usability Scale (SUS)

SUS adalah salah satu metode uji pengguna yang menyediakan alat ukur yang “*quick and dirty*” yang dapat diandalkan. Metode uji pengguna ini diperkenalkan oleh John Brooke pada tahun 1986 yang dapat digunakan untuk melakukan evaluasi berbagai jenis produk ataupun layanan, termasuk di dalamnya *hardware*, *software*, perangkat mobil, *website*, dan aplikasi [10].

SUS merupakan metode evaluasi kegunaan yang memberikan hasil yang memadai berdasarkan pertimbangan jumlah sampel yang kecil, waktu, dan biaya. Hasil dari perhitungan dengan menggunakan metode SUS akan dikonversi ke dalam sebuah nilai, yang dapat dijadikan pertimbangan untuk menentukan apakah sebuah aplikasi layak atau tidak layak untuk diterapkan [10].

SUS juga pernah digunakan untuk mengetahui sejauh mana faktor pengalaman menggunakan sebuah produk (termasuk *website*) akan dapat mempengaruhi pengguna dalam memberikan nilai SUS yang lebih tinggi. Hasil dari penelitian ini menunjukkan bahwa pengguna *website* yang lebih berpengalaman cenderung memberikan nilai SUS yang lebih tinggi dan lebih menguntungkan daripada pengguna *website* yang belum berpengalaman [10].

Tabel berikut ini menyajikan pertanyaan-pertanyaan yang digunakan pada metode SUS [10].

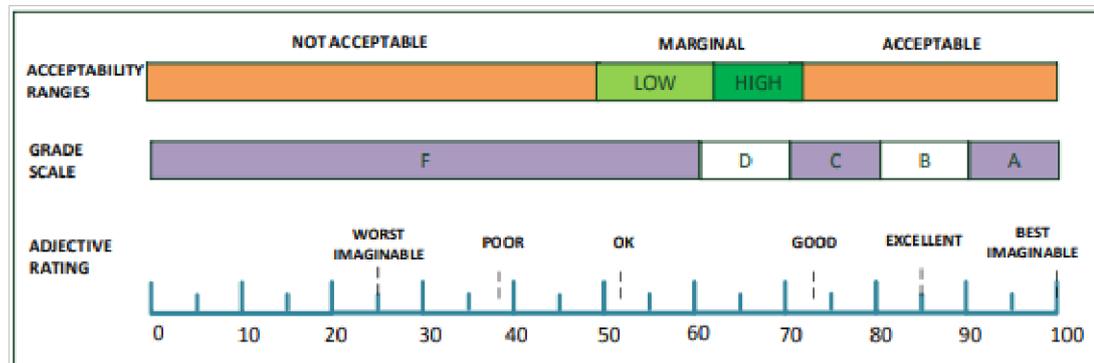
Tabel 2.1 Pertanyaan-Pertanyaan pada *System Usability Scale*

No.	Pertanyaan
R1	Saya sepertinya akan sering menggunakan aplikasi ini
R2	Saya melihat ada bagian fitur aplikasi ini yang cukup merepotkan, yang mestinya hal itu tidak perlu terjadi
R3	Saya rasa aplikasi ini mudah digunakan
R4	Saya sepertinya akan membutuhkan bantuan seorang teknisi agar bisa lancar menggunakan aplikasi ini
R5	Saya rasa fitur-fitur aplikasi ini sudah terintegrasi dengan baik satu sama lain
R6	Saya menemukan terlalu banyak ketidakkonsistenan dalam aplikasi ini
R7	Saya pikir orang-orang akan sangat cepat bisa menggunakan aplikasi ini
R8	Saya rasa aplikasi ini sangat sulit untuk digunakan
R9	Saya merasa mantap menggunakan aplikasi ini
R10	Saya mesti belajar banyak hal terlebih dahulu sebelum mulai menggunakan aplikasi ini

Rumus untuk perhitungan skor SUS adalah sebagai berikut [11].

$$\text{Skor SUS} = ((R1 - 1) + (5 - R2) + (R3 - 1) + (5 - R4) + (R5 - 1) + (5 - R6) + (R7 - 1) + (5 - R8) + (R9 - 1) + (5 - R10)) * 2.5$$

SUS terdiri dari 10 (sepuluh) pertanyaan dengan menggunakan skala *Likert* 1 sampai 5. Pertanyaan nomor ganjil (1, 3, 5, 7, dan 9) merupakan pertanyaan yang bernada positif, sedangkan pertanyaan nomor genap (2, 4, 6, 8, dan 10) merupakan pertanyaan yang bernada negatif seperti yang ditunjukkan pada tabel 2.1. Setiap pertanyaan diberi bobot antara 0 s.d. 4. Pada pertanyaan ganjil (bernada positif), skor tiap pertanyaan dihitung dengan cara bobot tiap pertanyaan ( $x_i$ ) dikurangi 1, sehingga ditulis  $x_i - 1$ . Begitu pula pertanyaan genap (bernada negatif), skor dihitung dengan cara 5 dikurangi bobot tiap pertanyaan ( $x_i$ ) sehingga ditulis menjadi  $5 - x_i$ . Total skor didapatkan dengan menjumlahkan seluruh skor tiap pertanyaan (genap maupun ganjil). Sedangkan skor SUS didapatkan dengan cara mengalikan total skor dengan 2.5. Skor akhir SUS akan berada pada kisaran 0 s.d. 100. Berdasarkan skor akhir SUS tersebut akan bisa diketahui seberapa tinggi tingkat *usability* dan akseptabilitas (*acceptable*) desain sistem aplikasi yang dikembangkan. Penilaiannya berdasarkan pada tiga kategori, yaitu *Not Acceptable* dengan rentang skor SUS 0 s.d. 50.9, *Marginal* 51 s.d. 70.9, dan *Acceptable* 71 s.d. 100. Rentang skor SUS dan pengkategorian dapat dilihat pada gambar berikut ini [11].



Gambar 2.1 Skala *System Usability Scale*

Penjelasan dari gambar di atas [11]:

1. *Acceptable*: Merupakan *rating* sebuah aplikasi yang memenuhi penilaian mencapai 71 s.d. 100, dikatakan sempurna dan sangat layak dipakai.
2. *Marginal*: Merupakan *rating* sebuah aplikasi yang memenuhi penilaian mencapai 51 s.d. 70,9, dikatakan belum sempurna, masih ada yang mesti diperbaiki, tetapi aplikasi yang mencapai level ini masih layak dipakai
3. *Not Acceptable*: Merupakan *rating* sebuah aplikasi yang memenuhi penilaian mencapai 0 s.d. 50,9, dikatakan tidak layak dipakai.

## 2.6 System Flowchart

*Flowchart* adalah teknik analisis bergambar yang digunakan untuk menjelaskan beberapa aspek dari sistem informasi secara jelas, ringkas, dan logis. *Flowchart* mencatat cara proses bisnis dilakukan dan cara dokumen mengalir melalui organisasi. *Flowchart* juga digunakan untuk menganalisis cara meningkatkan proses bisnis dan arus dokumen. Sebagian besar *flowchart* digambarkan dengan menggunakan program perangkat lunak, seperti Microsoft Visio, Microsoft Word, Microsoft Excel, atau Microsoft PowerPoint. *Flowchart* menggunakan seperangkat simbol standar untuk menjelaskan gambaran prosedur pemrosesan transaksi yang digunakan oleh perusahaan dan arus data yang melalui sistem [12].

*System flowchart* merupakan kombinasi dari DFD logis dan fisik, karena memberikan rincian proses yang dilakukan (perspektif logis) serta sumber daya fisik yang digunakan untuk melaksanakannya (perspektif fisik). Namun, *system flowchart* tidak hanya menunjukkan apa prosesnya, seperti halnya DFD logis, atau yang terlibat seperti halnya DFD fisik. Sebaliknya, *system flowchart* memberikan lebih banyak detail

tentang apa yang sebenarnya terjadi di dalam sistem. *System flowchart* menunjukkan apa yang sebenarnya terlibat di dalam pencatatan rekuisisi yang masuk. *System flowchart* juga analog dengan peta proses, dimana akan memberikan lebih banyak detail tentang apa yang sebenarnya terjadi di dalam setiap peta proses. Namun, kontras dengan peta proses, dimana entitas tercantum di bawah margin kiri, sedangkan pada *system flowchart*, entitas di bagian atas dokumen [13].

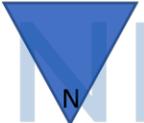
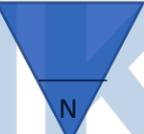
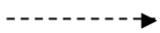
Hal lain yang perlu diperhatikan adalah *system flowchart* memiliki jangkauan simbol yang lebih luas yang dapat digunakan sehingga memungkinkan lebih banyak detail dan wawasan bagi pengguna dokumentasi. Akuntan akan terpapar dengan *system flowchart* serta perlu membaca dan mempersiapkannya ketika dihadapkan dengan tugas untuk mendeskripsikan proses komputerisasi, operasi manual, serta *input* dan *output* dari sistem aplikasi. Auditor menggunakan *system flowchart* untuk mengidentifikasi poin-poin kontrol kunci dalam struktur pengendalian internal sistem akuntansi, karena teknologi menjadi realitas yang terus meningkat dalam kenyataan dalam kegiatan sehari-hari akuntan dan auditor, demikian juga paparan dokumentasi sistem seperti *system flowchart* [13].

Berikut ini adalah simbol-simbol yang biasanya digambarkan pada *system flowchart* [13].

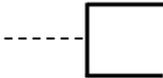
Tabel 2.2 Simbol-Simbol *System Flowchart*

Simbol	Deskripsi
	<i>Start</i> atau <i>Stop</i> , atau entitas eksternal. Menunjukkan awal dan akhir dari suatu proses dan digunakan setiap kali ada sesuatu yang masuk atau ke luar dari sistem yang diinginkan.
	<i>Document</i> – satu dokumen tunggal
	<i>Multiple Document</i> . Kasus ini bisa menjadi tiga salinan dari dokumen yang sama atau ada dokumen berbeda yang dikelompokkan bersama.
	Penyimpanan <i>disk</i> magnetik
	<i>Tape drive</i> atau penyimpanan <i>tape</i> magnetik

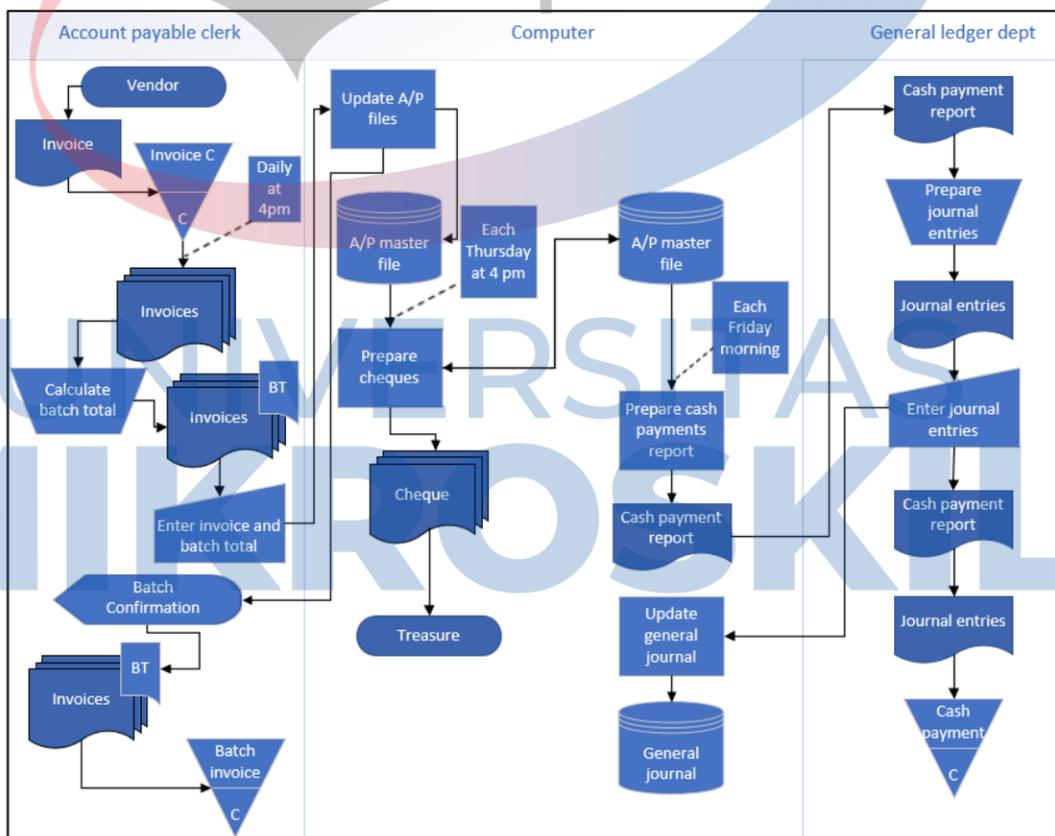
Tabel 2.2 Simbol-Simbol *System Flowchart* (Sambungan)

Simbol	Deskripsi
	<i>Manual Input</i> - input data secara manual ke komputer
	<i>Manual Process</i> - proses yang dilakukan oleh seseorang, misalnya menghitung secara manual berapa banyak faktur dalam tumpukan sebelum memrosesnya atau melengkapi formulir kertas
	<i>Computer Process</i> - proses yang dilakukan secara elektronik. Misalnya penjualan yang dimasukkan ke dalam sistem digunakan untuk memperbarui <i>file</i> piutang dagang
	<i>Offline Process</i> - misalnya data yang dikumpulkan dalam pembaca <i>handheld barcode</i> yang kemudian diunggah ke komputer pusat
	<i>On-page Connector</i> - bergabung dengan dua lokasi berbeda pada halaman yang sama dari <i>flowchart</i> . Dalam hal ini bergabung dengan dua titik berlabel 'A'.
	<i>Off-page Connector</i> - bergabung dengan dua lokasi berbeda pada halaman terpisah dari <i>flowchart</i>
	<i>Punch Card</i>
	<i>Temporary Paper Data Store</i> . Data di toko dapat diajukan secara numerik, menurut abjad, atau kronologis. Ini ditunjukkan dengan 'N', 'A', atau 'C' di dalam simbol penyimpanan data.
	<i>Permanent Paper Data Store</i> . Data di toko dapat diajukan secara numerik, menurut abjad, atau kronologis. Ini ditunjukkan dengan 'N', 'A', atau 'C' di dalam simbol penyimpanan data.
	<i>On-Screen Display</i> - data ditampilkan pada layar komputer atau monitor
	Jurnal umum atau buku besar umum
	Aliran dokumen atau proses
	Aliran data atau informasi
	Pengiriman data antara dua tempat berbeda melalui tautan telekomunikasi

Tabel 2.2 Simbol-Simbol *System Flowchart* (Sambungan)

Simbol	Deskripsi
	Jumlah total
	<i>Annotation</i> - digunakan untuk memberikan deskripsi atau penjelasan dalam <i>flowchart</i>
	Barang fisik atau benda bergerak melalui suatu proses (misalnya barang yang dikirim oleh pemasok atau barang untuk dikirim ke pelanggan)

Berikut ini adalah contoh penggambaran dengan menggunakan *system flowchart* [13].



Gambar 2.2 Contoh Penggambaran *System Flowchart*