

BAB II TINJAUAN PUSTAKA

2.1 Interaksi Manusia dan Komputer

Interaksi manusia dan komputer adalah disiplin ilmu yang berhubungan dengan perancangan, evaluasi, dan implementasi sistem komputer interaktif untuk digunakan oleh manusia, serta studi fenomena-fenomena besar yang berhubungan denganya [1].

Interaksi manusia dan komputer mempunyai peran yang cukup berpengaruh dalam keberhasilan pengembangan implementasi sistem. *User* adalah siapa saja yang terlibat dalam pemrosesan informasi. Komputer didefinisikan sebagai perangkat elektronik yang dapat dipakai untuk mengolah data dengan perantaraan sebuah program yang mampu memberikan informasi dari hasil pengolahan tersebut. Interaksi adalah komunikasi antara *user* dan komputer. Interaksi Manusia dan Komputer adalah studi tentang interaksi antara manusia, komputer, dan tugas (*task*). Bagaimana manusia dan komputer secara interaktif melaksanakan dan menyelesaikan tugas serta bagaimana sistem interaktif tersebut dibuat. Interaksi Manusia dan komputer melibatkan banyak ilmu pengetahuan, misalnya [2]:

1. Psikologi dan ilmu kognitif

Pemahaman akan psikologi orang yang akan menggunakan *software* sangatlah dibutuhkan dalam interaksi manusia dan komputer, mengingat setiap *user* memiliki sifat dan kelakuan yang berbeda.

2. Ergonomik

Ergonomik berhubungan dengan aspek fisik untuk mendapatkan lingkungan kerja yang nyaman.

3. Sosiologi

Pada interaksi manusia dan komputer, sosiologi berhubungan dengan pengaruh sistem komputer dalam struktur sosial. Faktor ini merupakan konteks dari sebuah interaksi.

4. Ilmu komputer dan teknik

Ilmu komputer dan teknik merupakan kemampuan dalam pembuatan suatu produk (*software*).

5. Bisnis

Bisnis mempengaruhi perkembangan interaksi manusia dan komputer. Produk-produk yang dibuat, baik itu produk perusahaan, kelompok, atau perorangan.

6. Desain Grafis

Konteks multimedia digunakan sebagai sarana dialog yang sangat efektif antara manusia dan komputer. Dengan konteks multimedia, tampilan dari interaksi manusia dan komputer akan menjadi lebih menarik dan lebih mudah dimengerti oleh pengguna.

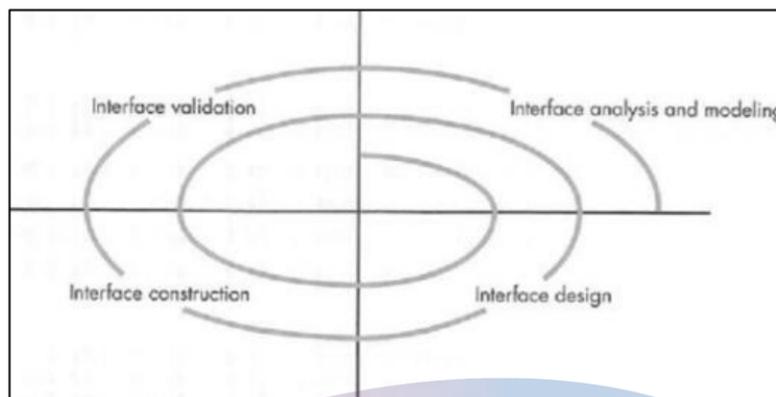
2.2 *User Interface (UI)*

Sebuah sistem komputer memiliki tiga aspek, yaitu perangkat keras (*hardware*), perangkat lunak (*software*), dan manusia (*brainware*) yang saling terkait dan berhubungan. *User interface* atau antarmuka pengguna menggunakan bentuk tampilan grafis yang berhubungan langsung dengan pengguna. *Interface* adalah sebuah tempat dimana interaksi antara pengguna dan sistem operasi efektif serta umpan balik dari sistem operasi yang membuat keputusan operasional [3].

User Interface (UI) adalah bagian dari komputer dan perangkat lunak dimana orang bisa melihat, mendengar, menyentuh, berbicara, atau mengerti. UI pada intinya memiliki dua komponen, yaitu *input* dan *output*. *Input* adalah cara seseorang menyampaikan kebutuhannya atau keinginannya kepada komputer. Perangkat *input* yang umum adalah *keyboard* dan *mouse*. *Output* adalah cara komputer menyatakan hasil dari perhitungan dan kebutuhan pengguna [3].

Antarmuka (*user interface*) sangat diperlukan dalam suatu pembuatan sistem. Dengan adanya antarmuka diharapkan pengguna akan mudah memahami apa yang harus dilakukan dan menghindari terjadinya kebingungan saat menggunakan sistem. Desain *user interface* menciptakan sebuah media komunikasi yang efektif antara manusia dengan komputer [4].

Gambar berikut ini menunjukkan proses-proses dalam pembuatan *user interface* [4].



Gambar 2.1 Proses *User Interface*

Terdapat langkah-langkah dalam pembuatan antarmuka (*user interface*):

1. *Interface analysis and modeling*

Interface analysis berfokus pada *user* untuk berinteraksi dengan sistem, kemudian dianalisis untuk mendefinisikan satu set objek dan aksi *interface*. Informasi yang dikumpulkan digunakan untuk membuat model analisis untuk *interface*.

2. *Interface design*

Interface design mendefinisikan satu set objek dan aksi *interface* yang memungkinkan *user* untuk melakukan semua tugas desain tata letak.

3. *Interface construction*

Interface construction biasanya dimulai dengan membuat sebuah prototipe yang memungkinkan skenario penggunaan untuk dievaluasi dan digunakan untuk menyelesaikan konstruksi *interface*.

4. *Interface validation*

Setelah pembuatan prototipe, *interface validation* berfokus pada evaluasi secara keseluruhan untuk menentukan kemampuan *interface* untuk menjalankan setiap perintah, apakah berjalan dengan benar, dan sejauh mana *interface* mudah digunakan dan mudah dipelajari, serta memenuhi kebutuhan *user*.

Dalam perancangan *user interface* harus memenuhi delapan aturan emas. Adapun delapan aturan emas tersebut adalah sebagai berikut [5]:

1. Berusaha untuk konsisten

Rangkaian aksi yang konsisten harus digunakan dalam keadaan, seperti pada *prompts*, *menus*, dan layar *help*, serta perintah yang konsisten.

2. Menyediakan *universal usability*

Universal usability mengacu pada desain informasi dan komunikasi produk serta layanan yang dapat digunakan oleh semua kalangan *user*.

3. Memberikan umpan balik yang informatif

Untuk setiap aksi yang dilakukan, hendaknya selalu tersedia fasilitas umpan balik agar pengguna mengerti apa yang telah dilakukannya.

4. Merancang dialog yang memberikan penutupan

Urutan aksi hendaknya dibagi ke dalam kelompok dengan awal, tengah, dan akhir. Ketika telah mencapai bagian akhir, hendaknya *user* diberitahu melalui umpan balik. Tanpa adanya dialog untuk mencapai keadaan akhir, maka pengguna akan menjadi bingung.

5. Menawarkan penanganan kesalahan sederhana

Sebisa mungkin, desainlah sistem sehingga pengguna tidak dapat melakukan kesalahan yang serius. Jika kesalahan dibuat, maka sistem harus mampu mendeteksi kesalahan dan membantu memberikan solusi untuk penanganan kesalahan.

6. Memungkinkan pembalikan aksi yang mudah

Fitur ini mengurangi kecemasan, karena *user* tahu bahwa kesalahan dapat dibatalkan, sehingga akan mendorong eksplorasi fungsi-fungsi lainnya.

7. Mendukung pusat kendali internal

Dengan pengaturan yang menyeluruh, *user* dapat menggunakan sistem sesuai kebutuhan mereka dan menggunakan sistem lebih maksimal.

8. Mengurangi beban ingatan jangka pendek

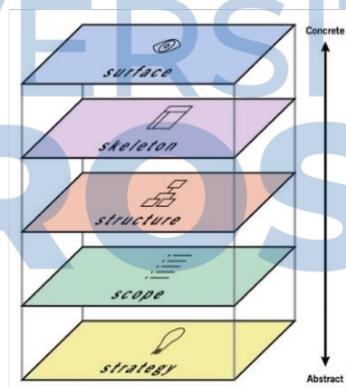
Keterbatasan manusia dalam mengolah informasi dalam jangka waktu yang pendek harus diperhatikan dalam membuat tampilan sehingga tidak menyulitkan *user*.

2.3 User Experience (UX)

User Experience (UX) merupakan bagaimana layanan atau produk bekerja ketika seseorang datang atau menggunakannya, bukan mengenai kinerja internalnya. Untuk itu, UX merupakan bagian yang penting dalam berbisnis. UX memiliki lima elemen dasar, yaitu [6]:

1. *Strategy plane*, merupakan lapisan paling bawah yang berupa strategi bisnis mendasar yang menentukan tujuan dari produk dan kebutuhan penggunanya.
2. *Scope plane*, merupakan translasi dari strategi untuk diteruskan dalam pembuatan spesifikasi fungsional.
3. *Structure plane*, dibentuk dari arsitektur informasi yang disusun untuk memfasilitasi pemahaman pengguna.
4. *Skeleton plane*, terdiri atas tiga bagian, meliputi:
 - a. *Information design*, merupakan presentasi dari informasi untuk memudahkan pemahaman.
 - b. *Interface design*, merupakan pengaturan elemen antarmuka untuk memungkinkan pengguna dalam berinteraksi.
 - c. *Navigation design*, merupakan kumpulan elemen yang memungkinkan pengguna untuk menjelajahi arsitektur informasi.
5. *Surface plane*, merupakan pengalaman sensorik ke pengguna, meliputi warna, gambar, dan ikon yang dibuat untuk menyelesaikan produk.

Kelima elemen tersebut merupakan urutan dari lapisan paling bawah (mendasar dan abstrak) hingga lapisan paling atas (visual). Lapisan tersebut ditunjukkan pada gambar berikut ini [6].



Gambar 2.2 Lima Elemen Dasar UX

2.4 Usability

Usability adalah analisis kualitatif yang menentukan seberapa mudah *user* menggunakan antarmuka suatu aplikasi. Suatu aplikasi disebut *usable* jika fungsi-fungsinya dapat dijalankan secara efektif, efisien, dan memuaskan. Efektifitas

berhubungan dengan keberhasilan pengguna mencapai tujuan dalam menggunakan suatu perangkat lunak. Efisien berkenaan dengan kelancaran pengguna terhadap perangkat lunak. Pengujian *usability* dilakukan untuk mengevaluasi apakah sebuah aplikasi sudah sesuai dengan kebutuhan pengguna atau belum [7].

Usability testing (uji kegunaan) adalah mengukur efisiensi, kemudahan dalam mempelajari, dan kemampuan untuk mengingat bagaimana berinteraksi tanpa mengalami kesulitan. Sejak mulai berkembangnya internet, para pakar dalam bidang uji kegunaan menekankan uji kegunaan dengan dua hal penting, yaitu [7]:

1. *Ease of Learning*

Mengukur ketergunaan dengan membandingkan waktu yang digunakan dalam mempelajari sistem komputer yang belum pernah dikenalnya sama sekali, dengan waktu yang diperlukan untuk melakukan hal yang sama dengan cara lain.

2. *Ease of use*

Mengukur jumlah tindakan yang digunakan untuk menyelesaikan suatu pekerjaan. Misalnya membandingkan jumlah klik *mouse* pada dua desain.

Usability sangat berguna untuk keberlangsungan dari *website* dan aplikasi mobil. *Website* dan aplikasi mobil harus dirancang sebaik mungkin, sehingga memudahkan pengguna untuk menggunakan *website* atau aplikasi mobil dan dapat mengunjungi kembali dengan mudah *website* atau aplikasi mobil tersebut. Uji kegunaan merupakan kombinasi dari lima aspek, yaitu [7]:

1. *Ease of learning* (mudah dipelajari)
2. *Efficiency of use* (efisien dalam penggunaan)
3. *Memorability* (mudah diingat)
4. *Error frequency and severity* (frekuensi kesalahan dan kesederhanaan)
5. *Subjective satisfaction* (kepuasan pengguna)

2.5 System Usability Scale (SUS)

System Usability Scale (SUS) merupakan metode pengujian *usability* suatu sistem secara sederhana dengan sepuluh skala yang memberikan pandangan secara menyeluruh dari evaluasi tujuan kebergunaan. SUS berupa skala Likert yang sederhana dengan responden diharuskan menjawab tingkat kesetujuan dan

ketidaksetujuan dengan skala 5 (lima). SUS dapat dipercaya, skala *usability* dengan biaya rendah yang dapat digunakan untuk pengujian *system usability* secara global [9].

SUS menghasilkan satu nomor yang mewakili ukuran gabungan dari kegunaan keseluruhan dari sistem yang dipelajari. Perhatikan bahwa skor untuk setiap item yang tidak bermakna pada mereka sendiri. Untuk menghitung skor SUS, pertama jumlahkan kontribusi skor dari setiap item. Setiap item kontribusi skor akan berkisar dari 0 (nol) sampai 4 (empat). Untuk item 1, 3, 5, 7, dan 9, kontribusi skor adalah skala posisi dikurangi 1 (satu). Untuk item 2, 4, 6, 8, dan 10, kontribusi adalah 5 (lima) minus posisi skala. Kalikan jumlah nilai sebesar 2,5 untuk mendapatkan nilai keseluruhan SUS. Skor SUS memiliki rentang 0 sampai 100 [9].

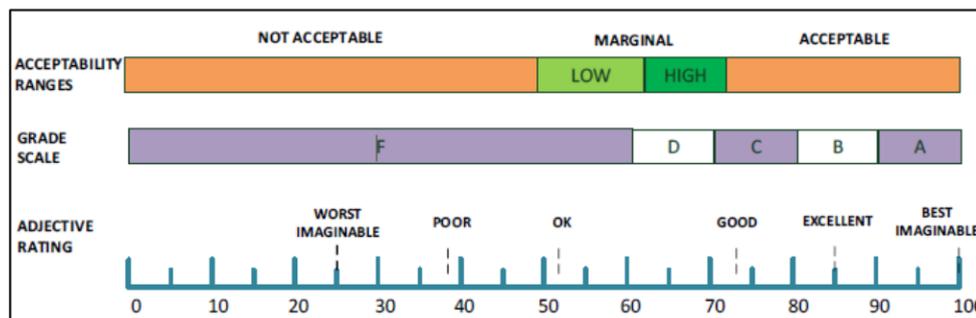
Rumus untuk perhitungan skor SUS adalah sebagai berikut [9].

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

Ketika SUS digunakan, peserta diminta untuk menilai 10 (sepuluh) item berikut dengan satu dari lima tanggapan yang berkisar dari Sangat Setuju sampai Sangat Tidak Setuju [9]:

1. Saya pikir saya ingin sering menggunakan sistem ini.
2. Saya menemukan sistem yang tidak perlu rumit.
3. Saya pikir sistemnya mudah digunakan.
4. Saya pikir saya akan membutuhkan dukungan dari orang teknis untuk dapat menggunakan sistem ini.
5. Saya menemukan berbagai fungsi dalam sistem ini terintegrasi dengan baik.
6. Saya pikir ada terlalu banyak inkonsistensi dalam sistem ini.
7. Saya akan membayangkan bahwa kebanyakan orang akan belajar menggunakan sistem ini dengan sangat cepat.
8. Saya menemukan sistem yang sangat rumit untuk digunakan.
9. Saya merasa sangat yakin menggunakan sistem ini.
10. Saya perlu belajar banyak hal sebelum saya dapat menggunakan sistem ini.

Penentuan yang kedua dilihat dari sisi *percentile range* (SUS skor) yang memiliki *grade* penilaian yang terdiri dari A, B, C, D, dan F. Dari kedua cara penentuan hasil penilaian tersebut, maka dapat dilihat hasil penilaian seperti gambar berikut ini [9].



Gambar 2.3 Penentuan Hasil Penilaian

Penjelasan dari gambar di atas sebagai berikut [9]:

1. *Acceptability Ranges, Grade Scale, Adjective Rating*: Penentuan *acceptability ranges, grade scale, dan adjective rating* digunakan untuk melihat sejauh mana perspektif pengguna terhadap perangkat lunak.
2. *SUS Skor Percentile Rank*: Penentuan hasil penilaian dengan cara *SUS skor percentile rank* memiliki perbedaan dengan cara penilaian *acceptability, grade scale, dan adjective rating*. Perbedaan terjadi pada kategori penilaian, dimana pada *SUS skor percentile rank* dilakukan perbandingan hasil penilaian pengguna secara umum, sedangkan pada *acceptability, grade scale, dan adjective rating* dibedakan ke dalam tiga kategori. Untuk itu, dalam menentukan *SUS skor percentile rank* seperti ketentuan sebagai berikut:
 - a. *Grade A*: dengan skor lebih besar atau sama dengan 80,3
 - b. *Grade B*: dengan skor lebih besar sama dengan 74 dan lebih kecil 80,3
 - c. *Grade C*: dengan skor lebih besar sama dengan 68 dan lebih kecil 74
 - d. *Grade D*: dengan skor lebih besar sama dengan 51 dan lebih kecil 68
 - e. *Grade F*: dengan skor lebih kecil dari 51

Pengertian dari :

1. *Acceptability Range* adalah sejauh mana perspektif pengguna terhadap perangkat lunak yang penilaiannya pada *SUS Score* mulai dari *Not Acceptable, Marginal Low & Marginal Hight, dan Acceptable*.
2. *Grade Scale* adalah sejauh mana pengguna terhadap *system* atau perangkat yang penilaiannya pada *SUS Score* berdasarkan grade (A,B,C,D,F)
3. *Adjective Rating* adalah penilaian pengguna terhadap *system* atau perangkat pada *SUS Score* yang penilaiannya berdasarkan rating mulai dari 0 s.d 100

2.6 System Flowchart

Secara konseptual, *system flowchart* merepresentasikan sebuah kombinasi DFD logis dan fisik, karena menyajikan rincian dari proses yang ditampilkan (perspektif logis) serta sumber daya fisik yang digunakan untuk menampilkannya (perspektif fisik). Namun, *system flowchart* tidak hanya menunjukkan apa prosesnya, seperti halnya DFD logis, atau siapa yang terlibat, seperti halnya DFD fisik. Sebaliknya, *system flowchart* memberikan lebih banyak detail tentang apa yang sebenarnya terjadi di dalam sistem. Bahkan menunjukkan bahwa ini sebenarnya hanya termasuk dalam pencatatan rekuisisi yang masuk. *System flowchart* juga sejalan dengan *process map* yang akan memberikan lebih banyak detail tentang apa yang sebenarnya terjadi di dalam setiap proses yang muncul dalam *process map*. Namun, berbeda dengan *process map*, dimana entitas tercantum di bawah margin kiri, sedangkan entitas *system flowchart* dibuat terdaftar di bagian atas dokumen [10].

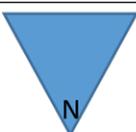
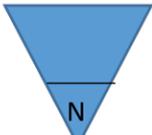
Hal lain yang perlu diperhatikan dalam perbandingan antara *process map* dan DFD, *system flowchart* memiliki simbol dengan jangkauan yang lebih luas yang dapat digunakan, sehingga memungkinkan lebih banyak detail dan pemahaman bagi penggunaan dokumentasi. Pengguna (akuntan) yang akan memaparkan *system flowchart* perlu mempersiapkan ketika dihadapkan dengan tugas yang harus menguraikan proses yang dikomputerisasi, operasi manual, serta *input* dan *output* dari sistem aplikasi. Auditor menggunakan *system flowchart* untuk mengidentifikasi poin-poin kontrol utama dalam struktur pengendalian internal sistem akuntansi. Karena teknologi menjadi realitas yang semakin meningkat dalam kegiatan sehari-hari akuntan dan auditor, demikian juga paparan dokumentasi sistem seperti *system flowchart* [10].

Terdapat beberapa simbol yang biasanya digunakan dalam menggambarkan *system flowchart*, dengan deskripsi singkat dari simbol-simbol tersebut, seperti ditunjukkan pada tabel berikut ini [10].

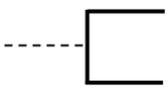
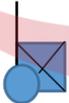
Tabel 2.1 Simbol-Simbol *System Flowchart*

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.

Tabel 2.1 Simbol-Simbol *System Flowchart* (Sambungan)

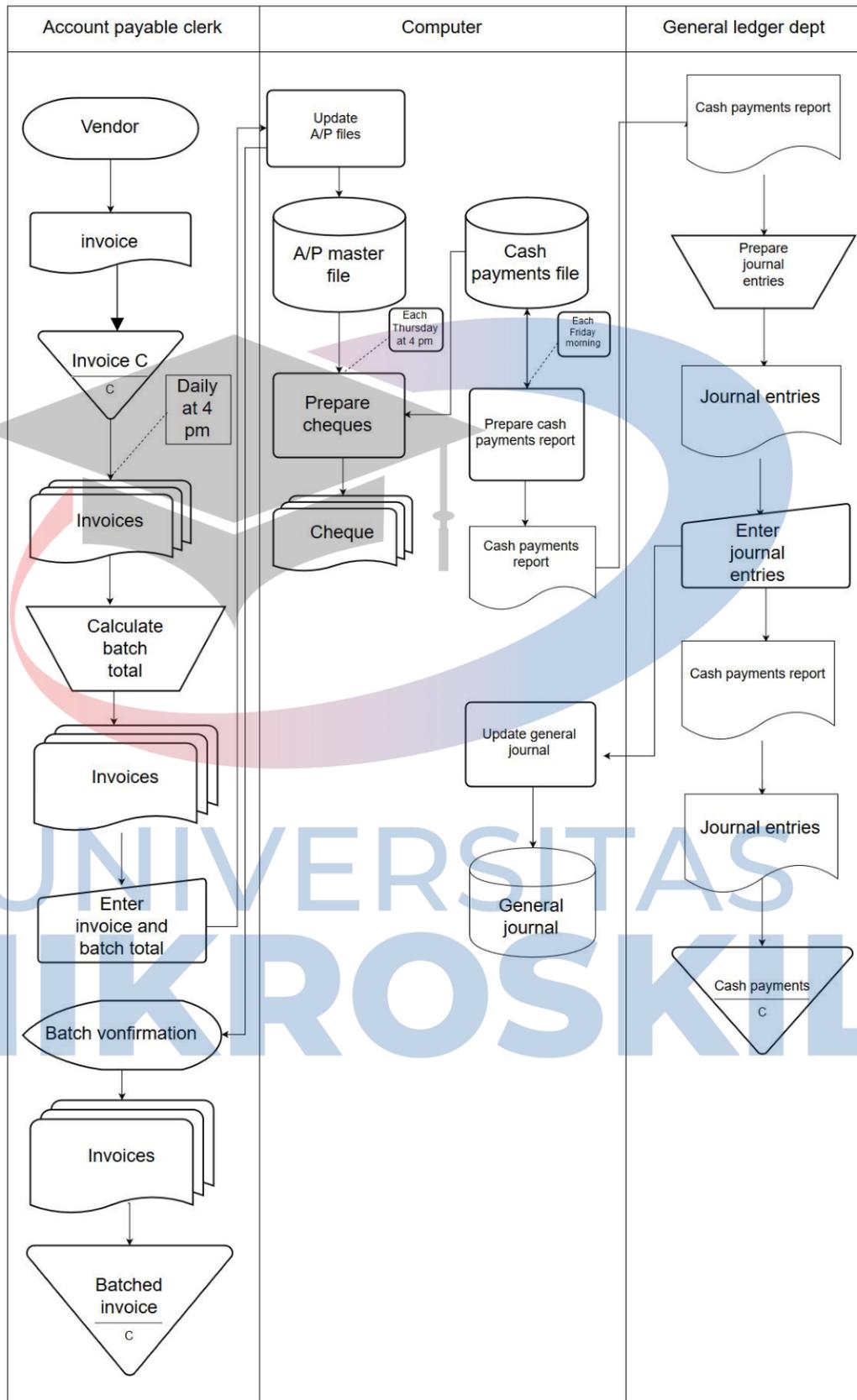
Simbol	Deskripsi
	<i>Document</i> – satu dokumen tunggal
	<i>Multiple Document</i> . Kasus ini bisa menjadi tiga salinan dari dokumen yang sama atau ada dokumen berbeda yang dikelompokkan bersama.
	Penyimpanan <i>disk</i> magnetik
	<i>Tape drive</i> atau penyimpanan <i>tape</i> magnetik
	<i>Manual Input</i> - <i>input</i> 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.

Tabel 2.1 Simbol-Simbol *System Flowchart* (Sambungan)

Simbol	Deskripsi
	<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
	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)

Ketika membaca *system flowchart*, biasanya akan dimulai dari sudut kiri atas dokumen dan dilanjutkan ke pojok kanan bawah. Awal *system flowchart* biasanya ditunjukkan oleh nama entitas eksternal. Juga dapat dilihat bahwa *system flowchart* memiliki beberapa kolom. Setiap kolom sesuai dengan entitas internal yang merupakan bagian dari proses yang didokumentasikan. Dengan demikian, salah satu hal yang harus diperhatikan ketika menggambar *system flowchart* dengan membandingkannya dengan diagram alir data fisik adalah bahwa jumlah kolom entitas internal dan pelabelan entitas internal dalam DFD fisik [10].

Gambar berikut ini menunjukkan contoh penggambaran dengan menggunakan *system flowchart* [10].



Gambar 2.4 Contoh Penggambaran System Flowchar