

## BAB II TINJAUAN PUSTAKA

### 2.1 Interaksi Manusia dan Komputer (IMK)

Interaksi Manusia dan Komputer (IMK) adalah sebuah disiplin ilmu yang mempelajari perancangan, implementasi, dan evaluasi sistem komputasi interaktif, serta berbagai aspek terkait. Istilah “interaksi manusia dan komputer” atau “interaksi manusia dan mesin” melingkupi dua sisi, yaitu mesin dan manusia. Istilah “mesin” lebih populer dengan sebutan komputer. Berbagai jenis komputer antara lain adalah *mainframe*, *workstation*, dan komputer pribadi. Sedangkan istilah “manusia” sebagai sekelompok orang atau organisasi yang didalamnya antara lain sistem terdistribusi, komunikasi antar manusia berbantuan komputer, atau suatu pekerjaan yang secara kooperatif dikerjakan oleh sekelompok orang yang menggunakan bantuan sistem komputer. Model interaksi membantu untuk mengerti apa yang terjadi diantara pengguna dan sistem, menerjemahkan tujuan, antara apa yang diinginkan *user*, dan apa yang harus dikerjakan sistem. Dialog antara manusia dan sistem dipengaruhi oleh bentuk *interface*. Interaksi mengambil tempat antara sosial dan organisasi *framework* yang mempengaruhi *user* dan sistem. Untuk melakukan *task* (tugas), maka *user* harus memenuhi persyaratan untuk berkomunikasi dengan sistem[2].

Bagaimana manusia dan komputer secara interaktif melaksanakan dan menyelesaikan tugas (*task*) dan bagaimana sistem yang interaktif itu dibuat. IMK melibatkan banyak bidang ilmu pengetahuan, seperti misalnya [3]:

1. Psikologi dan ilmu kognitif: persepsi *user*, kognitif, kemampuan memecahkan masalah.
2. Ergonomi: Kemampuan fisik *user*.
3. Sosiologi: Kemampuan memahami konsep interaksi.
4. Ilmu komputer dan teknik: Membuat teknologi.
5. Bisnis: Pemasaran.
6. Desain grafis: Presentasi *interface*.
7. Dan lain-lain.

Secara tradisional, ergonomi memfokuskan pada karakteristik fisik mesin dan sistem serta melihat unjuk kerja (*performance*) dari *user*. Keberhasilan dalam IMK

akan menentukan apakah sistem yang akan dikembangkan bersifat *user-friendly* atau tidak. IMK dalam konteks kerja dan tugas *user* melibatkan desain, implementasi, dan evaluasi. Berikut ini beberapa jenis paradigma IMK [3]:

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

## 2.2 User Interface

*User Interface* adalah bentuk tampilan yang berhubungan dengan *user*. Selain itu *user interface* sendiri bisa diartikan sebagai penghubung antara *user* dengan sebuah sistem operasi. *User interface* merupakan salah satu unsur yang paling penting dari sebuah sistem. Jika sebuah *user interface* dirancang sederhana, maka akan memperlambat *user* dalam mengoperasikan sistem operasi. Bahkan *user interface* yang lemah dapat menyebabkan kegagalan pada suatu sistem. Karena itu, perancangan *user interface* harus dibuat sebaik mungkin. Proses perancangan *user interface* dibuat dengan pembuatan model terlebih dahulu, kemudian digambarkan. Hasilnya adalah sebuah prototipe desain yang kemudian akan dievaluasi untuk

menguji kualitasnya. Dalam perancangan *user interface*, harus memperhatikan *user* dan konsep pengerjaannya, baik itu pemahaman tentang karakteristik dan perilaku dari *user* yang umum serta populasi dari *user*[4].

Berikut ini adalah kriteria sistem *interface* yang baik[3]:

1. *User interface* lebih dari apa yang manusia dapat lihat, sentuh, dan dengar.
2. *User interface* mencakup konsep, kebutuhan *user* untuk mengetahui sistem komputer, dan harus dibuat terintegrasi ke seluruh sistem.
3. *User interface* tidak cukup hanya berpenampilan bagus, tetapi harus dapat mendukung tugas yang dilakukan oleh manusia dan dibuat agar bisa menghindari kesalahan-kesalahan kecil.

Dalam desain sistem berdasarkan *user interface*, terdapat delapan aturan emas dalam membuat sebuah tampilan *user interface*. Delapan aturan tersebut dapat digunakan dalam sebuah sistem yang interaktif dan membutuhkan sebuah validasi dan pengaturan untuk sebuah tampilan domain yang spesifik[5].

Tabel 2.1 Delapan Aturan Emas Desain *User Interface*

Tahapan	Penjelasan
<i>Strive for consistency</i>	Konsistensi dilakukan pada urutan tindakan, perintah, dan istilah yang digunakan pada <i>prompt</i> , menu, serta layar bantuan. Contohnya konsisten formulir dalam hal alur <i>form</i> , warna, menu, tampilan, tulisan, dan masih banyak lainnya.
<i>Center to universal usability</i>	Memungkinkan pengguna untuk menggunakan <i>shortcut</i> . Ada kebutuhan dari pengguna yang telah ahli untuk meningkatkan kecepatan interaksi, sehingga diperlukan singkatan, tombol fungsi, perintah tersembunyi, dan fasilitas makro.
<i>Offer information feedback</i>	Untuk setiap tindakan operator, sebaiknya disertakan suatu sistem umpan balik. Untuk tindakan yang sering dilakukan dan tidak terlalu penting dapat diberikan umpan balik yang sederhana. Tetapi, ketika tindakan merupakan hal yang penting, maka umpan balik sebaiknya lebih substansi. Misalnya, muncul suara ketika salah menekan tombol pada waktu <i>input</i> data atau muncul pesan kesalahannya.
<i>Design dialogs to yield closer</i>	Urutan tindakan sebaiknya diorganisir dalam suatu kelompok dengan bagian awal dan akhir. Umpan balik yang informatif akan memberikan indikasi bahwa cara yang dilakukan telah benar dan dapat mempersepsikan kelompok tindakan berikutnya.
<i>Prevent errors</i>	Sedapat mungkin sistem dirancang sehingga pengguna tidak dapat melakukan kesalahan fatal. Jika kesalahan terjadi, maka sistem dapat mendeteksi kesalahan dengan cepat dan memberikan mekanisme yang sederhana dan mudah dipahami untuk penanganan

Tabel 2.1 Delapan Aturan Emas Desain *User Interface* (Sambungan)

Tahapan	Penjelasan
	kesalahan.
<i>Permit easy reversal of action</i>	Hal ini mengurangi kekhawatiran pengguna karena pengguna mengetahui kesalahan yang dilakukan dapat dibatalkan, sehingga pengguna tidak takut untuk mengeksplorasi pilihan-pilihan lain yang belum bisa digunakan.
<i>Support internal locus of control</i>	Pengguna ingin menjadi pengontrol sistem dan sistem akan merespon tindakan yang dilakukan pengguna daripada pengguna merasa bahwa sistem mengontrol pengguna. Sebaiknya sistem dirancang sedemikian rupa sehingga pengguna menjadi inisiator daripada responden.
<i>Reduce short-term memory load</i>	Keterbatasan ingatan manusia membutuhkan tampilan yang sederhana atau banyak tampilan halaman yang sebaiknya disatukan, serta diberikan cukup waktu pelatihan untuk kode, <i>mnemonic</i> , dan urutan tindakan.

### 2.3 User Experience

*User experience* 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 *User Experience* akan memastikan bahwa suatu produk memiliki penampilan yang menarik bagi pengguna dan produk dapat menjalankan fungsi sesuai konteks tujuan pengguna yang menggunakan produk. Suatu produk yang menggunakan *User Experience* 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. *User experience* 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[6].

Istilah *user experience* menempatkan pengguna akhir (*end user*) sebagai titik fokus dalam pengembangan sebuah desain dan pengembangan sebuah antarmuka, yang bertentangan dengan sistem, aplikasi, atau nilai estetikanya sendiri. Hal ini

menjadikan persepsi pengguna sebagai pertimbangan penting dalam pengembangan dan pengukuran *user experience*, apakah sistem informasi tersebut diterima atau tidak oleh pengguna. Keberhasilan *user experience* mengacu pada pengguna akhir, hal ini terkait dengan kemudahan pengguna ketika menggunakan sebuah sistem, bagaimana pengguna merasakan dan termasuk persepsi pengguna dalam menemukan atau menangani hal-hal praktis yang disediakan oleh sistem. Dengan pengalaman pengguna, seorang *user* dapat merasakan kemudahan dan efisiensi dalam berinteraksi dengan sebuah sistem. Hal ini termasuk persepsi seseorang terhadap aspek praktis, seperti kegunaan, kemudahan penggunaan, dan efisiensi dari sistem yang digunakan. Keberhasilan *user experience* mengacu pada pengguna akhir, hal ini terkait dengan kemudahan pengguna ketika menggunakan sebuah sistem, bagaimana pengguna merasakan, dan termasuk persepsi pengguna dalam menemukan atau menangani hal-hal praktis yang disediakan oleh sistem. Beberapa faktor yang mempengaruhi *user experience*[7]:

1. Kesesuaian desain situs/*website* dengan keinginan/tujuan awal;
2. Kemampuan dan keterbatasan situs.
3. Isi dan tampilan situs.
4. Fungsionalitas situs.

#### 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. *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. Berikut ini beberapa definisi *usability*[8]:

1. Jakob Nielsen: Mendefinisikan *usability* sebagai ukuran kualitas pengalaman pengguna ketika berinteraksi dengan produk atau sistem, apakah situs *web*, aplikasi perangkat lunak, teknologi bergerak, maupun peralatan-peralatan lain yang dioperasikan oleh pengguna.
2. International Organization for Standardization (ISO): Mendefinisikan *usability* sebagai tingkat dimana produk bisa digunakan oleh pengguna tertentu untuk mencapai tujuannya dengan lebih efektif, efisien, dan memuaskan dalam ruang lingkup penggunaannya.

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. Terdapat lima unsur yang menjadi pokok usability, yaitu Kegunaan, Efisiensi, Efektivitas, Kepuasan, dan 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 mereka berikan[8].

Untuk mengukur *usability* bergantung pada kemampuan pengguna menyelesaikan serangkaian tes. Beberapa parameter untuk mengukur *usability* meliputi[8]:

1. *Success Rate*, mengukur tingkat keberhasilan pengguna dalam menyelesaikan semua “tugas” yang ada pada suatu *website*.
2. *The Time a Task Requires*, mengukur waktu yang dibutuhkan oleh seorang pengguna dalam menyelesaikan suatu “tugas” pada *website* tersebut.
3. *Error Rate*, tingkat kesalahan yang dilakukan oleh pengguna pada saat menyelesaikan “tugas” pada *website* tersebut.
4. *User’s Subjective Satisfaction*, tingkat kepuasan pengguna dalam menyelesaikan keseluruhan “tugas” ketika berinteraksi dalam *website* tersebut.

Serangkaian tes tersebut secara umum merujuk pada lima kriteria usability yang meliputi: *Learnability*, *Efficiency*, *Memorability*, *Errors*, dan *Satisfaction*[8, 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 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.

Ukuran *usability* harus mencakup tiga aspek (ISO,1998) sebagai berikut [1]:

1. Efektivitas menunjukkan tingkat akurasi dan kesempurnaan yang dicapai pengguna saat menjalankan tugas tertentu.
2. Efisiensi menunjukkan sumber daya yang digunakan terkait dengan akurasi dan kesempurnaan yang dicapai pengguna dalam menjalankan tugas.
3. Kepuasan menunjukkan pengguna merasa bebas dari ketidaknyamanan dan menunjukkan perilaku positif terhadap penggunaan produk.

Dalam hal *errors*, Jakob Nielsen menyatakan terdapat sepuluh kesalahan yang paling banyak dilakukan dalam desain *interface web* yang bertentangan dengan usability. Kesepuluh kesalahan tersebut adalah[8]:

1. *Bad Search* (sistem pencarian yang buruk).
2. *PDF Files for Online Reading* (menampilkan materi bacaan dalam format PDF).
3. *Not Changing the Color of Visited Links* (tidak mengganti warna dari tautan yang sudah dibuka).
4. *Non-Scannable Text* (tulisan yang susah dibaca sekilas).
5. *Fixed Font Size* (ukuran huruf yang tidak bisa diubah).
6. *Page Titles with Low Search Engine Visibility* (judul halaman yang kurang terbaca mesin pencari).
7. *Anything That Looks Like an Advertisement* (seluruh materi terlihat seperti iklan).

8. *Violating Design Conventions* (tidak menggunakan aturan desain).
9. *Opening New Browser Windows* (membuka jendela *browser* baru).
10. *Not Answering Users' Questions* (tidak menjawab pertanyaan pengguna).

## 2.5 System Usability Scale (SUS)

*System Usability Scale* (SUS) merupakan kuesioner yang dapat digunakan untuk mengukur *usability* sistem komputer menurut sudut pandang subjektif pengguna. SUS dikembangkan oleh John Brooke sejak tahun 1986. Pengujian *usability* dijalankan untuk mengetahui seberapa efektif, efisien, dan memuaskan sebuah *website* menurut penggunaannya. Ada beberapa kuesioner penilaian *usability* yang siap digunakan, antara lain[1]:

1. SUS. SUS dikembangkan sebagai sebuah pengukuran *usability* yang “*quick and dirty*”. Survei terdiri dari 10 (sepuluh) pertanyaan, masing-masing memiliki 5 (lima) poin *Likert* sebagai tanggapan. *Output* SUS berupa skor yang tampak mudah dipahami, dengan *range* dari 0 hingga 100, dengan semakin besar skor berarti semakin baik *usability*-nya.
2. *Questionnaire for User Interface Satisfaction* (QUIS). QUIS didesain untuk menilai kepuasan subjektif pengguna terhadap aspek khusus interaksi manusia.
3. *Software Usability Measurement Inventory*(SUMI). SUMI adalah kuesioner berlisensi yang terdiri dari 50 pertanyaan. SUMI dapat digunakan untuk mengukur persepsi efisiensi, efektif, kegunaan, *system*, dan *learnability* pengguna atas sistem. SUMI tersedia dalam 12 bahasa. SUMI sangat reliabel (0.92). Harga lisensi SUMI sekitar USD \$700 sebulan.
4. *Post-Study Usability Questionnaires*(PSSUQ). PSSUQ adalah kuesioner dengan 16 item pertanyaan. PSSUQ mengukur kepuasan yang dirasakan pengguna terhadap produk atau sistem. PSSUQ memberikan skor kepuasan secara keseluruhan dengan rata-rata sub-skala, yaitu: kualitas sistem, kualitas informasi, dan kualitas antarmuka. PSSUQ tersedia secara gratis.

SUS berupa kuesioner yang terdiri dari 10 (sepuluh) item pertanyaan. Kuesioner SUS menggunakan 5 (lima) poin skala *Likert*. Responden diminta untuk memberikan penilaian “Sangat Tidak Setuju”, “Tidak Setuju”, “Netral”, “Setuju”, dan “Sangat Setuju” atas 10 (sepuluh) item pernyataan SUS sesuai dengan penilaian



subjektifnya. Jika responden merasa tidak menemukan skala respon yang tepat, maka responden harus mengisi titik tengah skala pengujian[1].

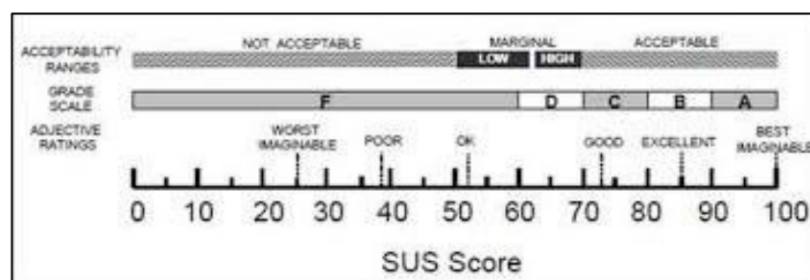
Tabel 2.2 Item Pernyataan SUS

Kode	Item Pernyataan
R1	Saya akan sering menggunakan/mengunjungi situs ini
R2	Saya menilai situs ini terlalu kompleks (memuat banyak hal yang tidak perlu)
R3	Saya menilai situs ini mudah dijelajahi
R4	Saya membutuhkan bantuan teknis untuk menggunakan/menjelajahi situs ini
R5	Saya menilai fungsi/fitur yang disediakan pada situs ini dirancang dan disiapkan dengan baik
R6	Saya menilai terlalu banyak inkonsistensi pada situs ini
R7	Saya merasa kebanyakan orang akan mudah menggunakan/menjelajahi situs ini dengan cepat
R8	Saya menilai situs ini sangat rumit untuk dijelajahi
R9	Saya merasa sangat percaya diri menjelajahi situs ini
R10	Saya perlu belajar banyak hal sebelum saya dapat menjelajahi situs ini dengan baik

Setiap item pernyataan memiliki skor kontribusi. Setiap skor kontribusi item akan berkisar antara 0 hingga 4. Untuk item 1, 3, 5, 7, dan 9, skor kontribusinya adalah posisi skala dikurangi 1. Untuk item 2, 4, 6, 8, dan 10, skor kontribusinya adalah 5 dikurangi posisi skala. Kalikan jumlah skor kontribusi dengan 2.5 untuk mendapatkan nilai keseluruhan *system usability*. Skor SUS berkisar dari 0 hingga 100. Berikut ini adalah rumus perhitungan skor SUS [1].

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

Skor SUS keseluruhan diperoleh dari rata-rata skor SUS individual. Berikut ini merupakan *rating score* penilaian dari hasil perhitungan kuesioner[10].



Gambar 2.1 Rating Score Penilaian

Untuk menentukan *grade* hasil penilaian ada 2 (dua) cara yang dapat digunakan. Penentuan pertama dilihat dari sisi tingkat penerimaan pengguna. *Grade* skala dan *adjektifrating* yang terdiri dari tingkat penerimaan pengguna terdapat tiga kategori, yaitu *not acceptable*, *marginal*, dan *acceptable*. Sedangkan dari sisi tingkat *grade* skala terdapat lima skala, yaitu A, B, C, D, dan F, serta dari *adjektifrating* terdiri dari *worst imaginable*, *poor*, *ok*, *good*, *excellent*, dan *best imaginable*. Penentuan yang kedua dilihat dari sisi *percentile range* (SUS skor) yang memiliki *grade* penilaian yang terdiri dari A, B, C, D, dan F. Dari dua cara penentuan hasil penilaian tersebut, maka dapat dilihat hasil penilaian sebagai berikut [10]:

1. *Acceptability, Grade Scale, Adjective Rating*: Penentuan *acceptability*, *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 [10]:

1. *Grade A*: dengan skor lebih besar atau sama dengan 80,3
2. *Grade B*: dengan skor lebih besar sama dengan 74 dan lebih kecil dari 80,3
3. *Grade C*: dengan skor lebih besar sama dengan 68 dan lebih kecil dari 74
4. *Grade D*: dengan skor lebih besar sama dengan 51 dan lebih kecil dari 68
5. *Grade F*: dengan skor lebih kecil dari 51

## 2.6 SystemFlowchart





Secara konseptual, *system flowchart* adalah kombinasi dari DFD logis dan DFD fisik, karena menyediakan rincian dari proses-proses yang dilakukan (perspektif logis) serta sumber daya fisik yang digunakan olehnya (perspektif fisik). Namun, *system flowchart* tidak hanya menunjukkan pada apa saja prosesnya, seperti halnya DFD logis, atau siapa saja yang terlibat, seperti halnya DFD fisik. Sebaliknya,

ini memberikan lebih banyak detail tentang apa yang sebenarnya terjadi di dalam sistem itu sendiri. *System flowchart* tidak hanya menunjukkan bahwa rekuisisi yang masuk saja yang bisa dicatat, seperti yang dilakukan DFD logika pada level 0, *system flowchart* juga dilengkapi dengan *process map*. Ini akan memberikan lebih banyak detail tentang apa yang sebenarnya terjadi dalam setiap proses yang muncul dalam peta proses. Namun, berbeda dengan *process map*, dimana entitas peta proses tersebut tercantum di bawah margin kiri, sementara *system flowchart* mencantumkan entitasnya di bagian atas dokumen [11].









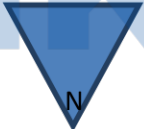
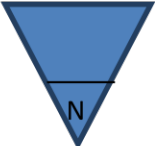

Hal lain yang perlu diperhatikan adalah bahwa dibandingkan dengan peta proses dan DFD, *system flowchart* memiliki jangkauan simbol yang lebih luas yang dapat digunakan, sehingga memungkinkan lebih banyak detail dan wawasan untuk dokumentasi pengguna. Dengan ini, sangat penting bahwa sebelum melangkah terlalu jauh ke bagian ini akan menghabiskan banyak waktu untuk membiasakan diri dengan simbol-simbol yang berbeda yang akan digunakan dalam *system flowchart* [11].

Tabel berikut ini berisi beberapa simbol yang biasanya digunakan dalam menggambarkan *systemflowchart*, dengan deskripsi singkat dari simbol-simbol yang akan digunakan [11].



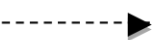

Tabel 2.3 Simbol-Simbol yang Digunakan pada *System Flowchart*

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 keluar 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

Tabel 2.3 Simbol-Simbol yang Digunakan pada *System Flowchart*(Sambungan)

Simbol	Deskripsi
	<i>Tape drive</i> atau penyimpanan <i>tape</i> magnetik
	<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

Tabel 2.3 Simbol-Simbol yang Digunakan pada *System Flowchart*(Sambungan)

Simbol	Deskripsi
	Jurnal umum atau buku besar umum
	Aliran dokumen atau proses
	Aliran data atau informasi
	Pengiriman data antara dua tempat berbeda melalui tautan telekomunikasi

Ada dua fitur utama yang harus diketahui untuk dapat memahami dan membaca *systemflowchart*: divisi organisasi dan simbol *flowchart*. Ketika membacasytem *flowchart*, mulai dari sudut kiri atas dokumen dan kemudian maju ke sudut kanan bawah. Awal *system flowchart* biasanya ditunjukkan oleh penggunaan simbol *start/stop* atau keberadaan simbol entitas eksternal yang berisi nama entitas eksternal [11].

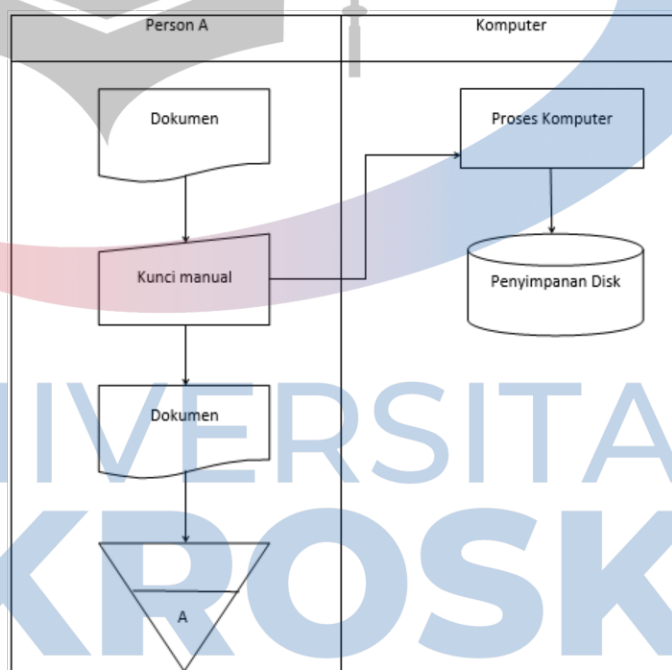
*System flowchart* memiliki beberapa kolom. Setiap kolom harus sesuai dengan entitas internal yang merupakan bagian dari proses yang didokumentasikan. Salah satu hal yang harus diperhatikan adalah ketika membaca *system flowchart* dan membandingkannya dengan DFD fisik adalah bahwa jumlah kolom dalam *system flowchart* dan pelabelan kolom di dalam *system flowchart* sama dengan jumlah entitas internal dan pelabelan entitas internal dalam DFD fisik[11].

Masing-masing entitas internal dalam *system flowchart* dipisahkan oleh garis yang solid dan nama entitas dapat ditemukan di bagian atas halaman. Segala sesuatu yang muncul dalam kolom untuk entitas internal adalah representasi grafis dari aktivitas entitas dan bagaimana entitas menjalankan aktivitasnya di dalam sistem[11].

Simbol *system flowchart* akan muncul dalam kolom dan menunjukkan apa yang terjadi dalam sebuah divisi. Setiap simbol menyampaikan makna yang berbeda-beda, seperti yang ditunjukkan pada tabel di atas. Simbol tidak hanya memberi tahu data apa yang digunakan atau tugas apa yang sedang dilakukan, namun juga menyampaikan bagaimana cara suatu tugas dapat dilakukan. Misalnya, perbedaan

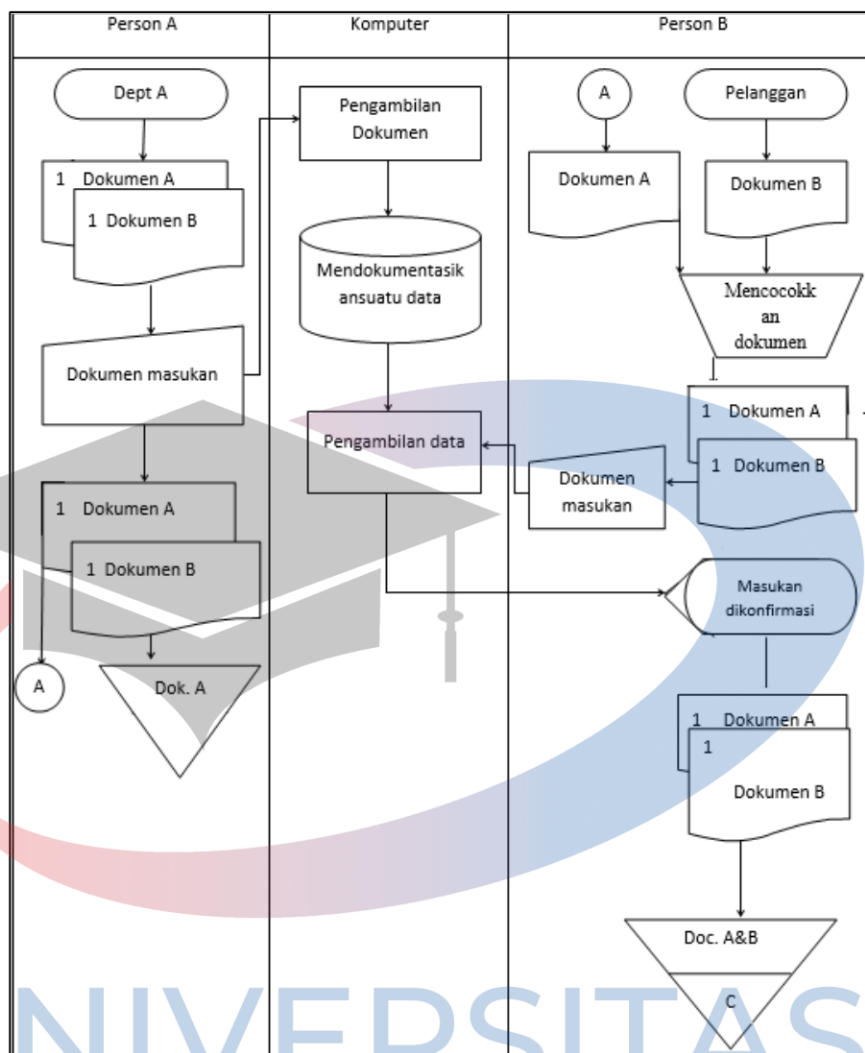
antara proses komputerisasi dan proses manual, aktivitas yang dilakukan oleh seorang manusia dengan aktivitas yang dilakukan oleh komputer sangatlah berbeda, dan telah terbukti juga melalui simbol yang berbeda untuk masing-masing aktifitas yang dilakukan. Ini akan menambah daya tarik semantik *flowchart*, karena menyampaikan tambahan yang tidak ada dalam DFD. Dengan demikian, akan menjadi tahu tidak hanya pada apa yang dilakukan saja, tetapi juga bagaimana cara itu dilakukan[11].

Gambar berikut ini menggambarkan orang yang secara manual memasukkan dokumen ke komputer. Komputer melakukan proses pada data dan menyimpan hasilnya pada *disk*. Dokumen tersebut kemudian dimasukkan ke dalam urutan abjad[11].



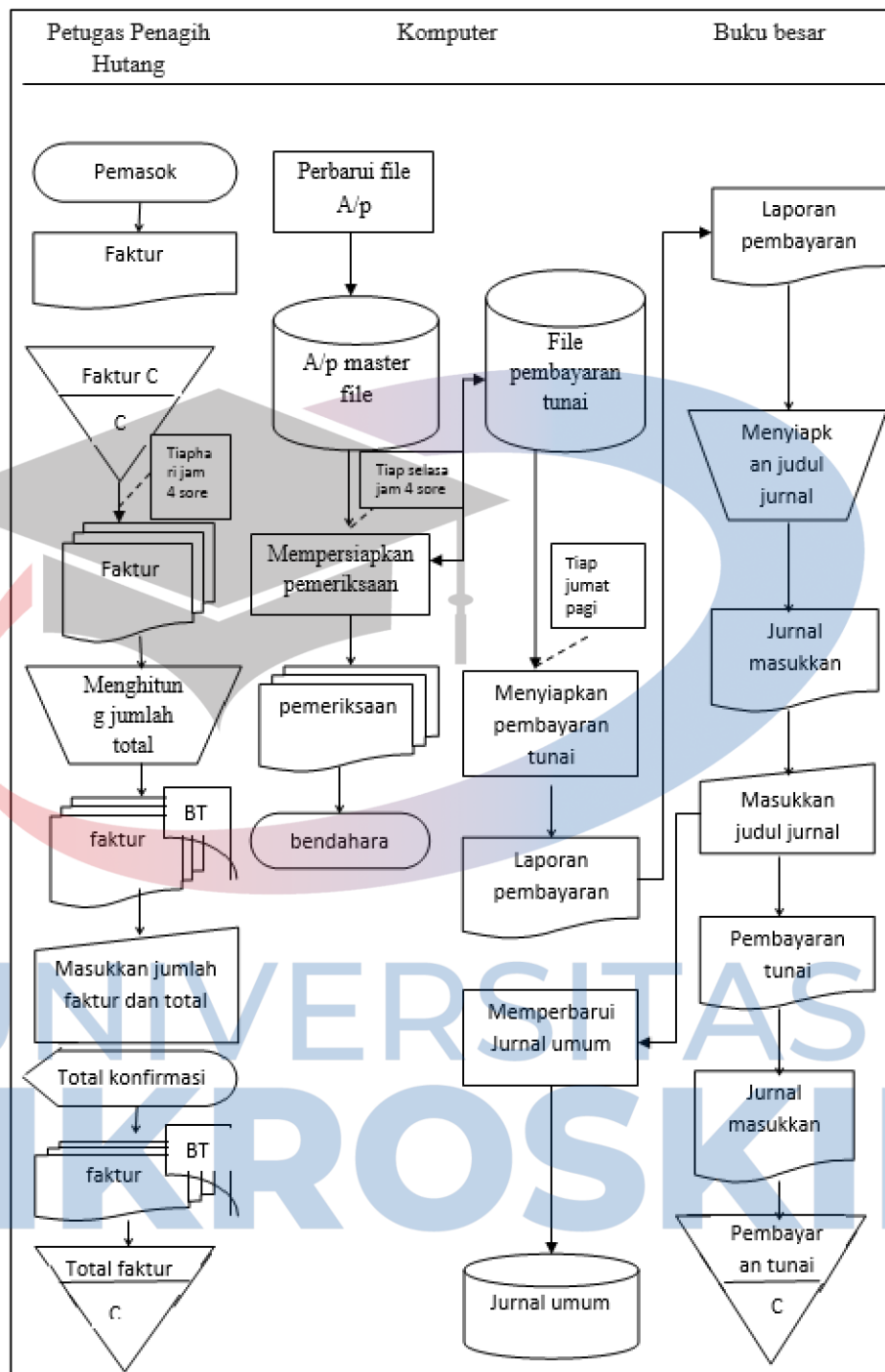
Gambar 2.2 Contoh Penggambaran *SystemFlowchart* (1)

Gambar berikut ini menggambarkan departemen mengirimkan dua salinan dokumen A ke A. Rincian dokumen A secara manual dimasukkan ke komputer, yang menyimpan hasilnya pada disk. Satu salinan dokumen A kemudian disimpan dalam urutan abjad, sementara salinan kedua diteruskan ke B. B menerima dokumen B dari pelanggan, dan secara manual dimasukkan ke komputer. Komputer memroses *input*, menyimpan rincian, dan menampilkan konfirmasi pada layar yang digunakan oleh B. B kemudian memiliki kedua *file* dokumen tersebut secara kronologis[11].



Gambar 2.3 Contoh Penggambaran *SystemFlowchart* (2)

Juga harus diperhatikan penggunaan konektor halaman. Konektor digunakan untuk menunjukkan pergerakan dokumen A dari A ke B. Menggunakan konektor menghindari masalah garis yang terjadi di semua tempat, menghindari kebingungan, dan kekacauan dalam diagram. Sebagai aturan umum, jika aliran mengalir ke kolom yang tidak bersebelahan, maka konektor halaman harus digunakan. Konektor halaman *offline* digunakan untuk tujuan yang sama. Perhatikan gambar berikut ini [11].



Gambar 2.4 Contoh Penggambaran *SystemFlowchart* (3)