BAB II
TINJAUAN PUSTAKA

2.1 Konsep Sistem Informasi

2.1.1 Sistem

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama sama untuk melakukan kegiatan atau untuk menyelesaikan suatu sasaran tertentu [1]. Sistem adalah sekumpulan elemen yang saling terkait / terpadu yang dapat mencapai suatu tujuan [2].

Jadi, dari beberapa pengertian sistem diatas dapat disimpulkan bahwa sistem adalah kumpulan dari beberapa komponen yang saling terhubung berkaitan satu sama lain dengan batasan – batasan tertentu yang saling bekerja sama untuk mencapai tujuan tertentu.

Sistem terdiri atas 3 komponen utama, yaitu:

1. Masukan, mencakup segala sesuatu baik itu data mentah, masukan perawatan dan masukan sinyal yang masuk kedalam sistem untuk kemudian dilakukan pemrosesan.
2. Pemrosesan, mencakup bagian yang memproses masukan untuk menjadi keluaran yang diinginkan. Contohnya pemrosesan transaksi, proses produksi, proses pernapasan manusia dan sebagainya.
3. Keluaran, mencakup hasil dari masukan yang telah diproses oleh sistem menuju tujuan akhirnya. Contohnya produk jadi, informasi, laporan dan lain sebagainya.

Konsep sistem yang memiliki 3 komponen dasar akan berjalan baik jika melibatkan dua elemen tambahan, yaitu pengendalian dan umpan balik. Maka sistem yang memiliki pengendalian dan umpan balik tersebut adalah sistem yang dapat mengawasi dan mengatur secara terotomatisasi.

a. Umpan balik

   Umpan balik biasanya membandingkan kinerja sistem saat ini dengan tujuan yang telah ditentukan dan memberikan kembali informasi yang menggambarkan kesenjangan antara kinerja aktual dan yang diinginkan [3].

b. Pengendalian

   Pengendalian mencakup pengawasan, penanganan, penanggulangan, dan evaluasi. Umpan balik untuk melihat dan menentukan apakah sistem tersebut bergerak sesuai dengan pencapaian tujuannya. Jika keluaran tidak

sesuai dengan yang diinginkan, akan dilakukan penyesuaian terhadap masukan sistem atau elemen pemrosesan, hingga dihasilkan keluaran yang sesuai dengan kebutuhan.

2.1.2 Informasi

Informasi merupakan data yang telah diproses ke dalam bentuk yang memiliki arti bagi penerima dan memiliki nilai yang bermanfaat [1]. Adapun beberapa jenis - jenis informasi sebagai berikut [4] :

- Informasi Absolut

  Merupakan induk dari informasi yang disampaikan dengan jaminan dan tidak diperlukan penjelasan selanjutnya.
- Informasi Substitusional

  Informasi ini memiliki konsep yang dipakai pada beberapa informasi. Istilah informasi substitusional bisa disebut juga komunikasi.
- Informasi filosofis

  Jenis informasi ini merupakan konsep informasi yang terhubung antara pengetahuan dan kebijakan.
- Informasi subjektif

  Jenis informasi ini memiliki keterkaitan antara perasaan dan informasi manusia. Informasi ini sangat mendukung penyajinya atau orang yang menyampaikan informasi.
- Obyek informasi

  Jenis informasi tertuju pada informasi informasi tertentu yang logis.
- Informasi budaya

  Jenis informasi yang ditekankan pada dimensi budaya.

Kualitas Informasi *(Quality of Information)* tergantung dari 4 hal, yaitu [1] :

1. Akurat Informasi harus benar-benar bebas dari kesalahan dan tidak menyesatkan.
2. Tepat Waktu Informasi yang datang kepada penerima tidak boleh terlambat.
3. Relevan Informasi harus mempunyai manfaat untuk pemakainya.
4. Lengkap Informasi yang akan digunakan harus selengkap mungkin, jangan setengah-setengah.

2.1.3 Sistem Informasi

Sistem informasi secara teknis dapat didefinisikan sebagai sekumpulan komponen yang saling terkait yang mengumpulkan (atau mengambil), memproses, menyimpan, dan mendistribusikan informasi untuk mendukung pengambilan keputusan dan pengendalian di dalam perusahaan. Selain mendukung pengambilan keputusan, pengoordinasian dan pengendalian, sistem informasi juga dapat membantu para manajer dan karyawan untuk menganalisis masalah, memvisualisasikan masalah-masalah yang rumit dan menciptakan produk baru [5]. Sistem informasi dapat dianalisis dari setidaknya tiga perspektif yang berbeda, tetapi saling melengkapi meliputi [6]:

1. Kontribusi yang mereka berikan.
2. Struktur dan perilaku mereka.
3. Fungsi yang mereka lakukan.

Sistem informasi yang dikembangkan untuk tujuan yang berbeda-beda, tergantung pada kebutuhan bisnis. Sistem informasi dapat dibagi menjadi beberapa bagian, meliputi [3] :

1. *Transaction Processing Systems* (TPS)

   TPS adalah sistem informasi terkomputerisasi yang dikembangkan untuk memproses data dalam jumlah besar untuk transaksi bisnis rutin seperti daftar gaji dan inventarisasi.

2. *Office Automation Systems* (OAS) dan *Knowledge Work Systems* (KWS)

   OAS dan KWS bekerja pada level *knowledge*. OAS mendukung pekerja data, yang biasanya tidak menciptakan pengetahuan baru melainkan hanya menganalisis informasi sedemikian rupa untuk mentransformasikan data atau memanipulasikannya secara keseluruhan dengan organisasi dan kadang-kadang diluar organisasi. Aspek-aspek OAS seperti pengolah kata, *spreadsheet*, penjadwalan elektronik, dan komunikasi melalui *voice mail, email* dan konferensi video.

3. *Management Information Systems*

   SIM tidak menginstal TPS, tetapi mendukung tugas-tugas organisasional yang lebih luas dari TPS, termasuk analisis keputusan dan pembuat keputusan. SIM

menghasilkan informasi yang digunakan untuk membuat keputusan, dan juga dapat membantu tanpa beberapa fungsi informasi bisnis yang sudah terkomputerisasi (data dasar).

4. *Decision Support Systems* (DSS)

   DSS hampir sama dengan SIM karena menggunakan basis data sebagai sumber data. DSS dimulai dari SIM karena pada fungsi mendukung pembuat keputusan di seluruh tahap-tahapnya, meskipun keputusan sebenarnya tetap berwenang berwenang membuat keputusan.

5. *Artificial Intelligence* (AI) *and Expert System* (ES)

   Bantuan untuk mengembangkan mesin-mesin yang bekerja secara cerdas. Dua cara untuk melakukan riset Al adalah memahami bahasa alaminya dan menganalisis kemampuannya untuk berpikir melalui masalah sampai kesimpulan logikanya. Sistem ahli menggunakan pendekatan-pendekatan pemikiran Al untuk menyelesaikan masalah serta memberikannya melalui pengguna bisnis.

6. *Group Decision Support Systems* (GDSS) *and Computer-Support Collaborative Work Systems* (CSCW)

   Bila kelompok, perlu bekerja bersama-sama untuk membuat keputusan semi-terstruktur dan tak terstruktur, maka kelompok Sistem pendukung keputusan membuat suatu solusi. GDSS dimaksudkan untuk membawa kelompok bersama-sama menyelesaikan masalah dengan memberikan bentuk pendapat, kuesioner, konsultasi dan skenario.

7. *Executive Support Systems* (ESS)

   ESS tergantung pada informasi yang dihasilkan TPS dan SIM dan ESS membantu eksekutif mengatur interaksinya dengan lingkungan pendukung komunikasi di tempat-tempat yang dapat diakses seperti eksternal dengan menyediakan grafik-grafik dan kantor.

2.2 Analisis Sistem

Analisis sistem merupakan suatu teknik penelitian terhadap sebuah sistem dengan menguraikan komponen- komponen pada sistem tersebut dengan tujuan untuk mempelajari komponen itu sendiri serta keterkaitannya dengan komponen lain yang

membentuk sistem sehingga didapat sebuah keputusan atau kesimpulan mengenai sistem tersebut baik itu kelemahan ataupun kelebihan sistem [7].

Dalam analisis sistem informasi dipelajari masalah dan kebutuhan dari organisasi untuk menentukan bagaimana orang, data, proses, komunikasi, dan teknologi informasi dapat meningkatkan manajemen bisnis. Tujuan analisis sistem adalah menemukan solusi bagi permasalahan di dalam sistem dan peluang-peluang peningkatan kinerja sistem.

Langkah-langkah dalam analisis sistem meliputi:

1. Mempelajari dan mendokumentasikan sistem yang ada
2. Menyelesaikan analisis pendokumentasian
3. Menganalisis alternatif sistem yang baru

2.3 Analisis Kebutuhan Sistem

Analisis kebutuhan adalah sebuah proses untuk mendapatkan informasi, model, spesifikasi tentang perangkat lunak yang diinginkan klien/pengguna. Kedua belah pihak, yaitu klien dan pembuat perangkat lunak terlibat aktif dalam tahap ini. Informasi yang diperoleh dari klien/pengguna inilah yang akan menjadi acuan untuk melakukan desain perangkat lunak [8].

Tahap penentuan kebutuhan sistem merupakan tahap yang penting dalam melakukan pengembangan sistem. Inti dari tahapan ini adalah mengidentifikasi hal – hal yang harus dikerjakan oleh sistem, yaitu proses-proses apa saja yang nantinya dilakukan oleh sistem dan kemampuan atau karakteristik yang harus dimiliki oleh sistem.

Kebutuhan sistem meliputi:

a. Kebutuhan fungsional
b. Kebutuhan non-fungsional

2.3.1 Kebutuhan Fungsional

Kebutuhan fungsional adalah mencakup proses-proses apa saja yang nantinya dilakukan oleh sistem baru dan juga berisi informasi-informasi apa saja yang harus ada dan dihasilkan oleh sistem [9]. Kebutuhan fungsional, dinyatakan secara eksplisit apa yang dilakukan oleh sistem, seperti:

- Pernyataan layanan sistem yang tersedia
- Bagaimana sistem bereaksi terhadap masukan tertentu
- Bagaimana kelakuannya dalam situasi tertentu

Spesifikasi kebutuhan fungsional sistem harus lengkap (seluruh layanan yang dibutuhkan oleh pengguna didefinisikan) dan konsisten. Meskipun pada prakteknya sulit untuk mencapai kebutuhan yang lengkap dan konsisten.

Adapun alat bantu yang tersedia untuk membantu mendefinisikan analisis kebutuhan fungsional meliputi : *Data Flow Diagram* (DFD) *physical, Data Flow Diagram* (DFD) *logical, Use Case Diagram* dan *Context Diagram*. Pada tahap analisis kebutuhan fungsional penulis akan menggunakan tools *Data Flow Diagram* (DFD) *physical* untuk sistem berjalan dan *Data Flow Diagram* (DFD) *logical* untuk sistem usulan.

2.3.1.1 *Data Flow Diagram* (DFD)

Salah satu perangkat dalam memodelkan sistem yang paling umum adalah *Data Flow Diagram* (DFD), terutama untuk menggambarkan aliran data dalam proses bisnis. DFD memberikan gambaran sistem secara menyeluruh, lengkap dengan lingkup sistem, hubungan dengan sistem lainnya dan komponen-komponen sistem secara detail yang terkait dengan aliran data di dalam sistem. Analisis sistem dimulai dengan menganalisis *current system* (seperti *current logical* DFD), kemudian dilanjutkan dengan menganalisis dan menambahkan fitur-fitur untuk memperbaiki atau meningkatkan kinerja sistem tersebut.

Data Flow Diagram (DFD) terdiri atas 2 jenis, yaitu:

1. **DFD *Logical model*** menunjukkan struktur dan cara kerja sistem. Model tersebut bersifat *implementation-independent*, yaitu memberi gambaran tentang sistem terlepas dari implementasi teknis.
2. **DFD *Physical model*** tidak hanya menunjukkan apa sebenarnya sistem tersebut atau apa yang dilakukannya, tetapi bagaimana sistem tersebut diimplementasikan secara fisik dan teknis. Model tersebut *implementation-dependent* karena merefleksikan pilihan teknologi dan batasan pilihan teknologi. DFD Fisik menampilkan bagaimana sistem diimplementasikan, termasuk *hardware*, *software*, *file*, dan orang yang terlibat dalam sistem.

DFD Fisik terkait dengan bagaimana sistem dikonstruksi dan biasanya meliputi:

a. Proses untuk penambahan, penghapusan, pengubahan, dan pembaharuan *record*.
b. Proses memasukkan data dan memverifikasi.
c. Proses validasi untuk memastikan keakuratan data yang dimasukkan.
d. Proses pengurutan untuk mengatur kembali urutan *record.*
e. Proses untuk menghasilkan setiap keluaran sistem yang unik melanjutkan penyimpanan data.
f. Nama-nama *file* aktual yang digunakan untuk menyimpan data.
g. Kontrol untuk menandai selesainya tugas [10].

Berikut adalah beberapa aturan dasar yang harus diikuti, meliputi [3]:

1. Diagram aliran data harus memiliki setidaknya satu proses, dan tidak boleh memiliki objek atau objek yang berdiri sendiri yang terhubung dengan dirinya sendiri
2. Suatu proses harus menerima setidaknya satu aliran data yang masuk ke proses dan membuat setidaknya satu aliran data yang keluar dari proses
3. Sebuah penyimpanan data harus terhubung ke setidaknya satu proses
4. Entitas eksternal tidak boleh terhubung satu sama lain. Meskipun mereka berkomunikasi secara independen, komunikasi itu bukan bagian dari sistem yang dirancang menggunakan DFD.

2.3.2 Kebutuhan Non – Fungsional

Kebutuhan non-fungsional adalah kebutuhan yang menggambarkan bagaimana aplikasi/sistem bekerja beserta batasan-batasan yang menyertainya dalam menjalankan fungsionalitas sistem. Kebutuhan non-fungsional mendefinisikan atribut atau parameter kualitas dari sebuah aplikasi/sistem [11].

Kebutuhan Non-Fungsional, berlaku untuk keseluruhan sistem. Batasan layanan atau fungsi yang ditawarkan sistem mencakup batas pemilihan waktu, batas proses pengembangan, dan batas yang dikenakan oleh standar. Misalnya kehandalan, waktu respon, keamanan, kinerja atau kapabilitas I/O, perangkat, representasi data yang digunakan dalam antarmuka dengan sistem lain. Analisis kebutuhan non-fungsional dilakukan untuk melihat spesifikasi kebutuhan untuk sistem. Pada tahap analisis kebutuhan non – fungsional penulis akan menggunakan *tools* PIECES.

2.3.2.1 PIECES

Analisis PIECES adalah metode analisis sebagai dasar untuk memperoleh pokok-pokok permasalahan yang lebih spesifik. Dalam menganalisis sebuah sistem, biasanya akan dilakukan terhadap beberapa aspek antara lain analisis terhadap kinerja, informasi, ekonomi, pengendalian, efisiensi dan pelayanan. Analisis ini disebut analisis PIECES (*Performance*, *Information, Economic, Control, Efficiency* and *Service*) [12].

Analisis PIECES meliputi [13] :

1. Performance

Persyaratan kinerja mewakili kinerja yang harus ditunjukkan sistem untuk memenuhi kebutuhan pengguna, seperti:

• Berapa tingkat *throughput* yang dapat diterima?

• Berapa waktu tanggapan yang dapat diterima?

2. Information

Persyaratan informasi mewakili informasi yang relevan dengan pengguna dalam hal konten, ketepatan waktu, keakuratan, dan format.

• Apa *input* dan *output* yang diperlukan?

• Kapan itu harus terjadi?

• Data apa yang perlu disimpan ?

• Seberapa seberapa sering informasi tersebut harus dimutakhirkan?

• Apa antarmuka ke sistem eksternal?

3. Economy

Persyaratan ekonomi mewakili kebutuhan sistem untuk mengurangi biaya atau meningkatkan keuntungan.

• Apa area sistem di mana biaya harus dikurangi?

• Berapa biaya yang harus dikurangi atau keuntungan ditingkatkan?

• Berapa batasan anggaran?

• Bagaimana jadwal pengembangannya?

4. Control (and security)

Persyaratan pengendalian mewakili lingkungan di mana sistem harus beroperasi, serta jenis dan tingkat keamanan yang harus disediakan.

• Haruskah akses ke sistem atau informasi dikendalikan?

• Apa persyaratan privasi?

• Apakah kekritisan data memerlukan penanganan khusus (cadangan, penyimpanan di luar situs, dll) dari data?

5. Efficiency

Persyaratan efisiensi mewakili kemampuan sistem untuk menghasilkan keluaran dengan pemborosan minimal.

• Apakah ada langkah duplikat dalam proses yang harus dihilangkan?

• Apakah ada cara untuk mengurangi pemborosan dalam cara sistem menggunakan sumber dayanya?

6. Service

Persyaratan layanan mewakili kebutuhan agar sistem dapat diandalkan, fleksibel, dan dapat diperluas.

• Siapa yang akan menggunakan sistem, dan di mana mereka berada?

• Apakah akan ada tipe pengguna yang berbeda?

• Bagaimana persyaratan terkait dengan faktor ergonomis ?

• Perangkat pelatihan dan materi pelatihan apa yang akan dimasukkan ke dalam sistem?

• Perangkat pelatihan dan materi pelatihan apa yang akan dikembangkan dan dijalankan secara terpisah dari sistem, seperti program pelatihan berbasis komputer (CBT) atau program pelatihan basis data ?

• Apa persyaratan keandalan / ketersediaan?

• Bagaimana seharusnya sistem digabungkan menjadi suatu paket perangkat keras dan perangkat lunak *(package)* dan didistribusikan?

• Dokumentasi apa yang dibutuhkan?

Berikut adalah beberapa kriteria penilaian dimana kualitas suatu sistem bisa dikatakan buruk:

a. Sistem menghasilkan produk yang tidak akurat.

b. Sistem menghasilkan produk yang tidak konsisten.

c. Sistem menghasilkan produk yang tidak dipercaya.

d. Sistem tidak mudah dipelajari.

e. Sistem tidak mudah digunakan.

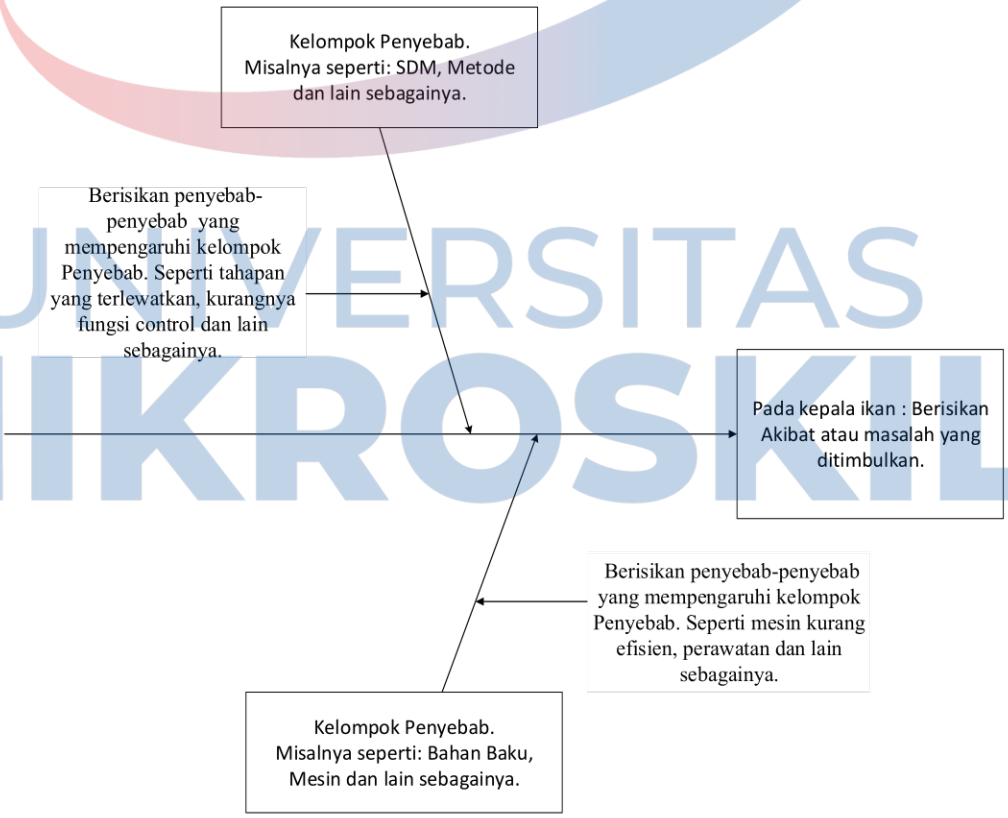
f. Sistem canggung untuk digunakan.

g. Sistem tidak fleksibel.

2.3.3 Diagram Fishbone

Diagram *fishbone* (tulang ikan) merupakan suatu alat visual untuk mengidentifikasi, mengeksplorasi, dan menggunakan media grafik dalam menggambarkan secara detail semua kemungkinan penyebab yang berhubungan dengan suatu permasalahan. Dalam diagram *fishbone,* permasalahan mendasar diletakkan pada bagian kanan dari diagram atau pada bagian kepala dari kerangka tulang ikan. Penyebab permasalahan digambarkan pada tulang dan durinya. Kategori penyebab permasalahan yang sering digunakan sebagai awal meliputi *materials* (bahan baku), *machines and equipment* (mesin dan peralatan), *manpower* (sumber daya manusia), *methods* (metode), *Mother Nature/environment* (lingkungan), dan *measurement* (pengukuran) [14].

Berikut konsep penggambaran Diagram *Fishbone* :



Gambar 2. 1 Diagram Fishbone

2.4 Perancangan Sistem

Pada saat membuat sebuah sistem yang akan digunakan pada suatu perusahaan, setiap pengembang aplikasi diharuskan membuat sebuah rancangan dari sistem yang ingin dibuat. Rancangan ini bertujuan untuk memberi gambaran dari sistem yang akan berjalan nantinya kepada setiap orang yang berkepentingan dengan sistem tersebut.

Perancangan sistem adalah proses untuk merancang suatu sistem baru atau memperbaiki suatu sistem yang telah ada sehingga sistem tersebut menjadi lebih baik dan biasanya proses ini terdiri dari proses merancang *output*, *input*, dan basis data.

Adapun tujuan perancangan sistem antara lain:

1. Untuk memenuhi spesifikasi fungsional maupun non-fungsional.
2. Untuk memberikan *blueprint* yang jelas dan lengkap kepada pemrogram komputer dalam mengimplementasikan sistem kedepan nya.
3. Untuk tercapainya pemenuhan kebutuhan berkaitan dengan pemecahan masalah yang menjadi sasaran perancangan sistem.
4. Untuk memaksimalkan solusi yang diusulkan melalui perancangan sistem.

Komponen sistem yang perlu didesain, meliputi [3]:

1. Perancangan *output* yang efektif
2. Perancangan *input* yang efektif
3. Perancangan basis data
4. Perancangan pengalaman pengguna dan interaksi manusia dengan komputer

Berikut tahapan-tahapan perancangan sistem, meliputi [15]:

1. Perancangan *Output*

   Perancangan *output* tidak dapat diabaikan, karena berbagai laporan yang disediakan harus memudahkan bagi setiap unsur manusia yang membutuhkan.

2. Perancangan *Input*

   Tujuan dari perancangan *input* dapat mengefektifkan biaya pemasukan data, mencapai keakuratan yang tinggi, dan dapat menjamin pemasukan data yang

akan diterima dan dimengerti oleh pemakai. Perancangan *input* ini termasuk bagian dari tahapan perancangan *user interface*.

3. Perancangan Proses Sistem

   Tujuan dari perancangan proses sistem adalah menjaga agar proses data lancar sehingga dapat menghasilkan informasi yang benar dan mengawasi proses dari sistem.

4. Perancangan Basis Data

   Sistem basis data adalah mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lainnya.

2.4.1 User Interface

Terdiri dari semua perangkat keras, perangkat lunak, layar, menu, fungsi, *input*, dan fitur yang mempengaruhi komunikasi dua arah antara pengguna dan komputer. Dapat dikatakan *user interface* adalah jembatan yang menghubungkan antara pengguna dengan sebuah *device* atau perangkat yang akan digunakan.

Adapun beberapa jenis antarmuka pengguna meliputi:

a. CLI (*Command Line Interface*) adalah *Interface* atau antarmuka berbasis teks yang digunakan untuk berinteraksi dengan perangkat lunak dan sistem operasi dengan mengetikkan perintah ke dalam antarmuka dan menerima respons dengan cara yang sama.

b. GUI (*Graphic User Interface*) adalah jenis antarmuka pengguna yang menggunakan metode interaksi pada peranti elektronik secara grafis (bukan perintah teks) antara pengguna dan komputer.

Adapun *Eight Golden Rules of Interface Design* yaitu [16]:

1. Upayakan konsistensi

Aturan ini adalah yang paling sering dilanggar, tetapi mengikutinya bisa jadi rumit karena ada banyak bentuk konsistensi. Urutan tindakan yang konsisten harus diperlukan dalam situasi serupa; terminologi identik harus digunakan dalam petunjuk, menu, dan layar bantuan; dan warna, tata letak, kapitalisasi, *font*, dan sebagainya yang konsisten harus digunakan sepanjang waktu. Pengecualian, seperti konfirmasi yang diperlukan untuk perintah delete atau

tidak ada kata sandi yang bergema, harus dapat dipahami dan jumlahnya dibatasi.

2. Memenuhi kegunaan universal

Mengenali kebutuhan beragam pengguna dan desain untuk plastisitas, memfasilitasi transformasi konten. Perbedaan ahli pemula, rentang usia, kecacatan, dan keragaman teknologi masing-masing memperkaya spektrum persyaratan yang memandu desain. Menambahkan fitur untuk pemula, seperti penjelasan, dan fitur untuk para ahli, seperti pintasan dan kecepatan yang lebih cepat, dapat memperkaya desain antarmuka dan meningkatkan kualitas sistem yang dirasakan.

3. Tawarkan umpan balik yang informatif

Untuk setiap tindakan pengguna, harus ada umpan balik sistem. Untuk tindakan yang sering dan kecil, responnya bisa sederhana, sedangkan untuk tindakan yang jarang dan besar, responnya harus lebih substansial. Presentasi visual dari objek yang menarik menyediakan lingkungan yang nyaman untuk menunjukkan perubahan secara eksplisit.

4. Desain dialog untuk menghasilkan penutupan

Urutan tindakan harus diatur ke dalam kelompok dengan awal, tengah, dan akhir. Umpan balik informatif pada penyelesaian sekelompok tindakan memberi operator kepuasan pencapaian, rasa lega, sinyal untuk membatalkan rencana kontingensi dari pikiran mereka, dan sinyal untuk mempersiapkan kelompok tindakan berikutnya.

5. Mencegah kesalahan

Sebisa mungkin, rancang sistem sedemikian rupa sehingga pengguna tidak dapat membuat kesalahan yang serius. Misalnya, item menu abu-abu yang tidak sesuai dan tidak mengizinkan karakter alfabet dalam bidang entri numerik. Jika pengguna membuat kesalahan, antarmuka harus mendeteksi kesalahan dan menawarkan instruksi sederhana, konstruktif, dan spesifik untuk pemulihan. Misalnya, pengguna tidak harus mengetik ulang seluruh formulir nama-alamat jika mereka memasukkan kode pos yang tidak valid, melainkan harus dipandu untuk memperbaiki hanya bagian yang salah. Tindakan yang salah harus

membiarkan status sistem tidak berubah, atau antarmuka harus memberikan instruksi tentang memulihkan status.

6. Izinkan pembalikan tindakan dengan mudah

Sebisa mungkin, tindakan harus dapat dibalik. Fitur ini mengurangi kecemasan, karena pengguna tahu bahwa kesalahan dapat diatasi, sehingga mendorong eksplorasi opsi yang tidak dikenal. Unit pembalikan mungkin satu tindakan, tugas entri data, atau kelompok tindakan lengkap, seperti entri nama dan blok alamat.

7. Mendukung lokus kontrol internal

Operator yang berpengalaman sangat menginginkan perasaan bahwa mereka bertanggung jawab atas antarmuka dan bahwa antarmuka merespons tindakan mereka. Tindakan antarmuka yang mengejutkan, urutan entri data yang membosankan, ketidakmampuan untuk memperoleh atau kesulitan dalam memperoleh informasi yang diperlukan, dan ketidakmampuan untuk menghasilkan tindakan yang diinginkan semuanya membangun kecemasan dan ketidakpuasan.

8. Mengurangi beban memori jangka pendek

Batasan pemrosesan informasi manusia dalam memori jangka pendek (aturan praktisnya adalah bahwa manusia dapat mengingat "tujuh plus atau minus dua potongan" informasi) mengharuskan tampilan dibuat sederhana, tampilan multi-halaman dikonsolidasikan, jendela frekuensi gerak berkurang, dan waktu pelatihan yang cukup dialokasikan untuk kode, mnemonik, dan urutan tindakan. Jika sesuai, akses online ke bentuk sintaks perintah, singkatan, kode, dan informasi lain harus disediakan.

2.4.2 Pseudocode

Pseudocode berarti kode yang mirip dengan pemrograman sebenarnya. *Pseudocode* berasal dari kata *Pseudo* yang berarti imitasi, mirip atau menyerupai dengan kode bahasa pemrograman. *Pseudocode* biasanya ditulis dengan huruf kapital, sedangkan komentar atau variabel ditulis dengan huruf kecil [17]. *Pseudocode* ditulis berbasiskan bahasa pemrograman yang akan digunakan,

misalnya C# dan lain-lain, sehingga lebih tepat digunakan untuk menggambarkan algoritma yang akan dikomunikasikan kepada *programmer.*

Dalam *pseudocode*, tidak ada sintaks standar yang resmi. Karena itu, pseudocode ini dapat diterapkan dalam berbagai bahasa pemrograman. Disarankan untuk menggunakan *keyword* yang umum digunakan seperti *: if, then, else, while, do, repeat, for,* dan lainnya. Keuntungan menggunakan notasi pseudocode adalah kemudahan mentranslasi ke notasi bahasa pemrograman, karena terdapat korespondensi antara setiap pseudocode dengan notasi bahasa pemrograman.

Oleh karena itu, *pseudocode* dapat didefinisikan sebagai bahasa tidak formal yang membantu *programmer* dalam mengembangkan algoritma tanpa harus menentukan batasan-batasan pada bahasa C#.

```
Algoritma EUCLIDEAN

{dibaca dua buah bilangan bulat tak-negatif m dan n (m
≠n). Carilah pembagi bersama terbesar, pbt, dari kedua
bilangan tersebu, yaitu bilangan bulat positif
terbesar yang habis membagi m dan n. }

DEKLARASI:
  m, n : integer  { bilangan bulat yang akan dicari
pbt-nya}
  r    : integer  { sisa hasil bagi }

DESKRIPSI:
   read(m,n) { m ≠n}
   while n ≠ 0 do
     r ← m MOD n { hitung sisa hasil pembagian }
     m ← n
     n ← r
   endwhile
   { kondisi selesai pengulangan: n = 0, maka pbt(m,n)
= m }
   write(m)
```

Gambar 2. 2 Contoh Pseudocode (Yuslena Sari, 2017)

2.4.3 Basis Data

Basis data adalah kumpulan data yang disimpan dan dapat digunakan oleh perusahaan atau organisasi tertentu. Sistem manajemen basis data (DBMS) terdiri dari kumpulan data yang saling terkait dan sekumpulan program untuk mengakses data

tersebut. Tujuan utama dari DBMS adalah untuk menyediakan lingkungan yang nyaman dan efisien untuk digunakan dalam mengambil dan menyimpan informasi *database*.

Sebagai satu kesatuan istilah, basis data (*database*) sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti [18] :

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan yang tidak perlu untuk memenuhi berbagai kebutuhan.
3. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis. Untuk selanjutnya di tugas akhir ini, akan menggunakan istilah Tabel (*Table*), sebagai komponen utama pembangunan basis data.

Sistem *database* dirancang untuk mengelola banyak informasi. Sistem *database* harus menyediakan keamanan informasi yang disimpan, meskipun sistem *crash* atau upaya akses yang tidak sah. Sistem juga harus menjaga konsistensi data yang dibagikan pada lebih dari satu pengguna.

2.4.4 Kamus Data

Kamus data tidak menggunakan notasi grafis sebagaimana halnya DFD. Kamus data adalah deskripsi formal mengenai seluruh elemen yang tercakup dalam Diagram Aliran Data. Kamus data berfungsi membantu pelaku sistem untuk memahami aplikasi secara detail, kamus data mengorganisasi semua elemen data yang digunakan dalam sistem dengan presisi yang sedemikian rupa sebagai pemakai dan penganalisis sistem memiliki dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses.

Selain menyediakan dokumentasi dan menghilangkan redundansi, kamus data dapat digunakan untuk [3]:

1. Memvalidasi diagram aliran data untuk kelengkapan dan keakuratannya.
2. Memberikan titik awal untuk mengembangkan layar dan laporan.
3. Menentukan isi data yang disimpan dalam file.
4. Mengembangkan logika untuk proses diagram aliran data.

5. Membuat XML (bahasa *markup* yang dapat diperluas).

Kamus data dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem. Pada tahap analisis sistem, kamus data dapat digunakan sebagai alat komunikasi antara analis sistem dapat memakai sistem tentang data yang mengalir yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem. Pada tahap perancangan sistem, kamus data digunakan untuk merancang *input*, merancang laporan-laporan dan *database*. Kamus data dibuat berdasarkan arus data yang ada di DFD.

Kamus data mendefinisikan elemen-elemen data dengan fungsi sebagai berikut:

1. Menjelaskan arti aliran data dan penyimpanan data dalam DFD.
2. Mendefinisikan komposisi paket data yang bergerak melalui aliran (misalnya alamat diuraikan menjadi kota, negara dan kode pos).
3. Mendeskripsikan komposisi penyimpanan data.
4. Menspesifikasikan nilai dan satuan yang relevan bagi penyimpanan dan aliran.
5. Mendeskripsikan hubungan detail antar penyimpanan (yang akan menjadi titik perhatian dalam *entity-relationship diagram*).

Notasi yang umumnya digunakan dalam menganalisis sistem dengan menggunakan sejumlah simbol yaitu:

Tabel 2. 1 Simbol-simbol Kamus Data

No.	Simbol	Uraian
1.	=	Terdiri dari, mendefinisikan, diuraikan menjadi
2.	+	Dan
3.	**( )**	Menunjukkan suatu elemen yang bersifat pilihan (opsional). Elemen-elemen yang bersifat pilihan ini bisa dikosongkan pada layar masukan atau bisa juga dengan memuat spasi atau nol untuk *field numeric* pada struktur file.
4.	**{ }**	Menunjukkan elemen-elemen yang berulang, juga disebut kelompok berulang atau tabel-tabel. Kemungkinan bisa ada satu atau beberapa elemen berulang di dalam kelompok tersebut. Kelompok berulang bisa mengandung keadaan-keadaan tertentu, seperti misalnya jumlah pengulangan yang

		pasti atau batas tertinggi dan batas terendah untuk jumlah pengulangan.
5.	**[ ]**	Menunjukkan salah satu dari dua atau lebih situasi tertentu. Satu elemen bisa ada sedangkan elemen lainnya juga ada, tetapi tidak bisa keduanya ada secara bersamaan. Elemen-elemen yang ada di dalam tanda kurung ini saling terpisah satu sama lain (dengan kata lain, memiliki salah satu dari jumlah alternatif, seleksi).
6.	|	Pemisah sejumlah alternatif pilihan antara simbol [ ].
7.	**@**	Identifikasi atribut kunci.
8.	**	Komentar.

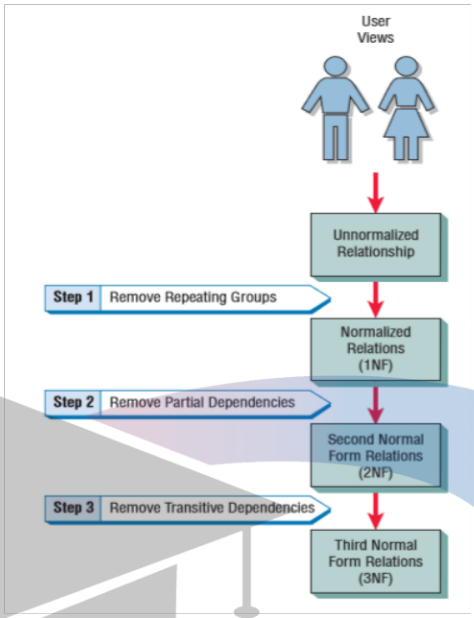
2.4.5 Normalisasi

Normalisasi adalah proses pengelompokan data untuk entitas data yang sederhana, tidak redundan, fleksibel, dan dapat disimpan ke satu set struktur data yang lebih kecil dan stabil. Selain lebih sederhana dan lebih stabil, struktur data yang dinormalisasi lebih mudah dirawat daripada struktur data lainnya [3].

Dimulai dengan tampilan pengguna atau penyimpanan data yang dikembangkan untuk kamus data, analis menormalkan struktur data dalam tiga langkah, setiap langkah melibatkan prosedur penting yang menyederhanakan struktur data.

Langkah pertama dari proses ini adalah menghapus semua kelompok berulang dan mengidentifikasi kunci utama. Kunci utama adalah suatu nilai dalam basis data yang digunakan untuk mengidentifikasi suatu baris dalam tabel dan memiliki nilai unik. Untuk melakukannya, hubungan tersebut perlu dipecah menjadi dua atau lebih. Pada titik ini, hubungan tersebut mungkin sudah dari bentuk normal ketiga, tetapi kemungkinan langkah lebih lanjut akan diperlukan untuk mengubah hubungan ke bentuk normal ketiga. Langkah kedua memastikan bahwa semua atribut bukan kunci sepenuhnya tergantung pada kunci utama. Semua dependensi parsial dihapus dan ditempatkan di hubungan lain. Langkah ketiga menghapus semua dependensi transitif. Ketergantungan transitif adalah atribut di mana atribut bukan kunci bergantung pada atribut bukan kunci lainnya.

Berikut gambar tahap normalisasi :



Gambar 2. 3 Gambar Normalisasi (K. E. Kendall dan J. E. Kendall, 2020)

Proses normalisasi menyederhanakan semua kekompleksan item data yang sering ditemukan dalam tinjauan pemakai. Pada proses normalisasi, mengubah data ke dalam bentuk akhir yang dikenal sebagai Bentuk Normalisasi Ketiga (3NF) mengikuti urutan-urutan sebagai berikut:

1. Bentuk Normalisasi Pertama (1NF)

Langkah pertama dalam menormalkan suatu hubungan adalah menghapus grup berulang. Sebagai contoh, hubungan *SALES-REPORT* yang tidak dinormalisasi akan dipecah menjadi dua hubungan yang terpisah. Hubungan baru ini akan dinamai *SALESPERSON* dan *SALESPERSON-CUSTOMER*.

Gambar 2.4 menunjukkan bagaimana hubungan asli, yang tidak dinormalisasi, *SALES-REPORT* dinormalisasi dengan memisahkan hubungan menjadi dua hubungan baru. Hubungan *SALESPERSON* berisi kunci utama *SALESPERSON-NUMBER* dan semua atribut yang tidak diulang (*SALESPERSON-NAME* dan *SALES-AREA*).

Hubungan kedua, *SALESPERSON-CUSTOMER*, berisi kunci utama dari relasi *SALESPERSON* (kunci utama *SALESPERSON* adalah *SALESPERSON-NUMBER*), serta semua atribut yang merupakan bagian dari grup berulang (*CUSTOMER-NUMBER*, *CUSTOMER-NAME*, *WAREHOUSE-NUMBER*, *WAREHOUSE-LOCATION*, dan *SALES-AMOUNT*). Mengetahui *SALESPERSON-NUMBER*, bagaimanapun tidak secara otomatis kita akan mengetahui *CUSTOMER-NAME,*

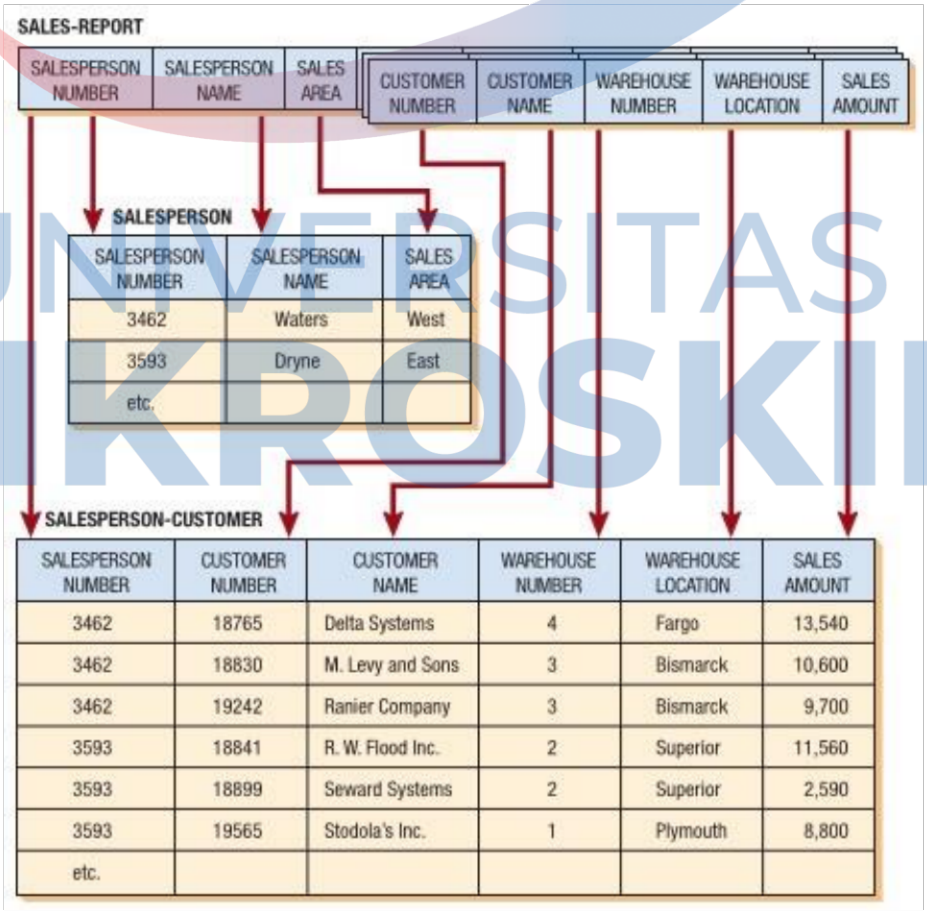
SALES-AMOUNT, WAREHOUSE-LOCATION, dan sebagainya. Dalam hubungan ini, harus menggunakan kunci gabungan (baik *SALESPERSON-NUMBER* dan *CUSTOMER-NUMBER*) untuk mengakses informasi lainnya. Hal ini dimungkinkan untuk menulis hubungan dalam notasi singkat sebagai berikut:

SALESPERSON (*SALESPERSON NUMBER*, *SALESPERSON-NAME, SALES AREA*)

dan

SALESPERSON-CUSTOMER (*SALESPERSON-NUMBER, CUSTOMER-NUMBER, CUSTOMER-NAME, WAREHOUSE-NUMBER, WAREHOUSE-LOCATION, SALES-AMOUNT*)

Berikut gambar normalisasi 1NF [3] :



Gambar 2. 4 Normalisasi (1NF) (K. E. Kendall dan J. E. Kendall, 2020)

2. Bentuk Normalisasi Kedua (2NF)

Dalam bentuk normal kedua, semua atribut akan secara fungsional tergantung pada kunci utama. Oleh karena itu, langkah selanjutnya adalah menghapus semua atribut yang sebagian tergantung dan menempatkannya di relasi lain. Gambar 2.5 menunjukkan bagaimana hubungan *SALESPERSON-CUSTOMER* dipecah menjadi dua hubungan baru: *SALES* dan *CUSTOMER-WAREHOUSE*. Hubungan-hubungan ini dapat dinyatakan sebagai berikut:

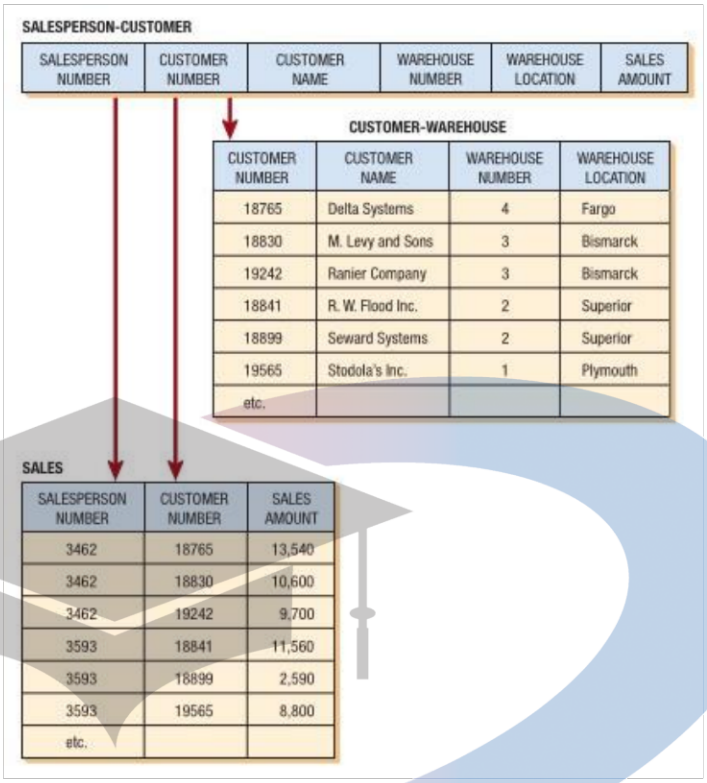
SALES (*SALESPERSON-NUMBER*, *CUSTOMER-NUMBER*, *SALES-AMOUNT*)

dan

CUSTOMER WAREHOUSE (*CUSTOMER-NUMBER*, *CUSTOMER-NAME*, *WAREHOUSE-NUMBER*, *WAREHOUSE-LOCATION*)

Hubungan *CUSTOMER-WAREHOUSE* masih bisa disederhanakan lebih lanjut karena ada dependensi tambahan dalam relasi. Beberapa atribut bukan kunci tidak hanya bergantung pada kunci utama, tetapi juga pada atribut bukan kunci. Ketergantungan ini disebut sebagai ketergantungan transitif.

Bentuk normalisasi kedua (2NF) yang dihasilkan lebih jelas dilihat pada Gambar 2.5 [3]:



Gambar 2. 5 Normalisasi (2NF) (K. E. Kendall dan J. E. Kendall, 2020)

3. Bentuk Normalisasi Ketiga (3NF)

Hubungan yang dinormalisasi ada dalam bentuk normal ketiga jika semua atribut non-kunci sepenuhnya bergantung secara fungsional pada kunci primer dan tidak ada dependensi transitif (non-kunci). Dengan cara yang mirip dengan langkah-langkah sebelumnya, adalah mungkin untuk memecah hubungan *CUSTOMER-WAREHOUSE* menjadi dua hubungan, seperti yang ditunjukkan pada Gambar 2.6.

Dua hubungan baru disebut *CUSTOMER* dan *WAREHOUSE*, dan dapat ditulis sebagai berikut:

CUSTOMER (*CUSTOMER-NUMBER*, *CUSTOMER-NAME*, *WAREHOUSE-NUMBER*)

dan

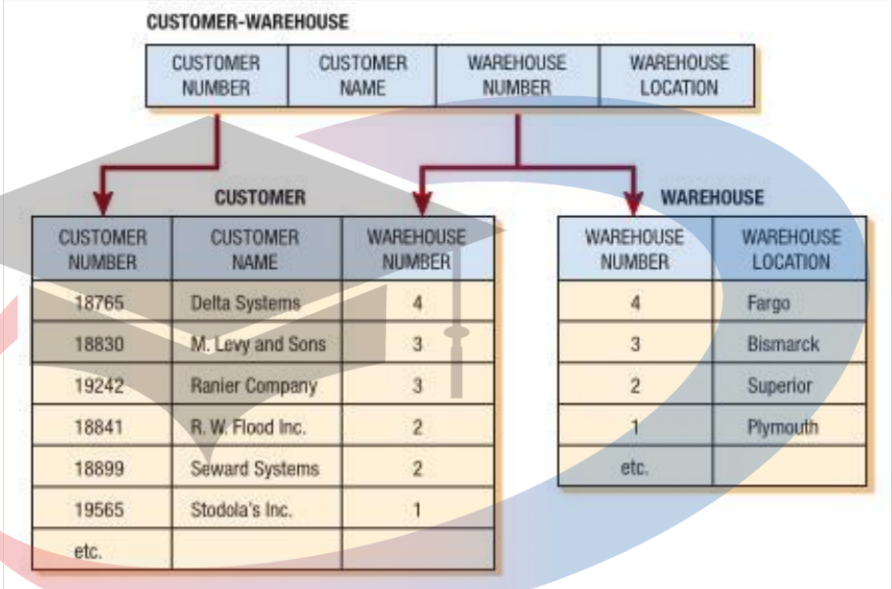
WAREHOUSE (*WAREHOUSE-NUMBER*, *WAREHOUSE-LOCATION*)

Kunci utama untuk hubungan *CUSTOMER* adalah *CUSTOMER-NUMBER*, dan kunci utama untuk hubungan *WAREHOUSE* adalah *WAREHOUSE-NUMBER*.

Selain kunci-kunci utama ini, kita dapat mengidentifikasi *WAREHOUSE-NUMBER* menjadi kunci asing dalam hubungan *CUSTOMER*. Kunci asing adalah

atribut yang non kunci dalam satu relasi tapi kunci utama dalam hubungan lain. Kita menunjuk *WAREHOUSE-NUMBER* sebagai kunci asing dalam notasi sebelumnya dan dalam angka dengan menggarisbawahi dengan garis: _____.

Berikut bentuk normalisasi ketiga (3NF) yang dihasilkan lebih jelas dapat dilihat pada Gambar 2.6 :



Gambar 2. 6 Normalisasi (3NF) (K. E. Kendall dan J. E. Kendall, 2020)

Pada Gambar 2.7, terlihat bahwa hubungan tunggal *SALES-REPORT* diubah menjadi empat hubungan berikut:

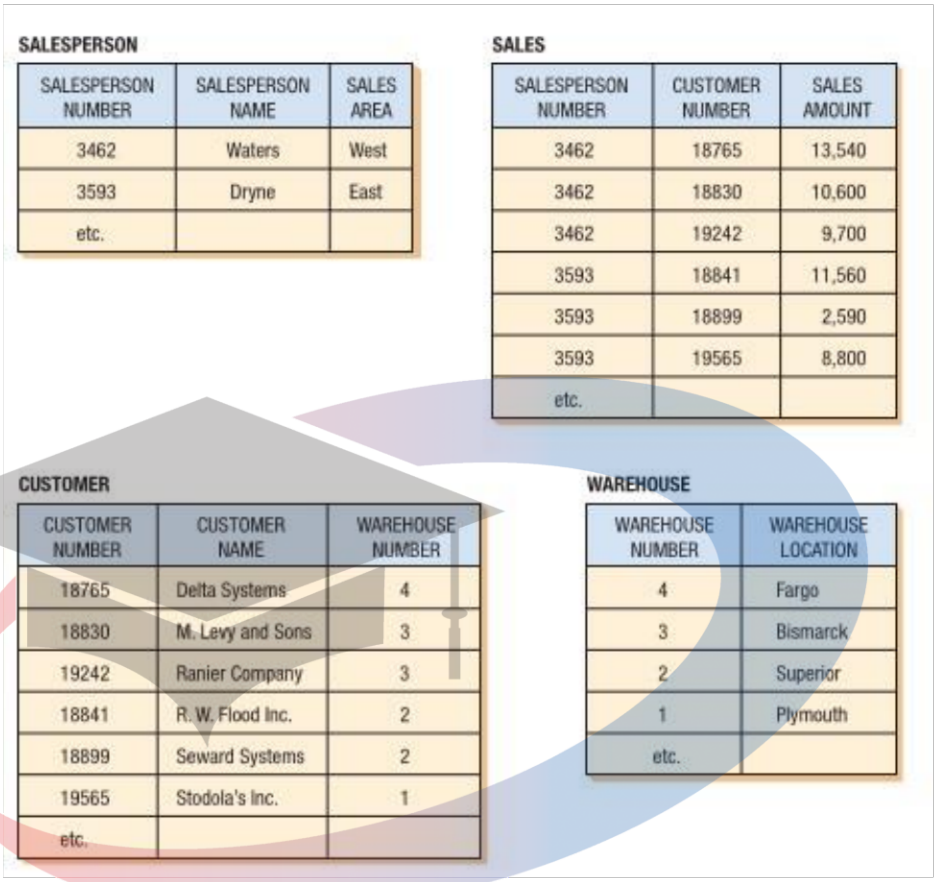
SALESPERSON (*SALESPERSON-NUMBER*, *SALESPERSON-NAME*, *SALES-AREA*)

SALES (*SALESPERSON-NUMBER*, *CUSTOMER-NUMBER*, *SALES-AMOUNT*)

CUSTOMER (*CUSTOMER-NUMBER*, *CUSTOMER-NAME*, *WAREHOUSE-NUMBER*)

dan

WAREHOUSE (*WAREHOUSE-NUMBER*, *WAREHOUSE-LOCATION*)



SALESPERSON

SALESPERSON NUMBER	SALESPERSON NAME	SALES AREA
3462	Waters	West
3593	Dryne	East
etc.		

SALES

SALESPERSON NUMBER	CUSTOMER NUMBER	SALES AMOUNT
3462	18765	13,540
3462	18830	10,600
3462	19242	9,700
3593	18841	11,560
3593	18899	2,590
3593	19565	8,800
etc.		

CUSTOMER

CUSTOMER NUMBER	CUSTOMER NAME	WAREHOUSE NUMBER
18765	Delta Systems	4
18830	M. Levy and Sons	3
19242	Ranier Company	3
18841	R. W. Flood Inc.	2
18899	Seward Systems	2
19565	Stodola's Inc.	1
etc.		

WAREHOUSE

WAREHOUSE NUMBER	WAREHOUSE LOCATION
4	Fargo
3	Bismarck
2	Superior
1	Plymouth
etc.	

Gambar 2. 7 Basis Data Lengkap pada Empat Hubungan (K. E. Kendall dan J. E. Kendall, 2020)

Bentuk normalisasi ketiga cukup untuk sebagian besar masalah desain *database*. Penyederhanaan yang diperoleh dari mengubah sebuah hubungan yang tidak dinormalisasi menjadi seperangkat hubungan 3NF memiliki manfaat ketika untuk memasukkan, menghapus, dan memperbarui informasi dalam *database* lebih terorganisasi.

2.10 Proses bisnis perusahaan dagang secara umum

2.10.1 Penjualan

Penjualan adalah kegiatan sejak diterimanya pesanan dari pembeli, pengiriman barang, pembuatan faktur (penagihan), penerimaan pembayaran dan pencatatan penjualan atau suatu kegiatan yang dilakukan manusia untuk menyampaikan barang kebutuhan yang telah dihasilkan kepada mereka yang memerlukannya dengan imbalan uang menurut harga yang ditentukan [19].

Terdapat dua jenis penjualan yaitu:

1. Transaksi penjualan tunai, barang atau jasa baru diserahkan oleh perusahaan kepada pembeli jika perusahaan telah menerima kas dari pembeli. Kegiatan penjualan secara tunai ini ditangani oleh perusahaan melalui sistem penjualan tunai
2. Transaksi penjualan kredit, jika *order* dari pelanggan telah dipenuhi dengan pengiriman barang atau penyerahan jasa sebelum penjual menerima pembayaran. Kegiatan penjualan secara kredit ini ditangani oleh perusahaan melalui sistem penjualan kredit.

Fungsi-fungsi yang terkait dengan penjualan yaitu [19]:

1. Fungsi penjualan

   Fungsi ini bertanggung jawab untuk menerima pesanan pembelian, mengisi faktur penjualan dan menyerahkan faktur tersebut kepada pembeli untuk kepentingan pembayaran harga barang ke fungsi kas.

2. Fungsi kas

   Fungsi ini bertanggung jawab sebagai penerima kas dari pembeli, mencatat jumlah penerimaan kas, serta menyiapkan laporan penerimaan kas ke fungsi akuntansi.

3. Fungsi gudang

   Fungsi ini bertanggung jawab menyediakan barang yang diperlukan oleh pelanggan sesuai dengan yang tercantum dalam tembusan faktur penjualan yang diterima dari fungsi penjualan.

4. Fungsi pengiriman

   Fungsi ini bertanggung jawab untuk membungkus barang dan menyerahkan barang yang telah dibayar harganya kepada pembeli.

5. Fungsi akuntansi

   Fungsi ini bertanggung jawab sebagai pencatat transaksi penjualan dan sebagai pembuat laporan penjualan.

2.10.2 Pembelian

Sistem pembelian digunakan dalam perusahaan untuk pengadaan barang yang diperlukan oleh perusahaan. Transaksi pembelian dapat digolongkan menjadi dua: pembelian lokal dan impor. Pembelian impor dalam pembelian dari pemasok dalam

negeri, sedangkan impor adalah pemasok dari luar negeri. Pembahasan sistem pembelian ini diterapkan dalam perusahaan dagang sebagai model [19].

Fungsi yang terkait dalam sistem akuntansi pembelian adalah:

1. Fungsi gudang

   Fungsi gudang bertanggung jawab untuk mengajukan permintaan pembelian sesuai dengan posisi persedian yang ada di gudang dan untuk menyimpan barang yang telah diterima oleh fungsi penerimaan.
2. Fungsi pembelian

   Fungsi pembelian bertanggung jawab untuk memperoleh informasi mengenai harga barang, menentukan pemasok yang dipilih dalam pengadaan barang, dan mengeluarkan *order* pembelian kepada pemasok yang dipilih.
3. Fungsi penerimaan

   Dalam sistem akuntansi pembelian, fungsi ini bertanggung jawab untuk melakukan pemeriksaan terhadap jenis, mutu, dan kuantitas barang yang diterima dari pemasok guna menentukan apakah barang tersebut dapat diterima atau tidak oleh perusahaan. Fungsi ini juga bertanggung jawab untuk menerima barang dari pembeli yang berasal dari transaksi retur penjualan.
4. Fungsi akuntansi

   Fungsi akuntansi yang terkait dalam transaksi pembelian adalah fungsi pencatat utang dan fungsi pencatat persediaan. Dalam sistem akuntansi pembelian, fungsi pencatat utang bertanggung jawab untuk mencatat transaksi pembelian ke dalam register bukti kas keluar dan untuk menyelenggarakan kartu utang sebagai buku pembantu utang.

Secara garis besar transaksi pembelian mencakup prosedur berikut ini:

1. Fungsi gudang mengajukan permintaan pembelian ke fungsi pembelian.
2. Fungsi pembelian meminta penawaran harga dari berbagai pemasok.
3. Fungsi pembelian menerima penawaran harga dari berbagai pemasok dan melakukan pemilihan pemasok.
4. Fungsi pembelian membuat *order* pembelian kepada pemasok yang dipilih.
5. Fungsi penerimaan memeriksa dan menerima barang yang dikirim oleh pemasok.

6. Fungsi penerimaan menyerahkan barang yang diterima kepada fungsi gudang untuk disimpan.
7. Fungsi penerimaan melaporkan penerimaan barang kepada fungsi akuntansi
8. Fungsi akuntansi menerima faktur tagihan dari pemasok dan atas dasar faktur dari pemasok tersebut, fungsi akuntansi mencatat kewajiban yang timbul dari transaksi pembelian .

2.10.3 Persediaan

Persediaan adalah suatu istilah umum yang menunjukan segala sesuatu atau sumber daya-sumber daya organisasi yang disimpan dalam antisipasinya terhadap pemenuhan permintaan. Perusahaan dagang biasanya membeli persediaannya dalam bentuk yang sudah siap untuk dijual. Mereka melaporkan harga pokok yang ditetapkan pada unit-unit tersimpan yang belum terjual sebagai persediaan barang dagang.

Untuk menjaga agar stok suatu barang yang diperdagangkan selalu tersedia, biasanya dibuat *buffer stock* (stok penyangga) yang merupakan persediaan barang yang merupakan barang persediaan selama barang pesanan (untuk mengisi gudang) sedang diproses dan dalam pengiriman. Manajemen persediaan mengatur agar perusahaan tidak mengalami kekurangan stok *(out of stock)* dengan melakukan pemesanan ulang barang yang hampir habis ketika jumlah barang yang tersedia kurang dari *buffer stock*.