

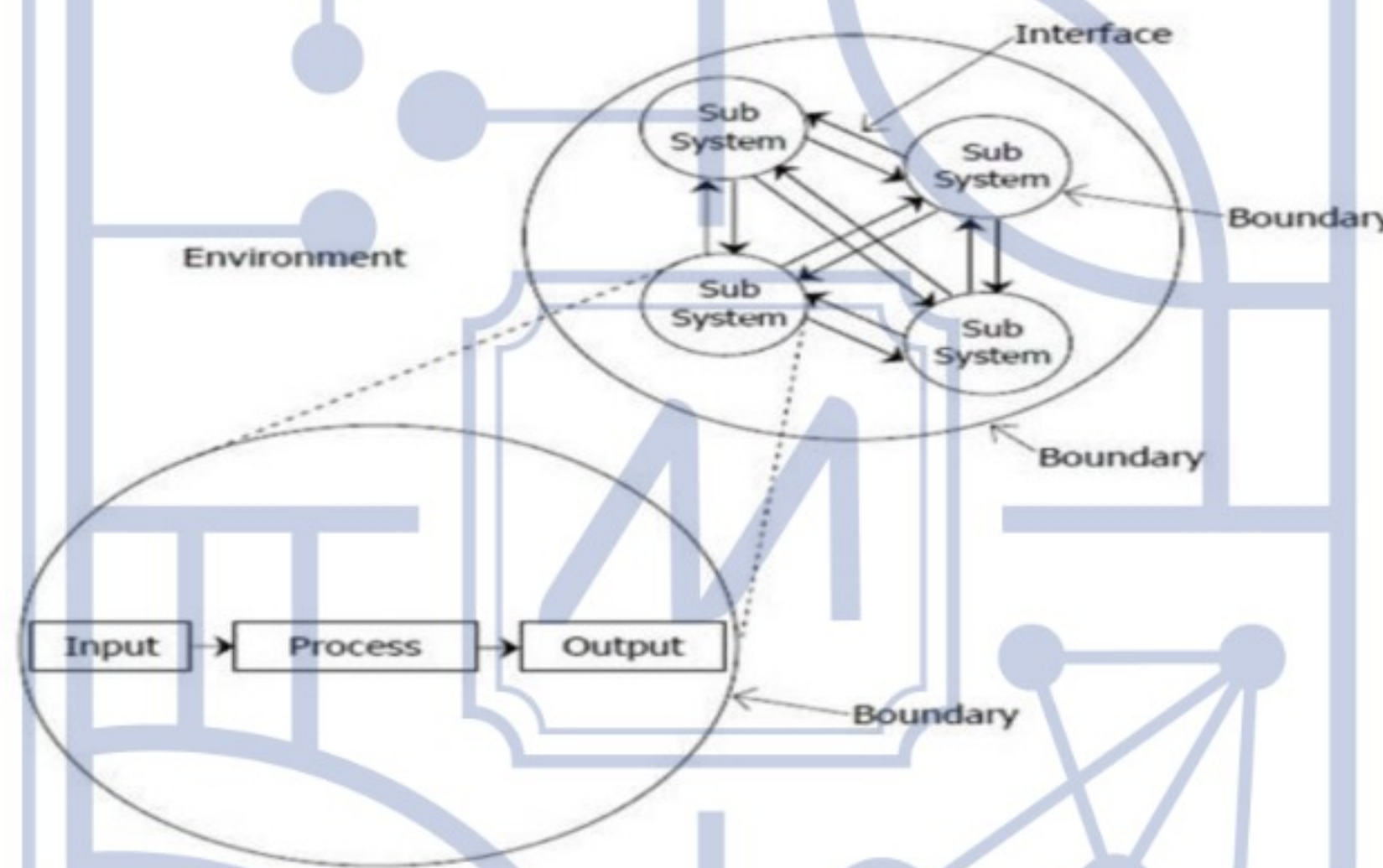
BAB II

KAJIAN LITERATUR

2.1 Konsep Sistem Informasi

2.1.1 Sistem

Sistem merupakan sekumpulan elemen atau komponen yang saling berintegrasi untuk mencapai tujuan tertentu. Fungsi dan interaksi antar komponen tersebut berlangsung secara konsisten karena setiap komponen mempunyai mekanisme kerja yang spesifik. Karakteristik sistem merujuk pada ciri-ciri yang wajib dimiliki oleh suatu sistem.



Gambar 2.1 Karakteristik Sistem

Sistem memiliki karakteristik sebagai berikut [3]:

1. Komponen Sistem (*Component*)

Komponen sistem adalah elemen sistem yang bekerja bersama untuk membentuk satu kesatuan untuk mencapai tujuan tertentu.

2. Lingkungan Luar Sistem (*Environment*)

Dampak proses sistem terhadap lingkungan eksternal merujuk pada lingkungan di luar sistem itu sendiri. Segala hal yang memberikan manfaat harus dijaga, sementara hal-hal yang merugikan dapat dikendalikan. Apabila terjadi perubahan dalam peraturan regulasi sistem, hal tersebut berpotensi memengaruhi perilaku sistem yang sedang berjalan.

3. Batasan Sistem (*Boundary*)

Dalam wilayahnya, ruang lingkup sistem membatasi sistem ke batas sistem lain. Batas - batas suatu sistem menentukan apa yang terkandung di dalam dan di luar sistem.

4. Penghubung Sistem (*Interface*)

Alat yang menghubungkan subsistem ke subsistem lain untuk memfasilitasi aliran data di seluruh sistem.

5. Masukan Sistem (*Input*)

Data yang dimasukkan ke sistem untuk diproses dan diolah disebut masukan sistem.

6. Keluaran Sistem (*Output*)

Keluaran sistem adalah data yang dibuat setelah sistem diproses dan dioperasikan dan merupakan masukan ke dalam subsistem lain.

7. Pengelola Sistem (*Process*)

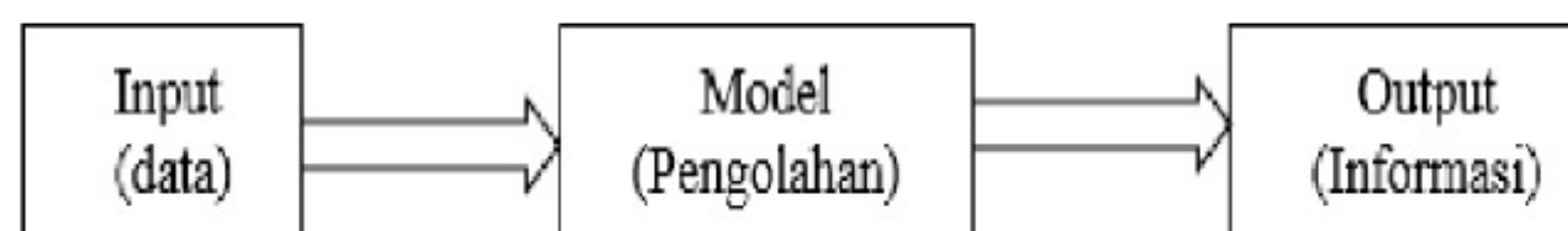
Manajemen sistem mencakup semua proses yang terjadi dalam sistem dan merupakan proses mengonversi atau memanipulasi data masukan menjadi data keluaran.

8. Sasaran Sistem (*Objective*)

Sasaran sistem adalah tujuan atau hasil yang ingin dicapai sistem. Setelah mencapai tujuan, sistem dianggap berhasil.

2.1.2 Informasi

Informasi adalah hasil dari pengolahan data yang memberikan bentuk pada pemikiran dan kemudian berkembang menjadi makna yang lebih luas, sehingga memiliki nilai guna yang lebih tinggi bagi penerimanya. Sumber informasi terdiri dari data yang menjelaskan peristiwa yang terjadi pada waktu tertentu. Sumber tersebut harus melalui proses yang dinamakan siklus pengelolaan data. informasi yang bernilai adalah informasi yang dapat dimanfaatkan untuk membuat keputusan yang tepat [4].



Gambar 2.2 Siklus Pengolahan Data

Informasi memiliki karakteristik utama yang menghilangkan ketidakpastian, menyediakan konteks, serta meningkatkan pemahaman terhadap suatu hal. Fungsi dari informasi adalah sebagai berikut [3]:

- a. Mendukung proses pengambilan keputusan dengan menyediakan pengetahuan yang diperlukan untuk mengambil keputusan yang tepat.
- b. Menyediakan fakta dan data yang dapat mengurangi keraguan.
- c. Membantu mengevaluasi kinerja dan memastikan tujuan organisasi tercapai.
- d. Menjadi dasar menciptakan produk atau proses baru yang lebih baik.
- e. Meningkatkan koordinasi dalam organisasi atau antar individu.
- f. Membangun dan memperkuat hubungan antar individu melalui pertukaran informasi.

Informasi memiliki kualitas dan karakteristik sebagai berikut [4]:

1. Akurasi (*Accuracy*)

Informasi haruslah tepat, tidak memihak, serta bebas dari kesalahan yang dapat menyesatkan. Keakuratan mengandung makna bahwa informasi yang disampaikan harus jelas, benar, bermakna, dan tidak menimbulkan ambiguitas saat diterima oleh pihak yang menerima informasi tersebut.

2. Tepat waktu (*Timeliness*)

Informasi tidak boleh terlambat sampai kepada penerima. Informasi yang sudah usang didefinisikan sebagai informasi yang tidak lagi relevan untuk digunakan dalam pengambilan keputusan yang tepat. Penyampaian informasi harus dilakukan secara efisien dan penerima harus mampu memperoleh informasi dengan cepat, sehingga diperlukan teknologi terkini guna memastikan informasi yang disampaikan merupakan informasi terbaru.

3. Relevan (*Relevance*)

Informasi yang berkualitas adalah informasi yang bermanfaat bagi penerimanya. Relevansi informasi muncul ketika terdapat perbedaan antara satu individu dengan individu lainnya.

2.1.3 Sistem Informasi

Sistem informasi adalah himpunan dan integrasi teknologi serta proses yang mengumpulkan, memproses, dan menyebarkan informasi yang berguna untuk mendukung

pengambilan keputusan yang akurat. Sistem informasi terdiri dari tiga komponen utama, yakni [5]:

1. Perangkat Keras (*Hardware*)

Perangkat keras merupakan komponen fisik yang memiliki peranan penting dalam mendukung proses pengolahan data. Perangkat keras berfungsi sebagai media tempat berlangsungnya proses pengolahan data tersebut. Perangkat keras meliputi alat-alat seperti komputer, printer, prosesor, monitor, dan lain sebagainya.

2. Perangkat Lunak (*Software*)

Perangkat lunak adalah program yang digunakan untuk mengelola data dalam suatu sistem. Perangkat lunak memiliki kemampuan untuk memberikan instruksi kepada perangkat keras agar dapat memproses data tertentu menjadi informasi.

3. Manusia (*Brainware*)

Brainware merupakan kemampuan otak manusia sebagai pengguna akhir dalam menjalankan, mengelola, dan mengembangkan sebuah sistem informasi.

Sistem informasi memiliki lima jenis berbeda yang masing – masing jenis memiliki peran dan tujuan sendiri [5]:

1. Sistem Informasi Manajemen (*Management Information System*)

Sistem Informasi Manajemen (*MIS*) menyediakan informasi kepada entitas bisnis guna mendukung proses perencanaan, pengelolaan, serta pengambilan keputusan. *MIS* dirancang untuk menyajikan informasi yang relevan dan akurat, yang berfungsi sebagai dasar dalam pengelolaan data mengenai perusahaan Anda demi pengambilan keputusan strategis. Contoh penerapan *MIS* adalah sistem pelaporan keuangan.

2. Sistem Pendukung Keputusan (*Decision Support System*)

Sistem Pendukung Keputusan (*DSS*) berperan dalam menyelesaikan permasalahan yang bersifat semi-terstruktur maupun tidak terstruktur dalam proses pengambilan keputusan. *DSS* mengintegrasikan data, model analitik, serta antarmuka guna menyediakan informasi yang mendukung pengambilan keputusan secara akurat dan tepat waktu. Contoh penerapan *DSS* meliputi pengelolaan rantai pasok dan manajemen risiko.

3. Sistem Informasi Geografis (*Geographic Information System*)

GIS digunakan untuk mengumpulkan, menyimpan dan menganalisis data lokasi geografis dan kemudian divisualisasikan untuk pemahaman yang lebih mudah oleh pengguna. Contoh dari penggunaan *GIS* adalah perencanaan transportasi.

4. *Enterprise Resource Planning*

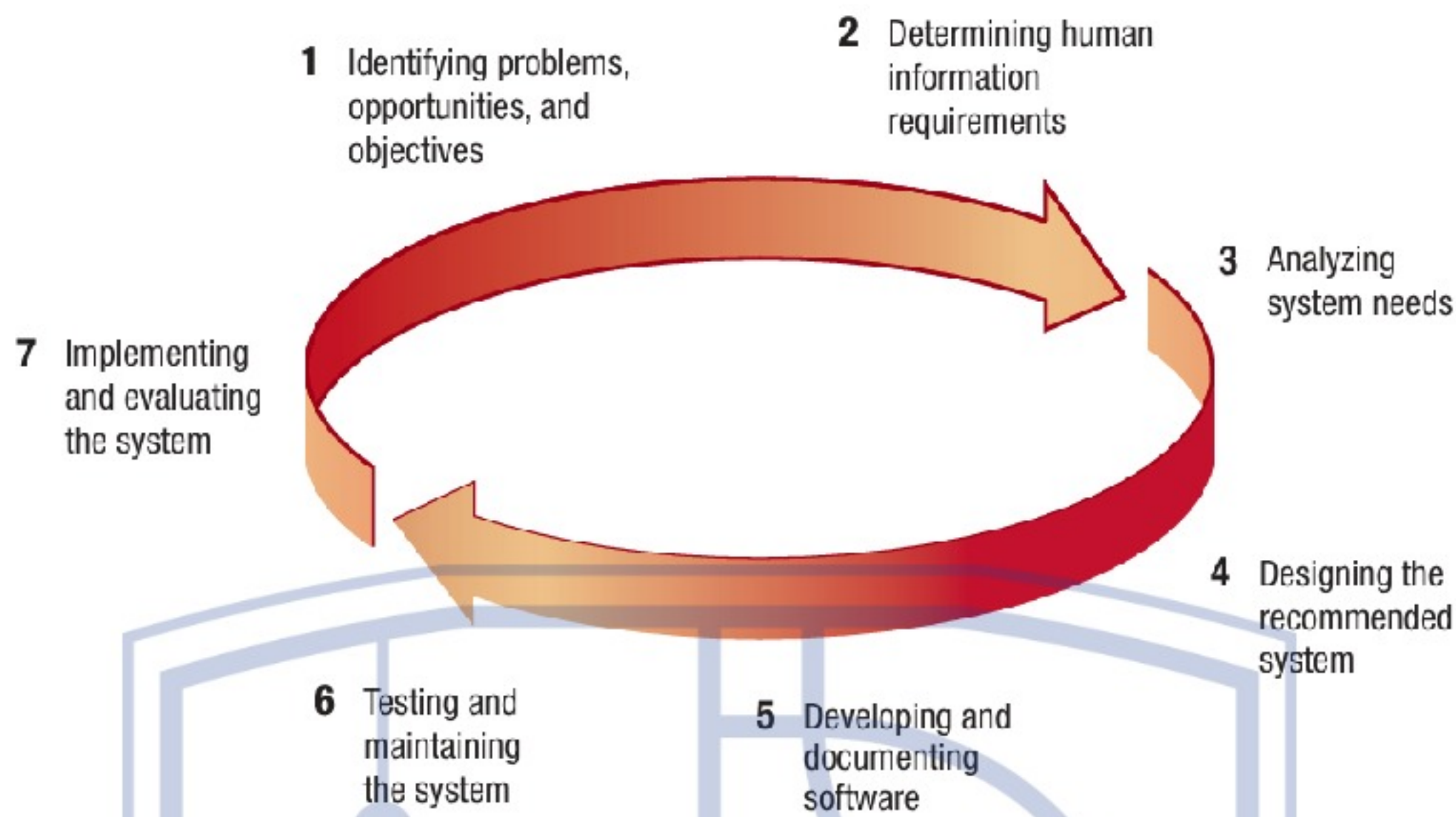
Sistem *ERP* dirancang untuk mengelola seluruh informasi serta proses bisnis dalam suatu organisasi secara menyeluruh. *ERP* mendukung berbagai fungsi bisnis yang beragam, meliputi keuangan, pemasaran, produksi, dan fungsi lainnya yang saling terintegrasi dalam sebuah aplikasi terpusat. Contoh implementasi *ERP* yang umum digunakan adalah *SAP* dan *Oracle*.

5. Sistem Transaksi (*Transaction Processing System*)

Sistem Pemrosesan Transaksi (*TPS*) dirancang untuk mencatat transaksi bisnis secara real time. *TPS* merupakan fondasi utama dalam operasional bisnis yang menitikberatkan pada pelaporan, analisis, serta pengambilan keputusan. Salah satu contoh penerapan *TPS* adalah pada sistem penggajian.

2.2 System Development Life Cycle (SDLC)

System Development Life Cycle (SDLC) merupakan pendekatan bertahap yang sistematis dalam melakukan analisis dan desain sistem. Pendekatan ini didasarkan pada asumsi bahwa sistem akan lebih baik dikembangkan melalui serangkaian aktivitas yang terstruktur. Model ini sering disebut sebagai metode *waterfall* karena prosesnya mengalir ke bawah dari satu tahap ke tahap berikutnya secara berurutan [6].



Gambar 2.3 Siklus Hidup Pengembangan Sistem

Beberapa tahapan yang terdapat dalam *SDLC* tersebut [6]:

1. Identifikasi Masalah, Peluang, dan Tujuan (*Identifying Problems, Opportunities, and Objectives*)

Pada tahap ini, analisis fokus pada mengenali masalah yang ada, peluang untuk meningkatkan, serta tujuan yang ingin dicapai melalui sistem informasi yang terkomputerisasi. Langkah ini sangat penting karena memastikan proyek tidak membuang waktu untuk menyelesaikan masalah yang salah. Hasil dari tahap ini adalah laporan kelayakan (*Feasibility Report*) yang berisi penjelasan masalah dan ringkasan tujuan, yang kemudian digunakan oleh manajemen untuk memutuskan apakah proyek layak dilanjutkan.

2. Menentukan Kebutuhan Informasi Manusia (*Determining Human Information Requirements*)

Analisis menentukan kebutuhan informasi yang diperlukan pengguna dengan beberapa metode seperti wawancara, pengamatan data, dan kuesioner. Pada tahap ini, analisis juga mulai mengeksplorasi aspek interaksi manusia-komputer (*Human Computer Interaction/HCI*), misalnya bagaimana membuat sistem mudah didengar, diterima, dan aman digunakan.

3. Menganalisis Kebutuhan Sistem (*Analyzing System Needs*)

Analisis menggunakan alat seperti diagram alir data (*Data Flow Diagram/DFD*) untuk memetakan input, proses, dan *output* sistem saat ini atau sistem yang akan

dirancang. Selanjutnya, analis menyusun proposal sistem (*Systems Proposal*) yang merangkum hasil analisis, menampilkan analisis biaya-manfaat dari berbagai alternatif, serta memberikan rekomendasi langkah berikutnya.

4. Merancang Sistem yang Direkomendasikan (*Designing the Recommended System*)

Tahap ini berfokus pada desain logis dari sistem informasi yang direkomendasikan. Aktivitas utama mencakup merancang prosedur pengguna, desain input yang efektif (seperti Formulir dan antarmuka), desain *database*, serta desain *output* sesuai dengan kebutuhan pengguna. Desain antarmuka pengguna (*User Interface/UI*) atau *HCI* dilakukan bersama pengguna untuk memastikan sistem yang dihasilkan menarik dan mudah dipakai.

5. Mengembangkan dan Mendokumentasikan Perangkat Lunak (*Developing and Documenting Software*)

Analis bekerja sama dengan pengembang kode untuk mengembangkan perangkat lunak yang diperlukan. Selain itu, analis juga bertugas meng-dokumentasikan *software*, meliputi prosedur penggunaan, logika program, dan bantuan online.

6. Menguji dan Memelihara Sistem (*Testing and Maintaining the System*)

Perangkat lunak yang telah dikembangkan diuji menggunakan berbagai alat untuk menemukan *bug* atau kesalahan. Memelihara sistem merupakan proses yang berkelanjutan untuk memperbaiki kesalahan yang ditemukan serta meningkatkan kemampuan perangkat lunak sesuai dengan perubahan kebutuhan organisasi.

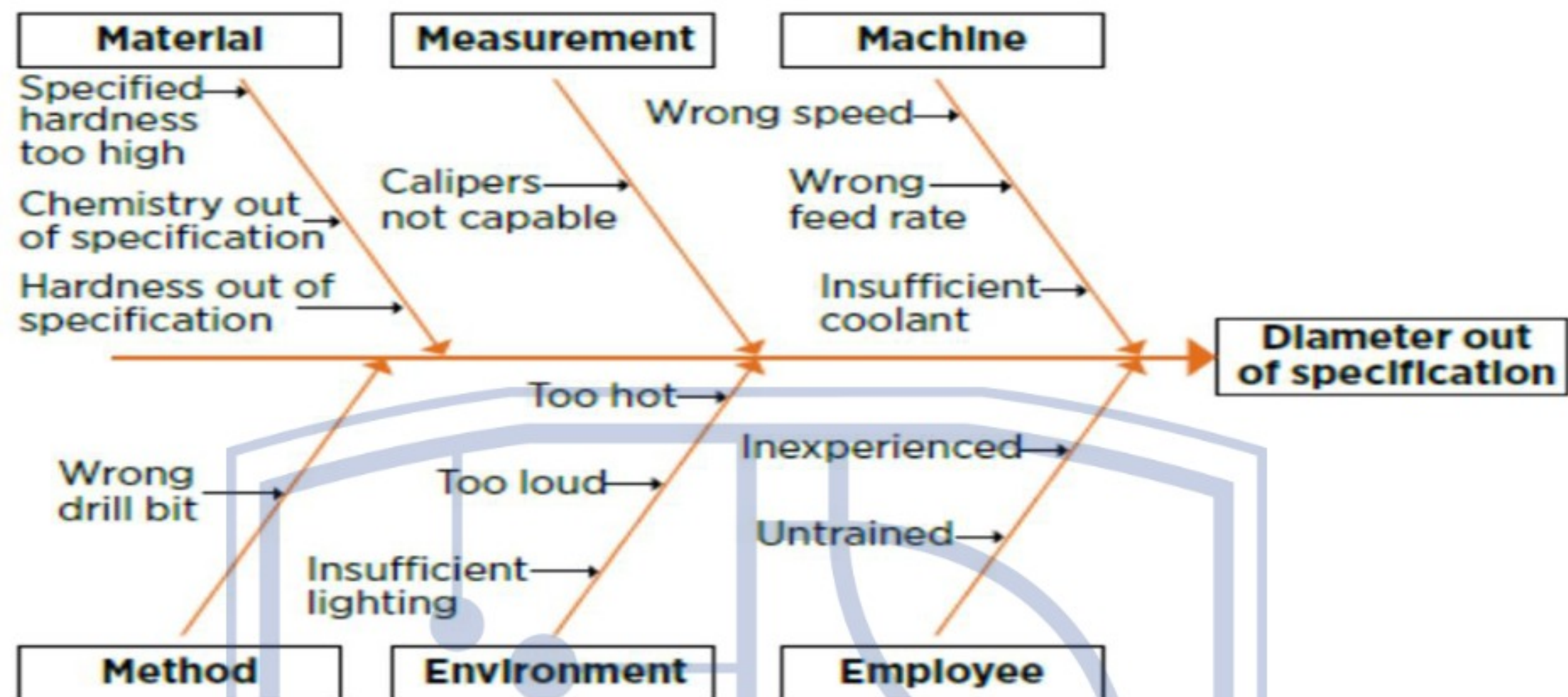
7. Mengimplementasikan dan Mengevaluasi Sistem (*Implementing and Evaluating the System*)

Tahap terakhir ini mencakup pelatihan pengguna, migrasi data dari Format lama ke baru, pemasangan perangkat, serta mengimplementasikan sistem ke tahap produksi. Evaluasi dilakukan sepanjang siklus *SDLC* dan sebagai langkah akhir untuk memastikan sistem benar-benar digunakan oleh pengguna yang dituju.

2.3 Fishbone Diagram

Diagram tulang ikan (*Fishbone Diagram*), yang juga dikenal sebagai diagram sebab akibat (*Cause-and-Effect Diagram*), merupakan alat yang penting dalam proses analisis masalah. Tujuannya adalah untuk mengidentifikasi serta mengelompokkan semua kemungkinan penyebab yang berkontribusi terhadap suatu masalah tertentu [7].

Cause and effect diagram

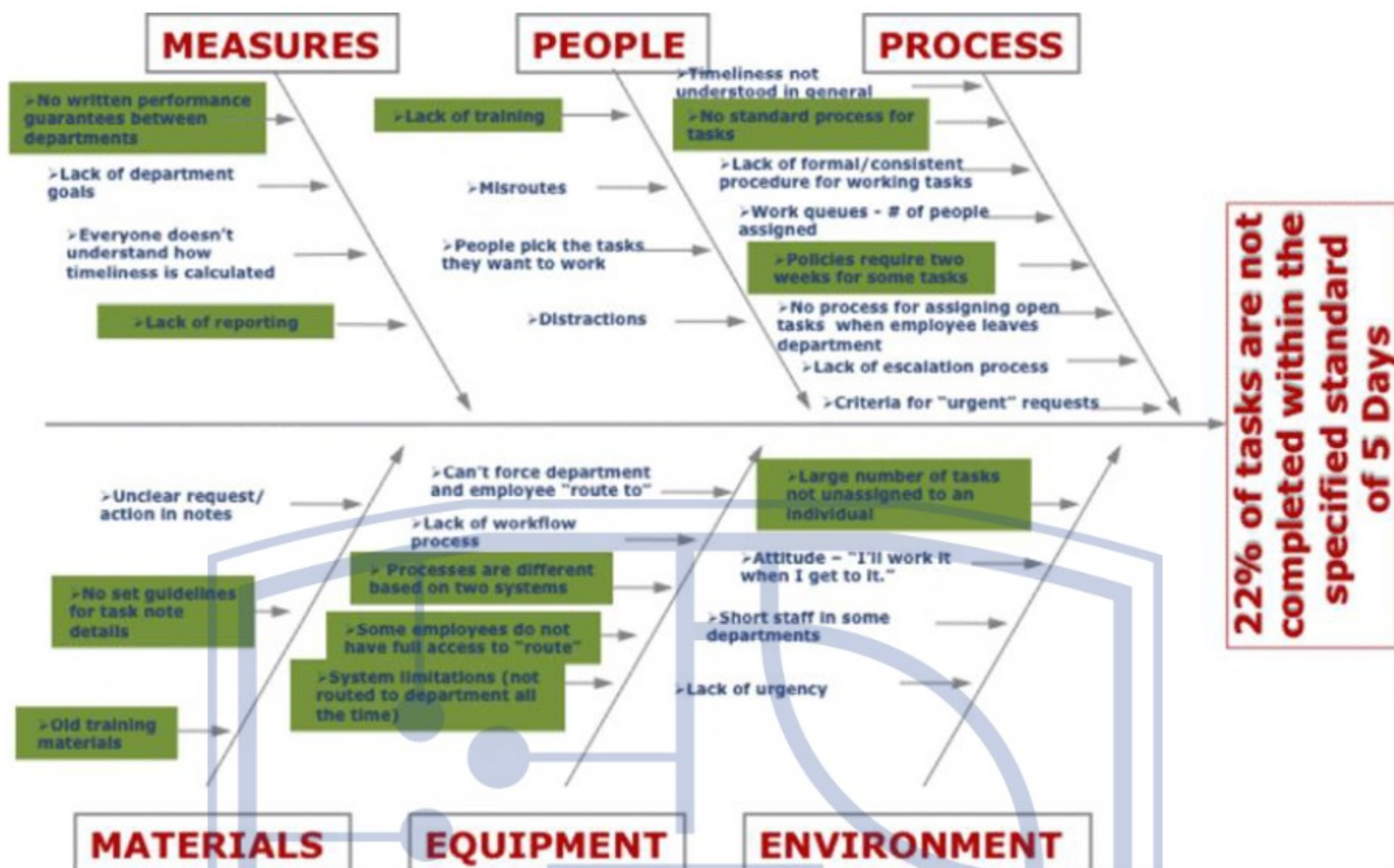


Gambar 2.4 Fishbone Diagram

Fishbone diagram merupakan diagram yang menyerupai kerangka tulang ikan di mana masalah utama ditempatkan pada bagian kepala, sementara penyebab-penyebabnya digambarkan sebagai tulang yang bercabang dari tulang punggung utama. Dalam mengidentifikasi akar penyebab masalah, terdapat 6M yang sering digunakan secara umum, dimana [8]:

1. *Materials* (Material): Mencakup bahan baku, komponen, suku cadang, dan persediaan pendukung.
2. *Machinery* (Mesin/Peralatan): Meliputi mesin produksi, alat penanganan material, hingga perangkat lunak (software).
3. *Methods* (Metode): Berkaitan dengan prosedur, teknik kerja, regulasi, dan proses operasional.
4. *Measurement* (Pengukuran): Melibatkan parameter indikator kunci, perangkat ukur, serta akurasi data yang dikumpulkan.
5. *Manpower* (Sumber Daya Manusia): Menitikberatkan pada aspek manusia, termasuk kompetensi, keterampilan, dan pelatihan yang dimiliki.
6. *Mother Nature / Environment* (Lingkungan): Faktor eksternal seperti kondisi lingkungan kerja dan pengaruh eksternalitas lainnya.

Berikut ini contoh gambar *Fishbone Diagram* dan penjelasannya [8]:



Gambar 2.5 Contoh Fishbone Diagram

Perusahaan ABC menghadapi tantangan dalam memenuhi ekspektasi layanan pelanggan karena adanya keterlambatan dalam merespon pertanyaan atau pengajuan nasabah. Masalah utama kegagalan ini adalah lamanya waktu yang dibutuhkan sejak permintaan diterima hingga akhirnya terselesaikan. Rincian faktor yang mempengaruhi masalah dapat diidentifikasi dalam diagram diatas [7]:

1. *People* (Faktor Manusia)

Analisis mengungkapkan bahwa performa karyawan dipengaruhi oleh beberapa isu manusiawi. Penyebab utamanya meliputi tingkat turnover karyawan yang tinggi yang menyebabkan kurangnya pengalaman di tim, serta kurangnya pelatihan (*training*) yang memadai bagi staf baru. Selain itu, tingkat ketidakhadiran yang tidak terduga juga berkontribusi pada kekurangan tenaga kerja saat puncak beban kerja.

2. *Process* (Faktor Proses)

Dari sisi alur kerja, ditemukan inefisiensi signifikan. Banyak langkah dalam proses pelayanan yang masih dilakukan secara manual, yang rentan terhadap kesalahan dan memakan waktu. Selain itu, adanya langkah-langkah yang redundan atau tidak perlu, serta keharusan melakukan perbaikan berulang kali akibat kesalahan awal, membuat proses menjadi lebih panjang dari yang seharusnya.

3. *Technology* (Faktor Teknologi)

Infrastruktur teknologi informasi teridentifikasi sebagai hambatan serius. Sistem yang digunakan seringkali mengalami down atau penurunan kecepatan, dan sering terjadi sistem *offline* pada saat krusial. Kurangnya integrasi antar perangkat lunak juga membuat data tidak bisa diakses dengan cepat.

4. *Policy* (Kebijakan & Prosedur)

Aspek manajemen dan regulasi internal juga menjadi penyumbang masalah. Kebijakan perusahaan yang mewajibkan persetujuan berlapis untuk kasus sederhana membuat proses berlarut-larut. Selain itu, keterbatasan wewenang yang dimiliki oleh staf garis depan membuat mereka harus menunggu persetujuan atasan untuk keputusan kecil sekalipun.

5. *Environment* (Faktor Lingkungan):

Lingkungan kerja fisik yang tidak kondusif berpengaruh pada konsentrasi staf. Kebisingan berlebihan, pencahayaan yang buruk, serta tata letak kerja yang tidak ergonomis menciptakan distraksi yang akhirnya memperlambat kecepatan kerja individu.

2.4 PIECES

PIECES merupakan kerangka kerja yang digunakan untuk menganalisis dan mengkategorikan suatu masalah, peluang, dan tujuan yang termasuk dalam ruang lingkup yang ada. *PIECES* memiliki enam variabel penting yang digunakan dalam menganalisis suatu sistem [9]:

1. *Performance*

analisis ini dilakukan dengan mengukur kinerja sistem yang ada. Kinerja diukur berdasarkan berapa banyak jumlah *output* yang diproses.

2. *Information*

Analisis ini dilakukan dengan tujuan untuk mengetahui jumlah *output* yang dihasilkan dan keakuratan informasi tersebut.

3. *Economic*

Analisis dilakukan berdasarkan biaya yang dikeluarkan saat sistem dikembangkan.

4. *Control and Security*

Analisis dilakukan untuk mengetahui keamanan pada sistem berfungsi dengan baik atau tidak.

5. *Efficiency*

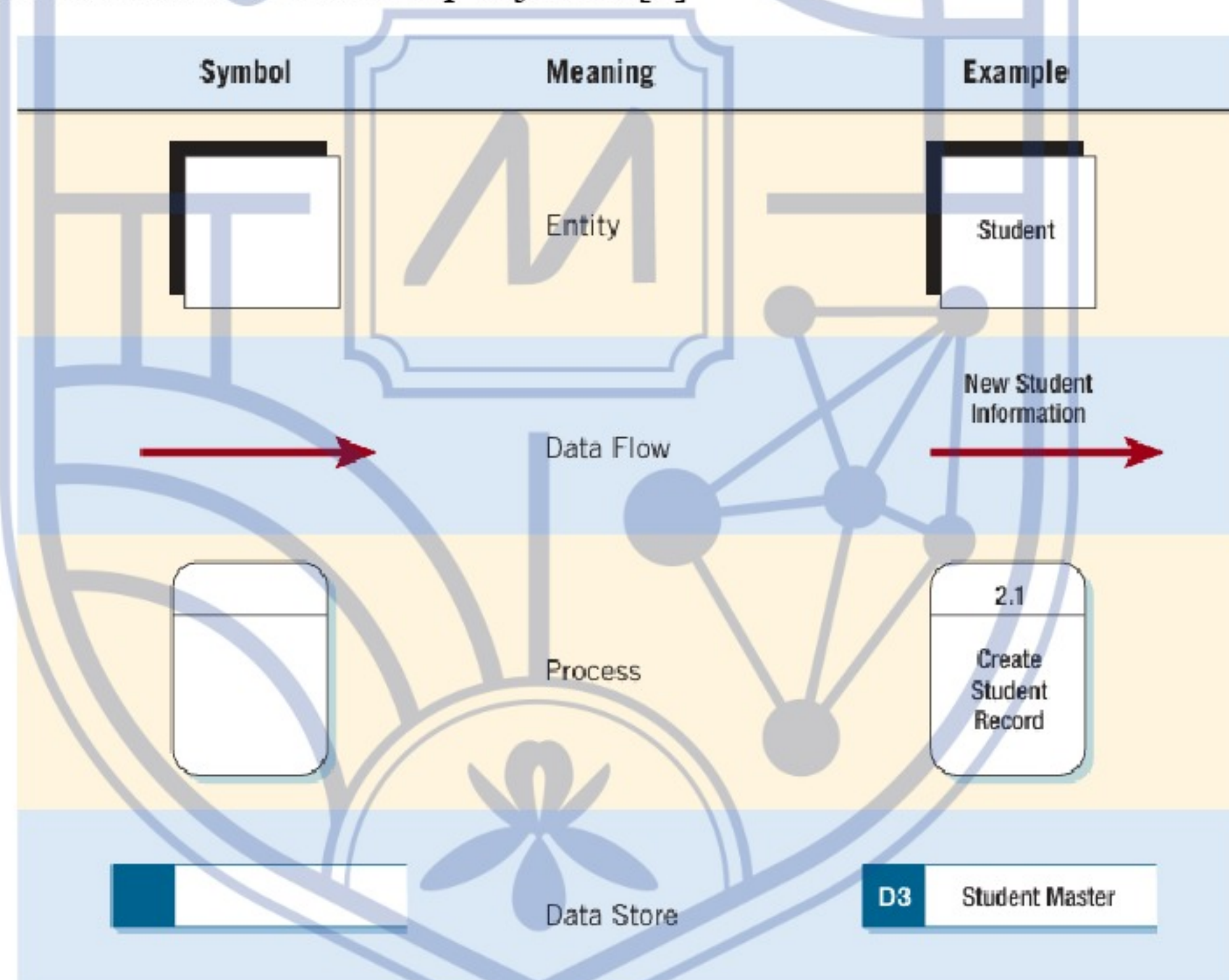
Analisis dilakukan untuk mengetahui apakah sistem mampu menghasilkan *output* yang memuaskan. Sistem yang dikembangkan harus mampu mencapai tujuan awal mengapa sistem dikembangkan.

6. *Service*

Analisis dilakukan untuk mengidentifikasi kesenjangan antara kinerja sistem dengan kebutuhan pengguna untuk memastikan keseimbangan fungsionalitas dan kepuasan pengguna.

2.5 Data Flow Diagram (DFD)

DFD adalah diagram yang menggambarkan proses masuk dan keluarnya sebuah data dalam sistem informasi. *DFD* digunakan agar analis dapat mudah memahami proses bisnis yang terjadi pada sebuah perusahaan. *DFD* berfokus pada proses bisnis dan data yang diperlukan serta dihasilkan dalam setiap kejadian [6]



Gambar 2.6 Simbol DFD

Ketentuan dan simbol dasar penggunaan *DFD* [6]:

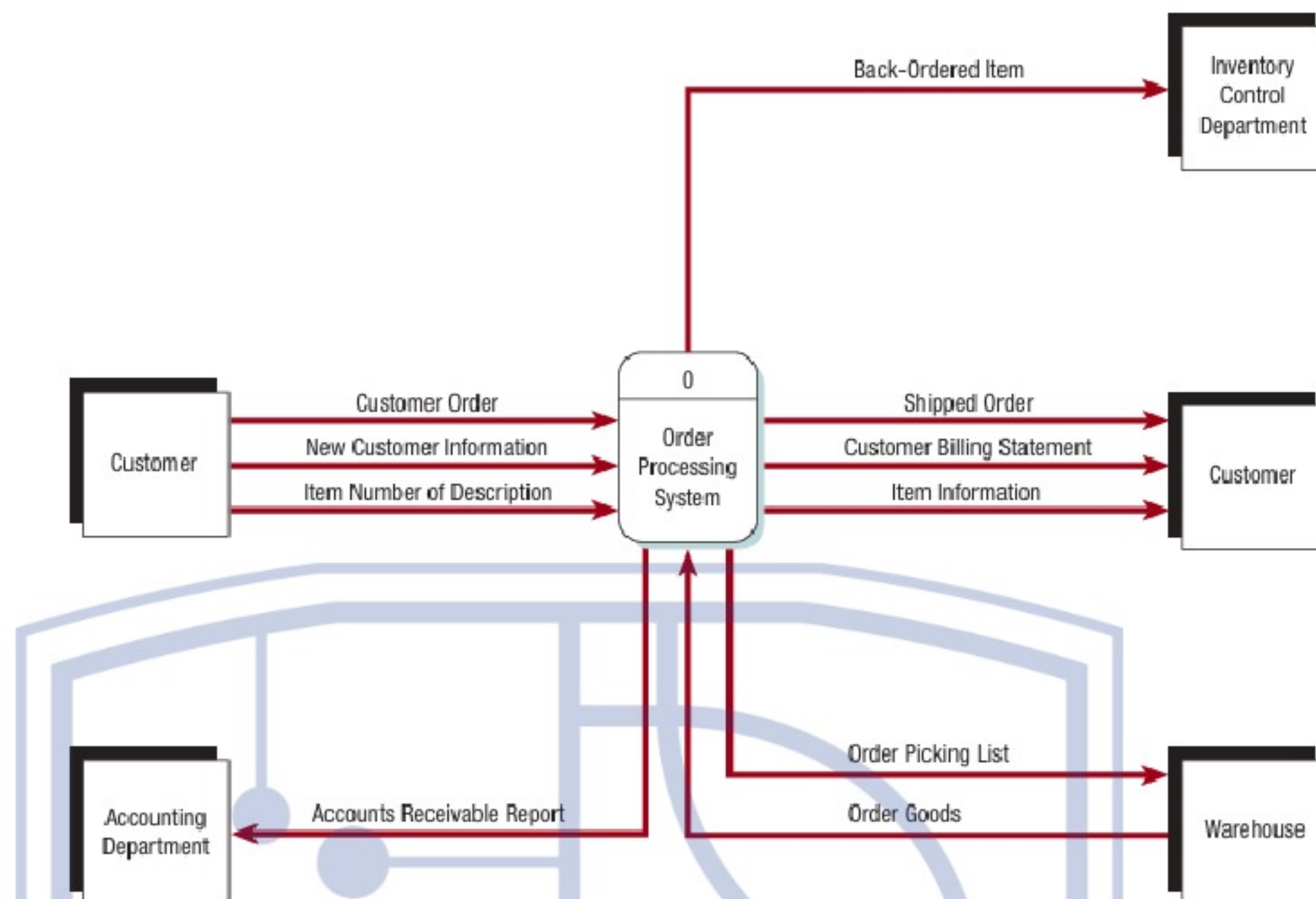
1. *Entity* (Entitas) merupakan entitas yang berperan untuk mengirim dan menerima data dari sistem. *Entity* dapat berupa orang, divisi, atau organisasi. *Entity* biasanya sesuai dengan aktor utama dalam *use case*. *Entity* memberikan data ke dalam sistem atau menerima data dari dalam sistem dan berfungsi untuk menetapkan batasan sistem yang ada. Setiap *entity* memiliki nama dan deskripsi masing-masing.

2. Proses merupakan suatu kegiatan mengubah dan mentransformasi data input menjadi data *output*. Proses bisa dilakukan secara manual maupun secara komputerisasi. Setiap proses harus memiliki setidaknya satu input dan *output*.
3. Aliran data atau arus data menunjukkan perpindahan data yang masuk dan keluar dari sebuah proses. Setiap aliran data harus dinamai. Penjelasan aliran data wajib mencantumkan tepat elemen data apa yang terkandung dalam aliran. Aliran data berfungsi sebagai behubung yang menyatukan entitas dan proses dengan yang lain. Salah satu ujung dari setiap aliran harus selalu berasal dari atau menuju ke proses lain, dengan panah yang menunjuk ke arah masuk atau keluar proses. Aliran data menunjukkan input dan *output* apa yang masuk dan dihasilkan oleh sebuah proses. Setiap proses harus menghasilkan setidaknya satu input dan *output*. Jika tidak ada input dan *output*, berarti proses tersebut tidak melakukan apapun.
4. Penyimpanan data (*Data Store*) merupakan kumpulan data yang disimpan ke dalam *database* yang telah dibentuk. Setiap penyimpanan data selalu diberi nama dan penomoran untuk memudahkan pemahaman diagram. Aliran data yang keluar dari penyimpanan data menandakan bahwa informasi diambil dari penyimpanan data. Aliran yang mengarah ke penyimpanan data menandakan bahwa data ditambahkan dan disimpan ke dalam penyimpanan data. Aliran yang masuk dan keluar tersebut biasanya menandakan bahwa informasi dalam penyimpanan data telah diubah. Semua penyimpanan data harus memiliki setidaknya input, kecuali penyimpanan data dikelola oleh sistem informasi yang berbeda. Penyimpanan data dalam *DFD* harus memiliki setidaknya satu aliran data keluar. Aliran data yang menandakan data disimpan dan data dikeluarkan dari suatu penyimpanan data selalu digambarkan sebagai dua aliran data yang terpisah.

Pendekatan *DFD* dilakukan secara *top-down*, yaitu mulai dari gambaran umum sistem lalu diperinci secara bertahap ke tingkat detail yang lebih spesifik [6]:

1. Diagram Konteks

Diagram konteks merupakan diagram paling dasar dari suatu organisasi, yang menunjukkan bagaimana proses mengubah data yang masuk menjadi informasi yang dikeluarkan. Diagram konteks menampilkan satu proses utama yang merepresentasikan seluruh sistem, serta semua entitas eksternal dan aliran data utama yang menghubungkannya.



Gambar 2.7 Ilustrasi Diagram Level Konteks Sistem Pemrosesan Pesanan World's Trend

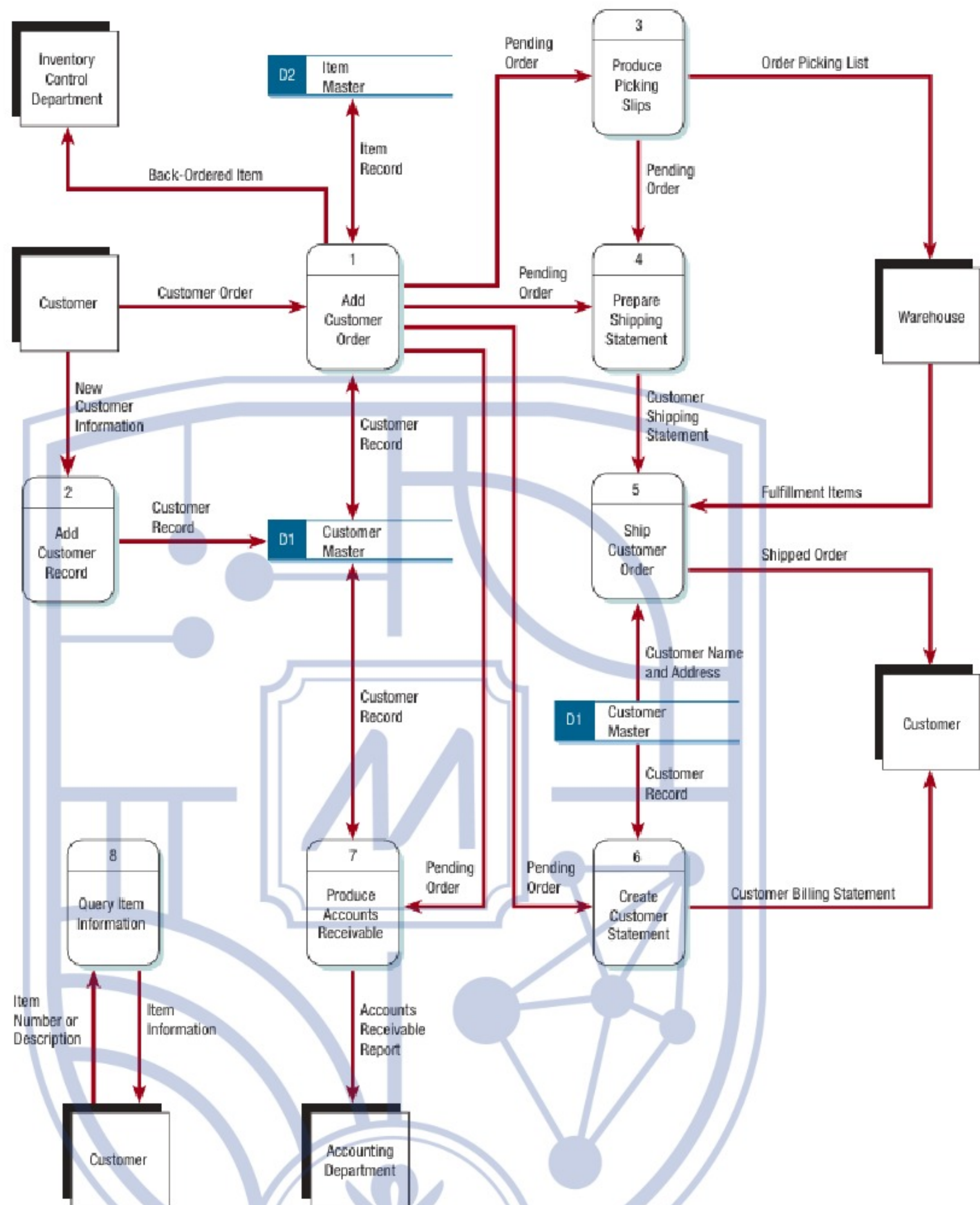
Gambar 2.7 menampilkan diagram konteks level untuk sistem pengolahan pesanan di perusahaan *World's Trend*. Diagram ini menjelaskan batas-batas sistem serta cara sistem berinteraksi dengan entitas di luar sistem. Setiap aliran data memiliki sumber dan tujuan yang terdefinisi dengan jelas, sehingga memastikan tidak ada data yang tersisa tanpa melalui proses pengolahan. Gambar ini menjelaskan cara sistem bekerja sebagai sarana komunikasi antar departemen agar menghindari pengulangan dan ketidaksesuaian data. Berikut adalah penjelasan dari diagram konteks diatas:

- a. Sistem Pemrosesan Pesanan (*Order Processing System*). Ini merupakan pusat dari seluruh kegiatan bisnis yang mengatur aliran informasi terkait proses pemesanan produk.
- b. Terdapat empat entitas yaitu, *Customer* (Pelanggan), *Warehouse* (Gudang), *Accounting Department* (Departemen Akuntansi), Departemen Pengendalian Inventaris (*Inventory Control Department*).
- c. *Data flow* pada diagram ini menjelaskan cara informasi masuk dan dikeluarkan dari sistem:
 - i. Pelanggan mengirimkan Pesanan Pelanggan, informasi Pelanggan Baru, dan Nomor/Deskripsi Produk. Pelanggan menerima Konfirmasi Pesanan Terkirim, Tagihan, dan Informasi Ketersediaan Produk.

- ii. Sistem mengirimkan daftar pengambilan produk kepada petugas gudang agar mereka dapat mempersiapkan produk yang diperlukan. Gudang mengirimkan informasi mengenai produk pesanan sebagai bukti bahwa produk tersebut sudah siap untuk diproses.
- iii. Sistem akan mengirimkan pemberitahuan mengenai *Back-Ordered Item* jika stok tidak mencukupi, sehingga bagian ini dapat melakukan pembelian kembali.
- iv. Sistem mengirimkan laporan Piutang (*Accounts Receivable Report*) untuk keperluan pencatatan keuangan serta proses penagihan selanjutnya.

2. Diagram 0 (*Level 0 Diagram*)

Diagram 0 merupakan tahap pecahan pertama dari proses tunggal yang terdapat dalam diagram konteks. Pada *level* ini, penggambaran diagram merinci proses utama yang ada dalam sistem dan menunjukkan hubungan aliran data antar proses, penyimpanan data, dan entitas. Tujuan dari diagram 0 adalah untuk menunjukkan semua proses utama dari sistem dan bagaimana mereka saling berhubungan satu sama lain untuk memproses data yang ada. Tujuan dipecahnya diagram konteks adalah agar menciptakan diagram yang seimbang dalam memastikan semua informasi yang disajikan dalam *DFD* lebih akurat dan tepat.



Gambar 2.8 Ilustrasi Diagram Level 0 Sistem Pemrosesan Pesanan World's Trend

Gambar 2.8 adalah Diagram 0, yang merupakan hasil pecahan dari diagram konteks pada tingkat sebelumnya. Diagram ini menjelaskan secara rinci mengenai proses-proses internal yang berlangsung di dalam sistem pemrosesan pesanan *World's Trend*. Berikut adalah penjelasan dari proses-proses utamanya:

- a. Pada awalnya, sistem memprioritaskan proses validasi identitas dan permintaan dari pelanggan:
 - i. Proses 2 (*Add Customer Record*): Memproses informasi pelanggan baru menjadi *record* pelanggan yang disimpan di D1 (*Customer Master*).

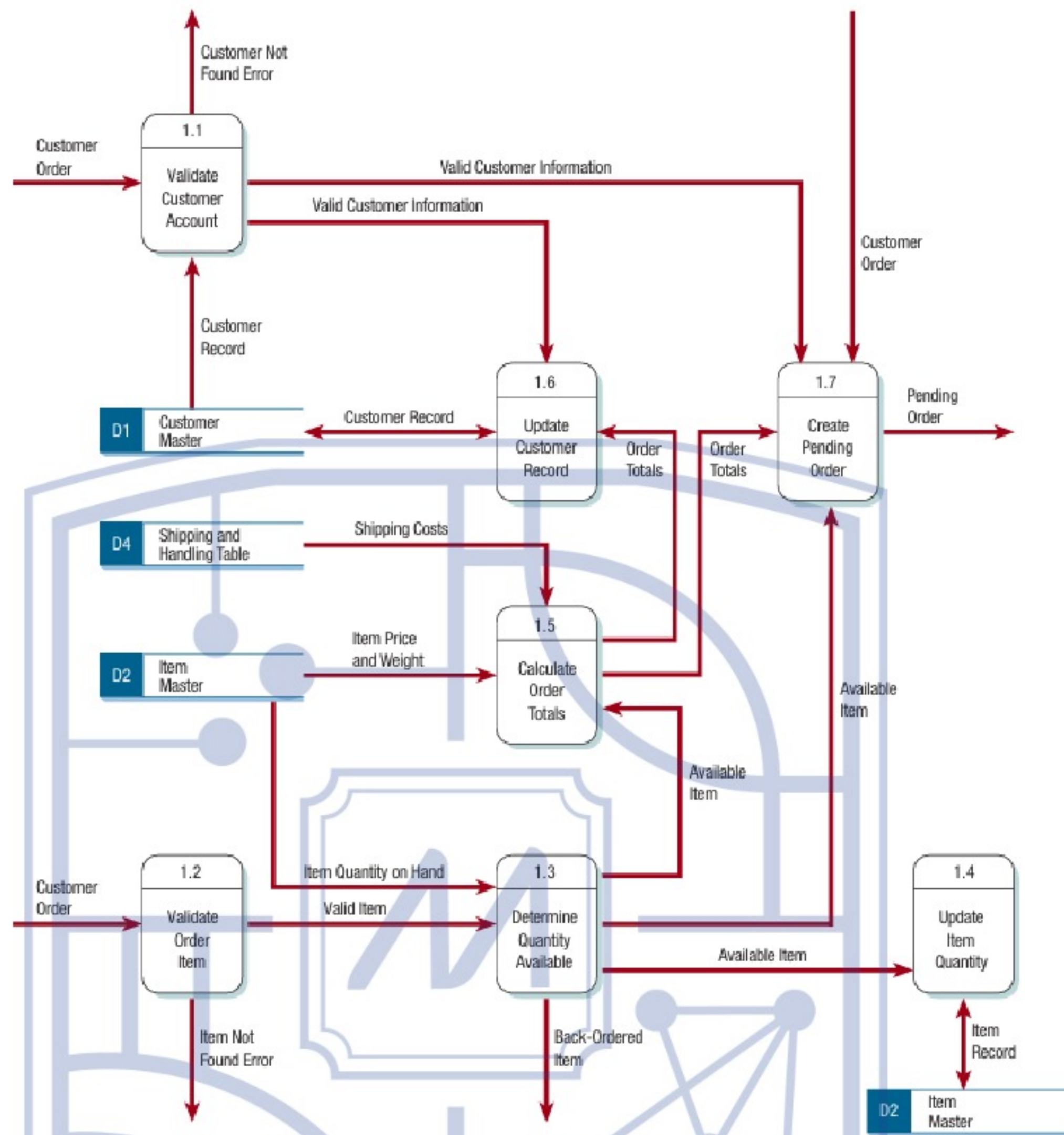
- ii. Proses 1 (*Add Customer Order*): Mengambil informasi dari D1 (*Customer Master*) dan D2 (*Item Master*) untuk memverifikasi kebenaran pesanan pelanggan. Jika stok tidak cukup, sistem akan mengirimkan informasi tentang produk yang belum tersedia ke departemen inventaris.
- iii. Proses 8 (*Query Item Information*): Memberikan layanan mandiri kepada pelanggan untuk mencari informasi detail produk melalui nomor item atau deskripsi produk.
- b. Setelah pesanan berhasil diverifikasi sebagai Pending Order, alur proses berpindah ke tahap operasional gudang:
 - i. Proses 3 (*Produce Picking Slips*) adalah membuat daftar pemilihan produk (*Order Picking List*) yang akan diberikan kepada gudang agar stok produk dapat diambil dari rak yang tersedia.
 - ii. Proses 4 (*Prepare Shipping Statement*): Mengumpulkan dan menyiapkan berbagai dokumen pengiriman sesuai dengan informasi yang terdapat dalam data pesanan.
 - iii. Proses 5 (*Ship Customer Order*): Setelah menerima konfirmasi mengenai produk dari gudang (*Fulfillment Items*), proses ini bertugas untuk mengirimkan produk fisik tersebut ke pelanggan beserta bukti pesanan yang telah dikirim.
- c. Tahap terakhir adalah memastikan bahwa semua transaksi dicatat secara finansial agar tidak terjadi pengulangan data.
 - i. Proses 6 (*Create Customer Statement*): Mengambil data dari dokumen D1 (*Master Pelanggan*) untuk mencetak laporan tagihan kepada pelanggan.
 - ii. Proses 7 (*Produce Accounts Receivable*) adalah menghasilkan laporan piutang yang kemudian dikirim ke bagian akuntansi untuk digunakan dalam pencatatan keuangan perusahaan.

Penggunaan D1 (*Customer Master*) oleh beberapa proses seperti proses 1, 2, 6, dan 7 menunjukkan bahwa data yang digunakan konsisten. Satu data pelanggan digunakan untuk berbagai tujuan seperti pemesanan, penagihan, serta pembuatan laporan piutang. Data *Pending Order* dari Proses 1 berfungsi sebagai pemicu bagi hampir semua proses lainnya, yaitu proses 3, 4, 6, dan 7. Ini menunjukkan sistem yang beroperasi berdasarkan peristiwa transaksi, yaitu sistem yang

diaktifkan oleh terjadinya suatu kejadian tertentu. Hubungan antara Proses 3 dan Proses 5 memastikan bahwa produk yang dikirim sesuai dengan produk yang telah diambil dari gudang.

3. Diagram Anak (*Child Diagrams*)

Diagram anak ini memberikan gambaran yang lebih rinci tentang proses-proses utama dalam diagram 0. Setiap proses yang ada pada diagram 0 akan diuraikan lebih lanjut menjadi diagram anaknya masing-masing. Misalnya apabila Diagram 0 memiliki 3 proses utama maka akan ada 3 diagram anak (diagram *level 1*) yang masing-masingnya menunjukkan cara kerja internal dari salah satu proses tersebut. Proses pada Diagram 0 disebut proses induk (*parent process*), sedangkan diagram yang dihasilkan disebut diagram anak (*child diagram*). Proses anak ini sepenuhnya mencakup semua fungsi dari proses induknya. Struktur hierarki *DFD* (Konteks $\rightarrow$ *Level 0* $>$ *Level 1* dan seterusnya) membantu menyederhanakan pemahaman akan sistem yang kompleks dengan memecah sistem besar menjadi diagram-diagram yang lebih kecil dan mudah dipahami. Semua aliran data input dan *output* dari proses “induk” pada *DFD Level 0* harus muncul di *DFD level 1* yang relevan. Namun, *DFD level 1* dapat memaparkan aliran data internal baru yang tidak terlihat di *level* sebelumnya. *DFD level 1* mengungkapkan detail yang lebih tepat, seperti sub-proses spesifik mana yang menangani data input atau data *output* tertentu. Entitas eksternal juga dapat ditampilkan pada *DFD level 1* untuk menyederhanakan pembacaan.



Gambar 2.9 Ilustrasi Diagram Level 1 Sistem Pemrosesan Pesanan World's Trend

Gambar 2.9 adalah *DFD* Level 1 yang menunjukkan pembagian atau pemecahan dari Proses 1, yang terdiri dari 7 subproses (1.1 sampai dengan 1.7). Proses ini adalah jalur utama untuk memasukkan data transaksi ke dalam sistem.

- a. Sistem melakukan pemeriksaan dua kali terhadap data yang diberikan oleh pelanggan untuk memastikan keaslian dan kebenaran data tersebut.
 - i. Proses 1.1 *Validate Customer Account*: Memeriksa apakah pelanggan telah terdaftar dengan membandingkan data ke dokumen D1 (*Customer Master*). Jika pelanggan tidak ditemukan, sistem akan menampilkan pesan kesalahan "*Customer Not Found Error*".
 - ii. Proses 1.2 *Validate Order Item*: Memastikan bahwa item yang dipesan terdaftar dalam katalog perusahaan dengan mengacu pada D2 (*Item Master*). Jika kode produk yang dimasukkan salah, akan muncul kesalahan dengan pesan *Item Not Found Error*.

- b. Setelah diverifikasi secara administratif, sistem kemudian memeriksa ketersediaan fisik produk.
- i. Proses 1.3 *Determine Quantity Available*: Membandingkan jumlah pesanan dengan jumlah item yang tersedia dari D2. Jika stok tersedia, data tersebut akan dikirim sebagai *Available Item*. Jika stok tidak mencukupi, sistem akan menciptakan item yang tercatat sebagai *Back-Ordered Item*, kemudian dikirim ke Departemen Inventaris.
 - ii. Proses 1.4 *Update Item Quantity*: Subproses ini secara otomatis memperbarui jumlah stok di D2 (*Item Master*) setelah produk dialokasikan untuk pesanan pelanggan. Ini penting untuk menghindari masalah sinkronisasi data yang tidak baik.
- c. Tahap perhitungan biaya proses 1.5 *Calculate Order Totals*: Di sini dilakukan integrasi data finansial. Proses ini menggunakan harga dan berat item dari D2 serta biaya pengiriman dari D4 (*Shipping and Handling Table*). Hasil akhirnya adalah *Order Totals*.
- d. Tahap Finalisasi Pesanan
- i. Proses 1.6 *Update Customer Record*: Total belanja pelanggan kembali dicatat dalam D1 (*Customer Master*) untuk memperbarui riwayat transaksi atau saldo piutang yang dimiliki oleh pelanggan.
 - ii. Proses 1.7 *Create Pending Order*: Ini merupakan tahap terakhir di mana seluruh informasi seperti data pelanggan, produk, serta total biaya digabungkan menjadi satu dokumen digital berupa *Pending Order*, yang selanjutnya akan memicu proses di bagian gudang (Proses 3 pada *DFD Level 0*).

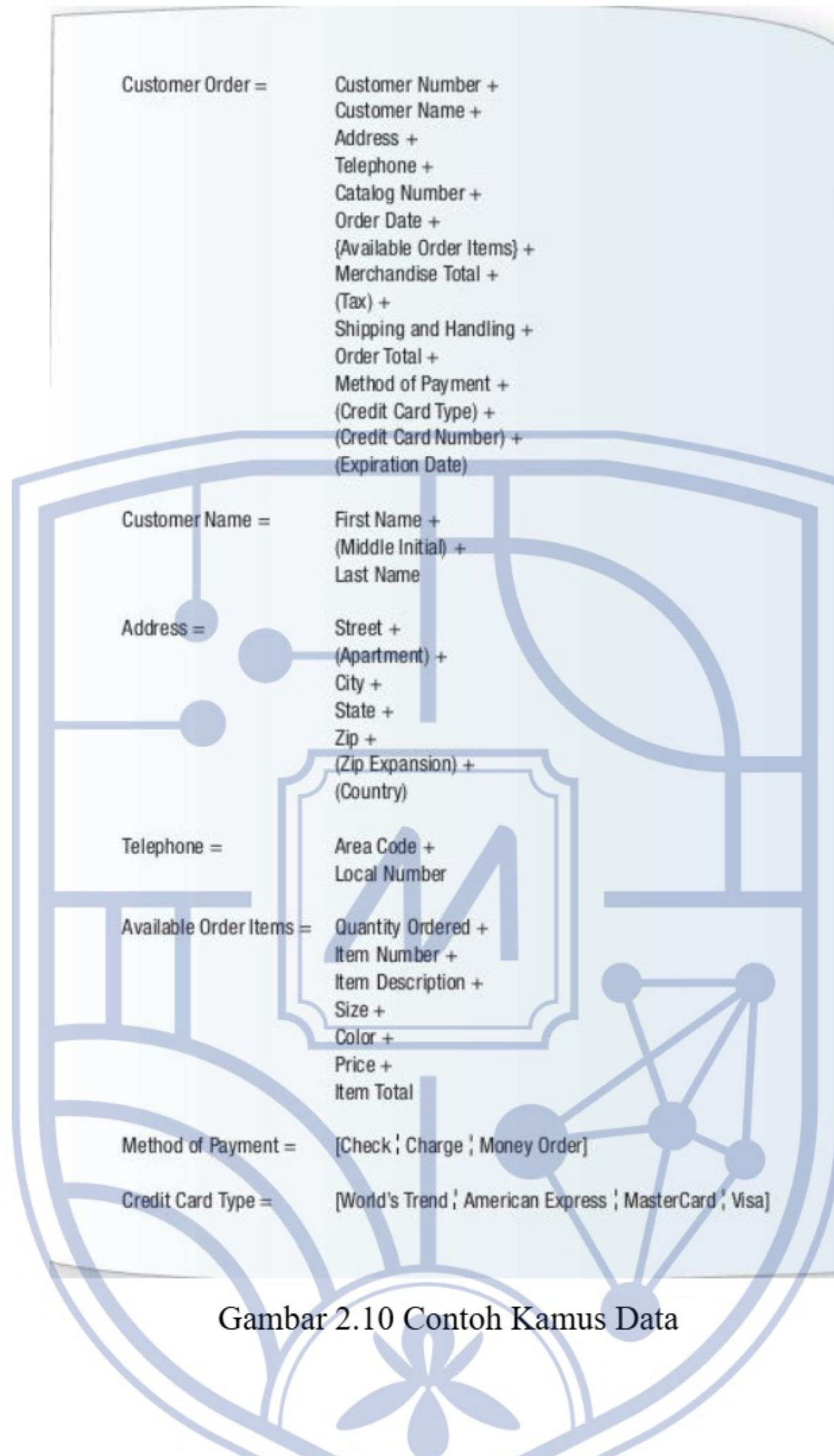
Adanya D4 (*Shipping and Handling Table*) menunjukkan bahwa sistem memiliki parameter biaya pengiriman yang tetap, sehingga membantu mengurangi kesalahan manual dalam perhitungan ongkos kirim. Pemisahan antara Proses 1.3 dan 1.4 memastikan sistem tidak hanya melakukan pemeriksaan tetapi juga mengunci stok (*Update Item Quantity*) secara langsung setelah pesanan diverifikasi, sehingga menghindari terjadinya *overselling*. D2 (*Item Master*) sering diakses untuk melakukan validasi, mengecek stok, mengambil harga, serta memperbarui jumlah produk. Ini menunjukkan peran *database* sebagai sumber kebenaran tunggal yang mencegah terjadinya redudansi data.

2.6 Kamus Data

Kamus data (*Data Dictionary*) adalah dokumen terorganisasi yang menyimpan penjelasan dan definisi dari semua elemen data, aliran data, *datastore*, serta proses yang ada dalam suatu sistem. Kamus data berfungsi sebagai alat utama untuk mengelola informasi terkait sistem secara menyeluruh. Tujuan dari kamus data ini adalah untuk memastikan setiap elemen data hanya didefinisikan sekali, merinci detail aliran data yang terdapat dalam *DFD*, menghindari kesulitan dalam melakukan modifikasi sistem karena semua informasi tersedia secara terpusat [6]. Struktur data menjelaskan susunan dari sebuah aliran data, serta menggambarkan bagaimana elemen-elemen data kecil digabungkan menjadi suatu kesatuan yang utuh. Notasi khusus digunakan untuk menggambarkan cara penyusunan tersebut [6].

Tabel 2.1 Notasi Kamus Data

Notasi	Arti	Deskripsi
=	Adalah komposisi dari	Menunjukkan elemen di sebelah kiri terdiri dari elemen di sebelah kanan
+	dan	Menunjukkan urutan atau gabungan antar elemen data
{ }	Iterasi/pengulangan	Menunjukkan elemen yang berulang (tabel atau grup berulang).
[]	Pilihan	Menunjukkan situasi yang hanya bisa terjadi salah satunya dari beberapa pilihan yang boleh ada
()	Opsional	Menunjukkan elemen yang bersifat opsional (boleh kosong)



Gambar 2.10 Contoh Kamus Data

Kamus data mendokumentasikan dua komponen dari *DFD*, yaitu:

1. Elemen Data (*Data Elements*)

Setiap aliran data yang ada pada *DFD* harus dijelaskan isinya. Elemen Data adalah unit terkecil yang tidak dapat dipecah lagi. Setiap elemen harus didokumentasikan secara detail. Dokumentasi dari setiap elemen mencakup [6]:

- a. Setiap elemen data dalam sistem harus didefinisikan sekali dalam kamus data dan dapat diisi melalui Formulir deskripsi elemen. Hal ini menjamin bahwa

setiap data memiliki identitas yang berbeda, jelas, serta sesuai dengan istilah yang biasa digunakan oleh pengguna utama.

- b. Penulisan struktur data menggunakan simbol-simbol khusus guna menggambarkan susunan atau organisasi datanya. Simbol sama dengan (=) berarti "berisi", tanda tambah (+) berarti "dan", kurung kurawal {} menunjukkan pengulangan atau tabel, kurung siku [] menunjukkan pilihan yang bisa dipilih (either/or), dan tanda kurung () menandakan elemen yang bisa dihilangkan atau tidak wajib.
- c. Elemen data dibagi menjadi dua kategori berdasarkan sumbernya, yaitu *element base* dan *element derived*. *Element base* adalah data yang dimasukkan langsung ke dalam sistem dan wajib disimpan dalam file, seperti nama atau alamat. Sementara itu, *element derived* adalah hasil dari proses perhitungan atau logika keputusan, misalnya total harga.
- d. Panjang elemen ditentukan sesuai dengan standar yang berlaku atau berdasarkan kebutuhan ruang maksimum agar data tidak terpotong. Tipe data bisa berupa angka, tanggal, huruf, varchar, atau karakter. Untuk data berupa angka, Format yang digunakan dapat berbentuk bilangan bulat, bilangan desimal, hingga Format khusus yang digunakan pada komputer besar seperti packed decimal.
- e. Untuk memastikan keakuratan, setiap elemen data dilengkapi dengan kriteria validasi yang dapat berupa rentang nilai untuk jenis data kontinu atau tabel kode untuk jenis data diskrit. Selain itu, elemen dapat memiliki nilai *default* yang muncul secara otomatis di layar entri, sehingga mempercepat proses pengetikan oleh operator.
- f. Jika suatu elemen memiliki nama yang berbeda dalam sistem lain (misalnya Customer Number disebut Client Number), maka nama tersebut dicatat sebagai alias. Area komentar tambahan juga tersedia untuk mencatat instruksi khusus, seperti Format tanggal tertentu atau metode digit pengecek yang digunakan.

2. Datastore

Semua elemen dasar harus disimpan dalam sistem, sedangkan elemen turunan, seperti total gaji bruto karyawan tahunan, dapat disimpan jika memang diperlukan. *datastore* dibuat untuk setiap entitas data yang berbeda dan akan disimpan; hal ini terjadi ketika elemen-elemen dasar dari aliran data dikelompokkan bersama untuk

membentuk sebuah catatan struktural, sehingga simpanan data yang unik dibuat untuk setiap catatan tersebut. Karena suatu aliran data tertentu mungkin hanya mencakup sebagian dari seluruh koleksi data yang terdapat dalam sebuah catatan struktural, analis sering kali perlu memeriksa beberapa struktur aliran data yang berbeda agar dapat membuat deskripsi tentang simpanan data yang komprehensif dan lengkap. Karakteristik yang harus dicantumkan dalam Formulir deskripsi simpanan data adalah sebagai berikut:

- a. *ID Datastore*: Merupakan entri wajib yang bertujuan mencegah analis menyimpan informasi yang berulang atau redundan, seperti D1 yang digunakan untuk *Customer Master*.
- b. Nama *Datastore*: Harus bersifat unik dan deskriptif.
- c. Alias adalah nama lain untuk tabel tersebut, seperti *Client Master* dapat digunakan sebagai alias untuk *Customer Master*.
- d. Deskripsi Singkat: Penjelasan singkat tentang isi atau manfaat dari *datastore* tersebut.
- e. Tipe File: Menentukan apakah file tersebut dikelola dengan cara komputerisasi atau dilakukan secara manual.
- f. Format: Menunjukkan apakah file tersebut berupa tabel *database* atau file datar yang sederhana.
- g. Statistik Pencatatan Data: Meliputi jumlah maksimum data yang dicatat serta rata-ratanya, serta perkiraan pertumbuhan setiap tahun untuk memperkirakan kebutuhan ruang penyimpanan cakram (*disk space*) dan perencanaan perangkat keras.
- h. Nama file atau nama dataset: Nama fisik file yang dapat dikosongkan pada tahap perancangan awal.
- i. Nama Struktur Data: Nama yang harus dicari dalam kamus data dan terkait dengan elemen-elemen yang terdapat dalam penyimpanan data ini.
- j. *Primary key* dan *secondary key*: Elemen yang digunakan untuk mengidentifikasi secara unik (seperti *Customer Number*) serta elemen yang digunakan untuk mengontrol urutan laporan dan melakukan pencarian data secara langsung.
- k. Komentar: Digunakan untuk menyampaikan Informasi tambahan yang tidak termasuk dalam kategori lain, seperti jadwal pencadangan, waktu pembaruan, aspek keamanan, atau pertimbangan lainnya yang relevan.

2.7 Use Case Diagram

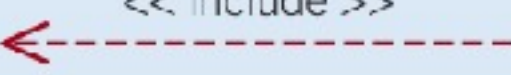
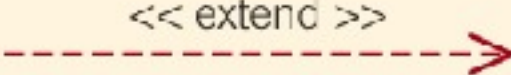
Use case diagram menunjukkan serangkaian kegiatan yang dilakukan untuk menghasilkan sebuah *output*. *Use case* menggambarkan bagaimana sebuah tindakan dilakukan oleh sebuah sistem atau penggunanya. Setiap *Use case* harus memiliki nama yang sederhana namun mencakup keseluruhan dari apa yang digambarkan [10].

Setiap *use case* memiliki hal penting dan berperan di dalamnya. Aktor merupakan seseorang atau organisasi yang melakukan dan menjalankan sebuah tindakan atau *event* pada suatu sistem. *Event* atau *Trigger* merupakan penyebab suatu *use case* berjalan. *Trigger* bisa berasal dari aktor eksternal seperti seorang pelanggan yang memesan produk atau jatuh tempo pada suatu benda yang memicu suatu proses berjalan pada sistem [10].

Penggunaan *use case* diagram digunakan agar analis mampu memahami lebih baik tentang situasi yang ada dan melihat bagaimana suatu sistem bekerja dan berinteraksi. *Use case* menjelaskan interaksi pengguna dengan sistem, namun tidak mencakup banyak detail yang diperlukan sebelum sistem dapat dikembangkan. *Use case* hanya menyampaikan sudut pandang pengguna [10].

Komponen utama dalam *use case* [6]:

1. **Aktor** : Merupakan orang atau sistem lain yang berinteraksi dengan sistem yang sedang dirancang. Aktor merepresentasikan peran yang dimainkan pengguna, bukan posisi atau jabatan tertentu. Aktor terdiri dari Aktor utama (memberikan data atau menerima informasi dari sistem. Aktor utama memiliki kepentingan langsung terhadap sistem tersebut) dan Aktor pendukung (berfungsi untuk membantu menjaga jalannya sistem atau memberikan layanan tambahan seperti analis atau pengembang kode).
2. **Use case** : Menunjukkan perilaku atau fungsi yang diberikan oleh sistem dan bermanfaat bagi aktor. *Use case* digambarkan dalam bentuk gelembung atau oval. *Use case* digunakan untuk merepresentasikan urutan transaksi atau perilaku sistem yang memberikan manfaat atau nilai bagi aktor yang terlibat.
3. **Boundary** : Sebuah kotak yang mengelilingi *Use case*, menandai batasan atau cakupan sistem yang sedang dikembangkan.

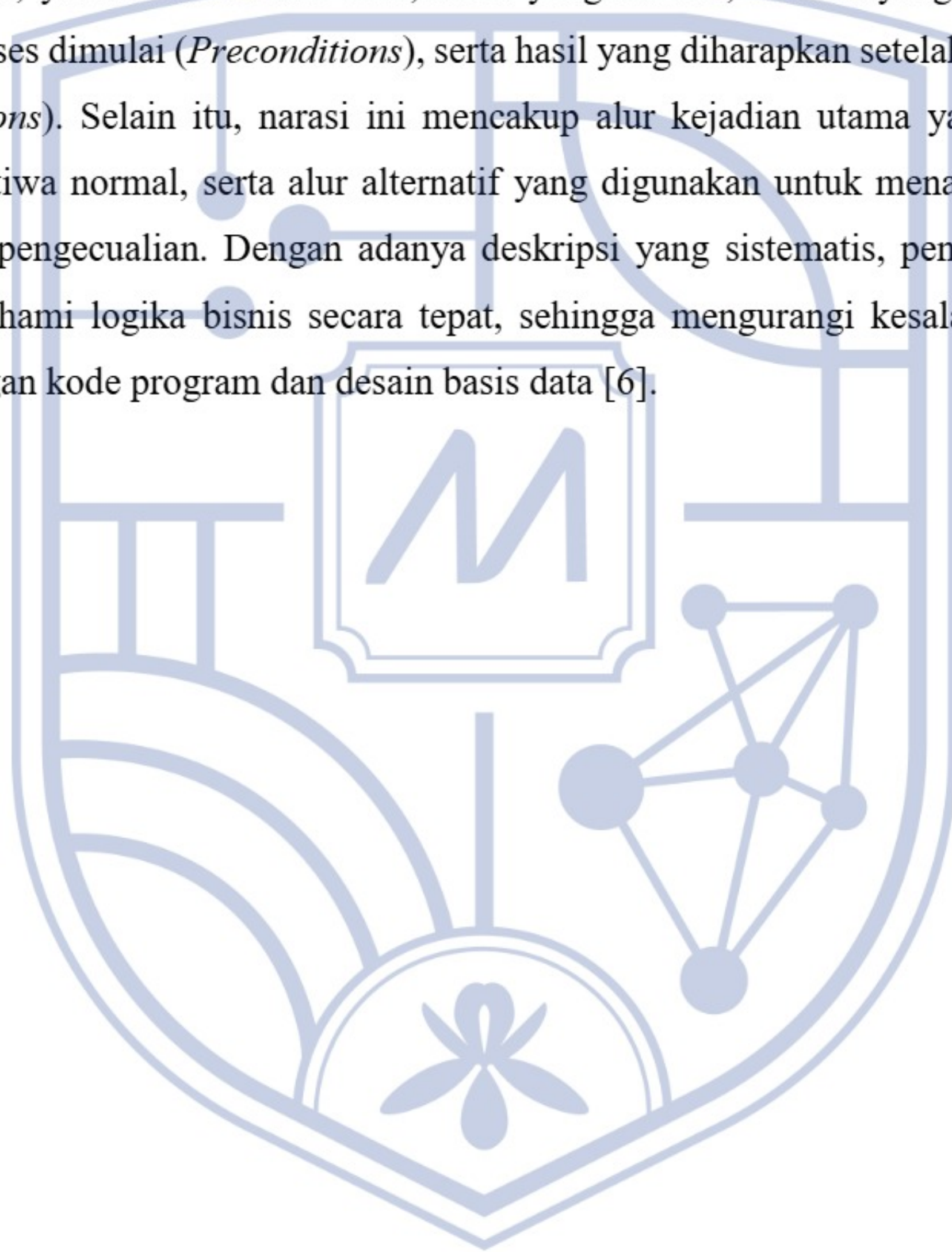
Relationship	Symbol	Meaning
Communicates		An actor is connected to a use case using a line with no arrowheads.
Includes		A use case contains a behavior that is common to more than one other use case. The arrow points to the common use case.
Extends		A different use case handles exceptions from the basic use case. The arrow points from the extended to the basic use case.
Generalizes		One UML "thing" is more general than another "thing." The arrow points to the general "thing."

Gambar 2.11 Simbol Use case

Empat jenis hubungan yang digunakan untuk menggambarkan interaksi antara aktor dan antar *use case* dalam suatu model adalah sebagai berikut [6]:

1. *Communicates* (Asosiasi) berbentuk garis lurus yang berfungsi untuk menunjukkan adanya interaksi antara aktor dan *use case*, seperti memulai, menerima informasi, atau berpartisipasi dalam suatu fungsi. Contohnya adalah mahasiswa yang berinteraksi dengan *use case* mendaftar mata kuliah.
2. *Includes* bersifat wajib yang digunakan untuk memecah proses yang kompleks menjadi bagian-bagian kecil yang selalu dilakukan, atau untuk menunjukkan fungsi yang sama yang digunakan oleh beberapa *use case*. Digambarkan dengan bentuk arah panah putus-putus yang mengarah dari *use case* yang umum. Contohnya adalah *use case* membayar iuran mahasiswa dalam mendaftar mata kuliah dan mengatur akomodasi.
3. *Extend* bersifat opsional atau kondisional yang dapat menambahkan fungsionalitas ke *use case* utama, tetapi hanya jika kondisi tertentu terpenuhi. Hubungan ini digunakan untuk menggambarkan alur alternatif, atau fitur tambahan dalam suatu proses. Hubungan ini digambarkan dengan bentuk arah panah yang mengarah dari *use case* yang diperluas ke *use case* dasar. contohnya adalah *use case* asuransi kesehatan mahasiswa *extends use case* dasar membayar biaya mahasiswa.
4. *Generalizes* berbentuk garis lurus dengan segitiga kosong (arah panah mengarah ke elemen yang lebih umum) ini bersifat hierarki ("adalah jenis dari") yang menunjukkan bahwa elemen yang lebih khusus/spesifik (*Child*) mewarisi semua sifat dan interaksi dari elemen yang lebih umum (*Parent*) yang digunakan untuk menggambarkan peran-peran yang berhubungan atau variasi fungsi yang serupa. Contoh yang dapat dilihat dari gambar adalah aktor mahasiswa paruh waktu *generalizes* ke mahasiswa. Ini berarti mahasiswa paruh waktu adalah jenis dari mahasiswa.

Written use case atau *use case scenario* adalah narasi teks yang digunakan untuk mendokumentasikan setiap fungsionalitas yang diwakili dalam diagram *use case*. Hal ini bertujuan untuk menjelaskan secara mendalam mengenai interaksi antara aktor dan sistem, yang tidak bisa dijelaskan hanya dengan menggunakan simbol visual. Deskripsi ini sangat penting pada tahap analisis karena memperjelas aturan bisnis, langkah-langkah prosedural, serta kondisi-kondisi yang harus terpenuhi agar suatu fungsi dapat berjalan dengan kata-kata yang Formal. Dalam penerapan sebenarnya, deskripsi ini membagi proses menjadi beberapa bagian utama, yaitu identitas *use case*, aktor yang terlibat, kondisi yang harus terpenuhi sebelum proses dimulai (*Preconditions*), serta hasil yang diharapkan setelah proses berakhir (*postconditions*). Selain itu, narasi ini mencakup alur kejadian utama yang menjelaskan urutan peristiwa normal, serta alur alternatif yang digunakan untuk menangani kesalahan atau situasi pengecualian. Dengan adanya deskripsi yang sistematis, pengembang sistem dapat memahami logika bisnis secara tepat, sehingga mengurangi kesalahan pada tahap pengembangan kode program dan desain basis data [6].

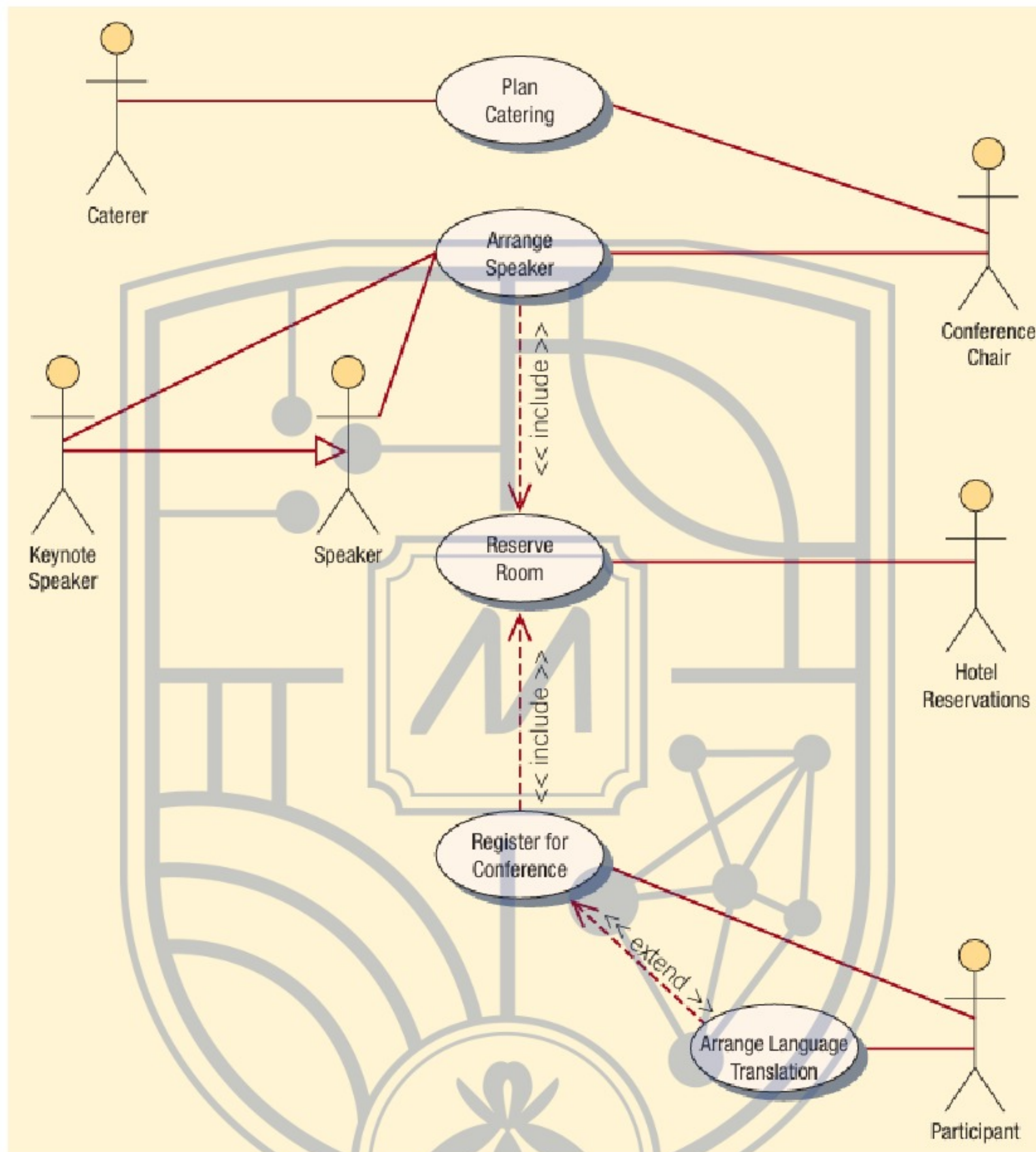


Use case name:	Register for Conference	UniqueID: Conf RG 003
Area:	Conference Planning	
Actor(s):	Participant	
Stakeholder:	Conference Sponsor, Conference Speakers	
Level:	Blue	
Description:	Allow conference participant to register online for the conference using a secure website.	
Triggering Event:	Participant uses Conference Registration website, enters userID and password, and clicks the logon button.	
Trigger type:	☒ External ☐ Temporal	
Steps Performed (Main Path)	Information for Steps	
1. Participant logs in using the secure web server.	userID, Password	
2. Participant record is read and password is verified.	Participant Record, userID, Password	
3. Participant and session information is displayed on the Registration web page.	Participant Record, Session Record	
4. Participant enters information on the Registration Web form and clicks Submit button.	Registration Web Form	
5. Registration information is validated on the Web server.	Registration Web Form	
6. Registration Confirmation page is displayed to confirm registration information.	Confirmation Web Page	
7. Credit card is charged for registration fees.	Secure Credit Card Web Page	
8. Add Registration Journal record is written.	Confirmation Web Page	
9. Registration record is updated on the Registration Master.	Confirmation Web Page, Registration Record	
10. Session record is updated for each selected session on the Session Master.	Confirmation Web Page, Session Record	
11. Participant record is updated for the participant on the Participant Master.	Confirmation Web Page, Participant Record	
12. Successful Registration Confirmation web page is sent to the participant.	Registration Record Confirmation Number	
Preconditions:	Participant has already registered and has created a user account.	
Postconditions:	Participant has successfully registered for the conference.	
Assumptions:	Participant has a browser and a valid userID and password.	
Success Guarantee:	Participant has registered for the conference and is enrolled in all selected sessions.	
Minimum Guarantee:	Participant was able to logon.	
Requirements Met:	Allow conference participants to be able to register for the conference using a secure website.	
Outstanding Issues:	How should a rejected credit card be handled?	
Priority:	High	
Risk:	Medium	

Gambar 2.12 Use case scenario

Skenario *use case* pada gambar 2.12 ini menjelaskan proses interaksi antara aktor *Participant* dengan sistem untuk melakukan pendaftaran konferensi. Proses dimulai dengan aktor melakukan *login* menggunakan *User ID* dan kata sandi yang benar sebagai syarat awal agar sistem dapat memverifikasi identitas pengguna sebelum lanjut ke tahap pengisian data. Setelah proses autentikasi selesai, peserta selanjutnya diminta untuk mengisi informasi pendaftaran dan memilih sesi konferensi yang tersedia. Pada tahap ini, sistem juga melakukan fungsi pemeriksaan pembayaran dengan mengirim tagihan ke kartu kredit yang dimiliki oleh peserta. Setiap perubahan data yang terjadi selama proses pendaftaran akan langsung diperbarui secara otomatis oleh sistem ke dalam tabel *Registration Master*, *Session Master*, dan *Participant Master*. Interaksi ini ditutup dengan respons dari sistem berupa tampilan konfirmasi pendaftaran yang dilengkapi dengan nomor referensi unik untuk

peserta. Sebagai hasil akhir, data transaksi akan disimpan secara permanen dalam basis data dan status kepesertaan aktor dianggap sah dalam sistem konferensi tersebut [6].



Gambar 2.13 Ilustrasi Use case

Gambar diatas merupakan *use case* yang merepresentasikan sistem perencanaan konferensi. Penjelasan untuk setiap elemen dalam diagram tersebut mencakup [6]:

1. Aktor mewakili peran yang diperankan oleh pengguna saat berinteraksi dengan sistem. Dalam gambar tersebut, aktor-aktor yang terlibat meliputi:
 - a. *Conference Chair* yang bertanggung jawab dalam merencanakan serta mengurus penyelenggaraan konferensi.

- b. *Speaker* dan *Keynote Speaker*, yaitu individu yang menyampaikan materi dalam acara tersebut
 - c. *Participant*, yaitu individu yang mendaftar pada acara tersebut.
 - d. *Hotel Reservation*, yaitu pihak penyedia akomodasi.
 - e. *Caterer* dapat berupa karyawan hotel atau penyedia layanan eksternal yang bertugas untuk menyediakan makanan.
2. Beberapa interaksi yang dapat terlihat pada gambar diatas meliputi:
 - a. *Conference Chair* dan *Caterer* bekerja sama dalam mengembangkan scenario perencanaan makanan dan perjamuan.
 - b. *Conference Chair* memiliki tanggung jawab khusus dalam mengatur pembicara, sementara *participant* berinteraksi melalui proses pendaftaran.
3. Gambar diatas menggambarkan hubungan *include* dan *extend*. Hubungan *include* terjadi antara *use case reserve room* dengan *use case arrange speaker* dan *register for conference*. Hal ini berarti pemesanan kamar menjadi hal yang wajib dilakukan karena pembicara dan peserta pasti memerlukan tempat menginap selama acara berlangsung. Hubungan *extend* dapat terlihat pada *arrange language translation* yang merupakan *extend* dari *register for conference*. Hal ini berarti layanan penerjemahan bahasa bersifat opsional, yaitu hanya dilakukan jika ada *participant* yang membutuhkannya, dan tidak berlaku untuk semua *participant*.
4. Pada bagian aktor dalam gambar tersebut, aktor *Speaker* merupakan bentuk generalisasi dari *Keynote Speaker*. Hal ini berarti *Keynote speaker* memiliki semua karakteristik dasar seorang *speaker*, tetapi memiliki peran yang lebih spesifik dalam sistem tersebut.

2.8 Normalisasi Data

Normalisasi didefinisikan sebagai proses mengubah struktur data yang kompleks dan pandangan pengguna menjadi bentuk yang lebih sederhana, stabil, serta lebih mudah dikelola. Data yang diproses dalam normalisasi biasanya berupa relasi yang belum ternormalisasi, yang memiliki ciri khas adanya kelompok data berulang. Contohnya, dalam laporan penjualan sebuah perusahaan, informasi mengenai pelanggan dan transaksi bisa berulang di setiap baris laporan. Normalisasi data penting dilakukan dalam mengimplementasikan *database management system (DBMS)* [6].

Proses normalisasi dilakukan dalam tiga tahap utama untuk menyederhanakan struktur data secara keseluruhan. pada normalisasi data [6]:

1. *UNF (Unnormalized Form)*

Bentuk tidak ternormalisasi atau *Unnormalized Form (UNF)* merupakan tahap awal dalam struktur data sebelum proses normalisasi dilakukan. Sebuah relasi atau tabel dikatakan berada dalam bentuk *UNF* karena mengandung kelompok berulang, yang merupakan ciri khas dari bentuk ini. Kelompok berulang adalah kumpulan atribut yang bisa memiliki beberapa nilai untuk setiap catatan dalam relasi tersebut. Karena adanya kelompok berulang, satu atribut tunggal tidak mampu bertindak sebagai kunci utama (*primary key*) untuk seluruh relasi. Sebaliknya, atribut kunci tersebut memiliki hubungan satu-ke-banyak dengan atribut-atribut lain dalam kelompok berulang, sehingga menyebabkan adanya redundansi data dan berbagai anomali.

2. *1NF (First Normal Form)*

Tahap kedua adalah menghilangkan semua kelompok data yang berulang dan menentukan kunci utama (*primary key*), sehingga relasi awal dibagi menjadi beberapa relasi yang lebih kecil. Setiap atribut ada suatu entitas hanya boleh memiliki satu nilai. Kunci utama (*Primary Key*) ditentukan diidentifikasi pada tahap ini. kedua adalah memastikan bahwa semua atribut yang bukan kunci (*nonkey attributes*) sepenuhnya bergantung pada kunci utama, dengan cara menghilangkan ketergantungan sebagian (*partial dependencies*) dan memindahkannya ke relasi yang terpisah.

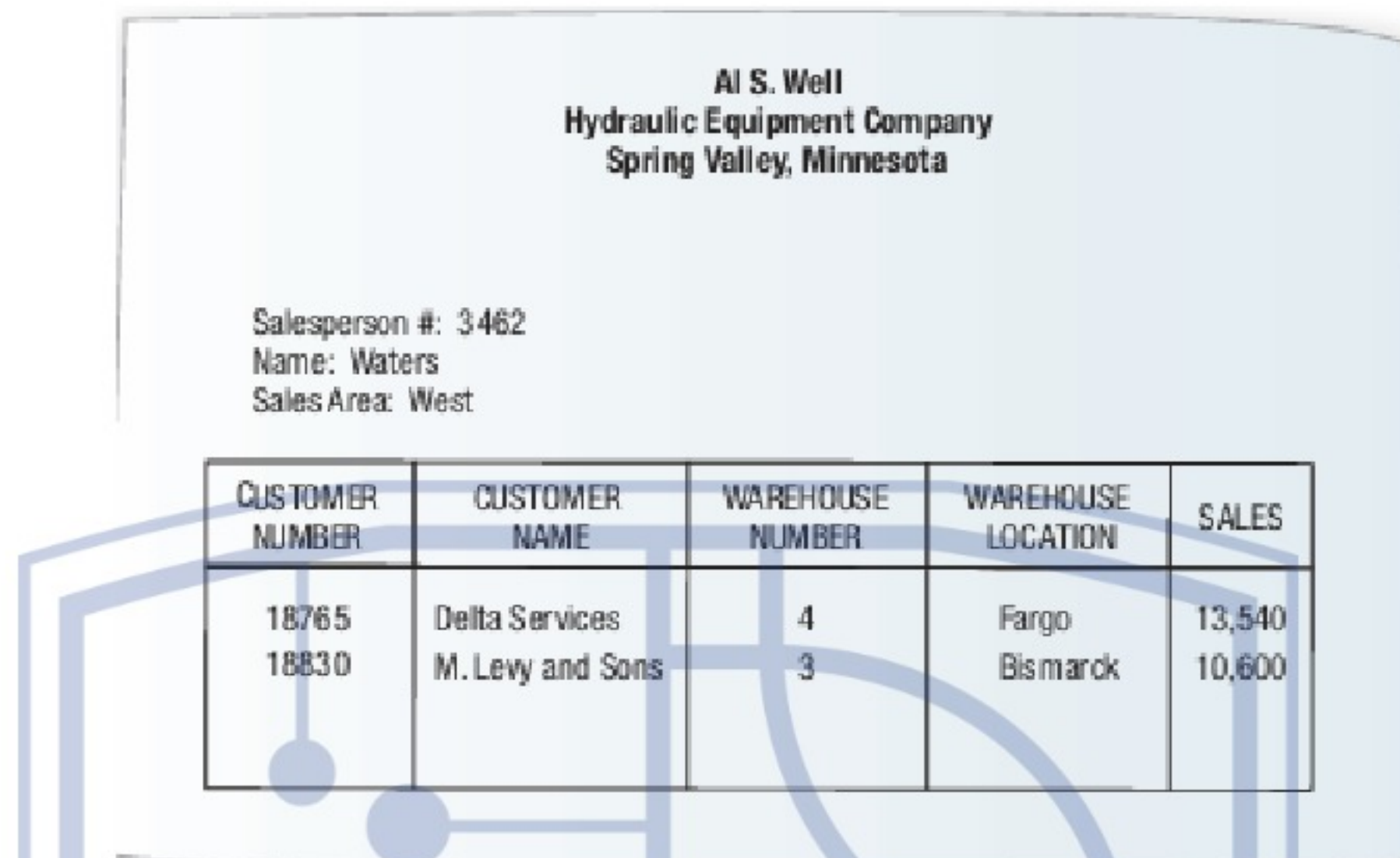
3. *2NF (Second Normal Form)*

Tahap ketiga adalah memastikan bahwa semua atribut yang bukan kunci (*nonkey attributes*) sepenuhnya bergantung pada kunci utama, dengan cara menghilangkan ketergantungan sebagian (*partial dependencies*) dan memindahkannya ke relasi yang terpisah.

4. *3NF (Third Normal Form)*

Tahap yang terakhir adalah menghilangkan ketergantungan transitif, yaitu situasi di mana atribut non-kunci bergantung pada atribut non-kunci lainnya. Tujuan dari tiga tahapan ini adalah mencapai bentuk data ideal yang dikenal sebagai *Third Normal*

Form (3NF), yang bertujuan untuk meminimalkan duplikasi dan ketidaksesuaian data.



CUSTOMER NUMBER	CUSTOMER NAME	WAREHOUSE NUMBER	WAREHOUSE LOCATION	SALES
18765	Delta Services	4	Fargo	13,540
18830	M. Levy and Sons	3	Bismarck	10,600

Gambar 2.14 Laporan Al S. Well Hydraulic Equipment Company

Gambar 2.14 menampilkan laporan operasional di *Al S. Well Hydraulic Equipment Company* yang merepresentasikan pandangan pengguna dalam bentuk tidak ternormalisasi (*Unnormalized Form*).



Gambar 2.15 Data Struktur yang didapatkan dari Kamus data

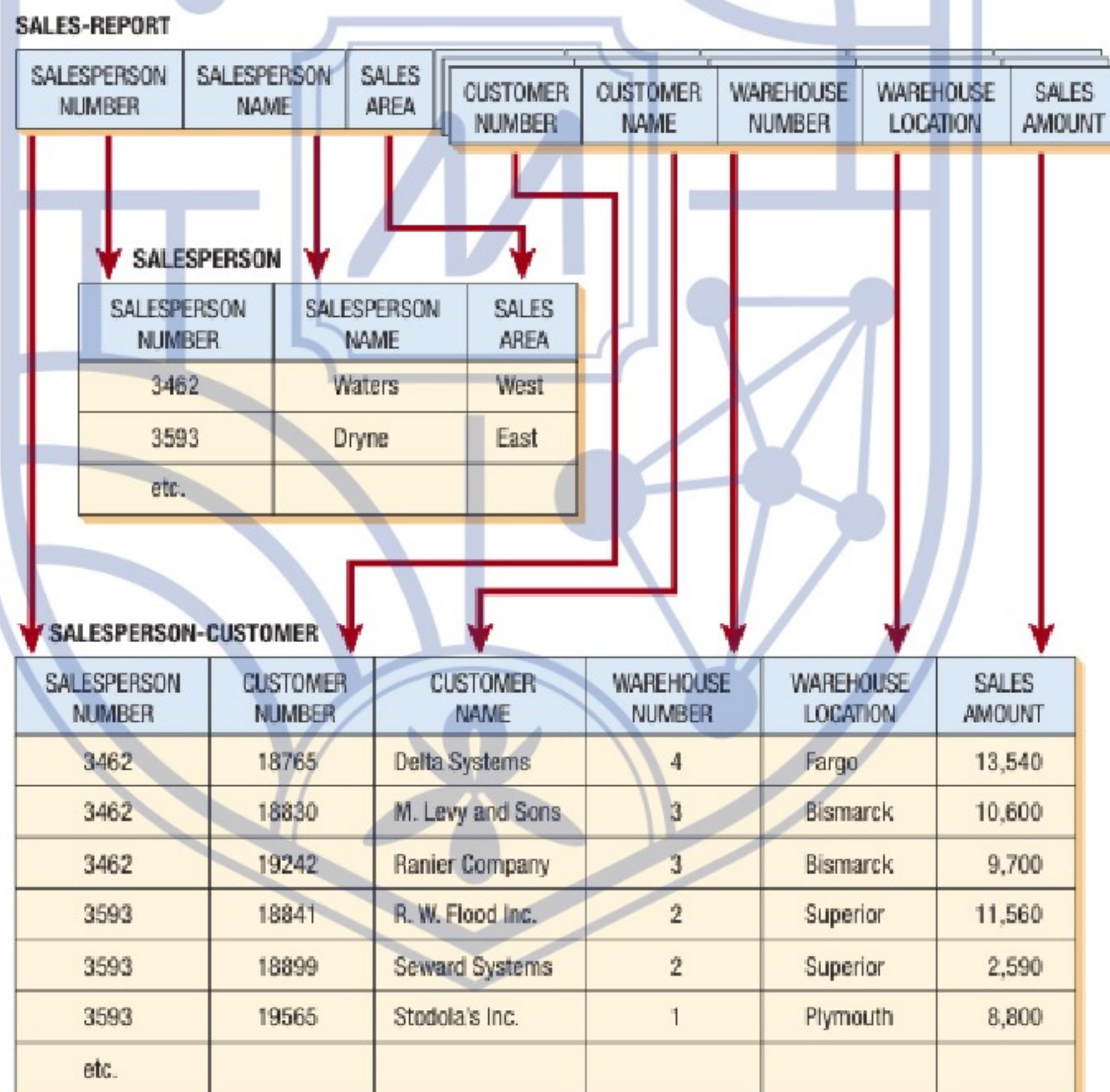
Struktur data pada gambar 2.15 menunjukkan adanya kelompok berulang atau grup yang berulang pada atribut pelanggan (*CUSTOMER-NUMBER*, *CUSTOMER-NAME*), data gudang, serta jumlah penjualan (*SALES-AMOUNT*), yang dinyatakan melalui indentasi dan tanda bintang (*). Dalam tinjauan basis data, laporan ini menjelaskan tingkat kekompleksan hubungan antar data, di mana satu tenaga penjual (*SALESPERSON-NUMBER*) memiliki keterhubungan satu-ke-banyak terhadap beberapa pelanggan. Kondisi ini menciptakan pengulangan data karena informasi Header seperti nama penjual dan area penjualan harus

ditulis ulang untuk setiap baris transaksi pelanggan. Oleh karena itu, normalisasi diperlukan untuk memecah struktur tersebut menjadi tabel-tabel relasional yang lebih sederhana agar dapat menghindari terjadinya anomali data serta memastikan bahwa setiap atribut yang bukan kunci memiliki ketergantungan fungsional yang tepat terhadap kunci utamanya. Laporan pada gambar 2.10 dapat dinyatakan dalam notasi berikut ini:

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

Gambar 2.16 Notasi Laporan

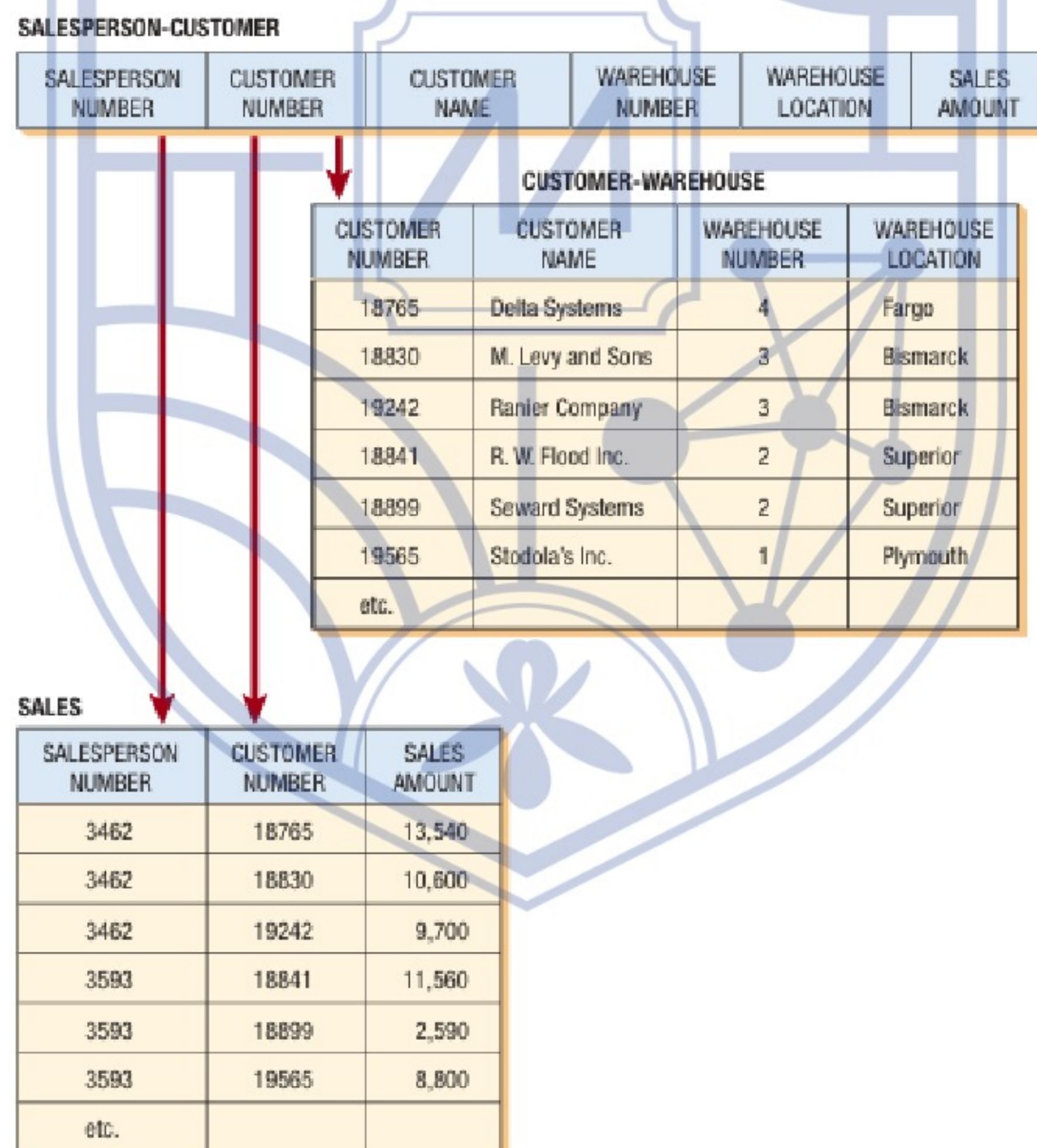
Gambar 2.16, himpunan kurung yang lebih dalam merepresentasikan kelompok yang berulang.



Gambar 2.17 Pemisahan Unnormalized relation ke Salesperson (3NF) dan Salesperson-customer (1NF)

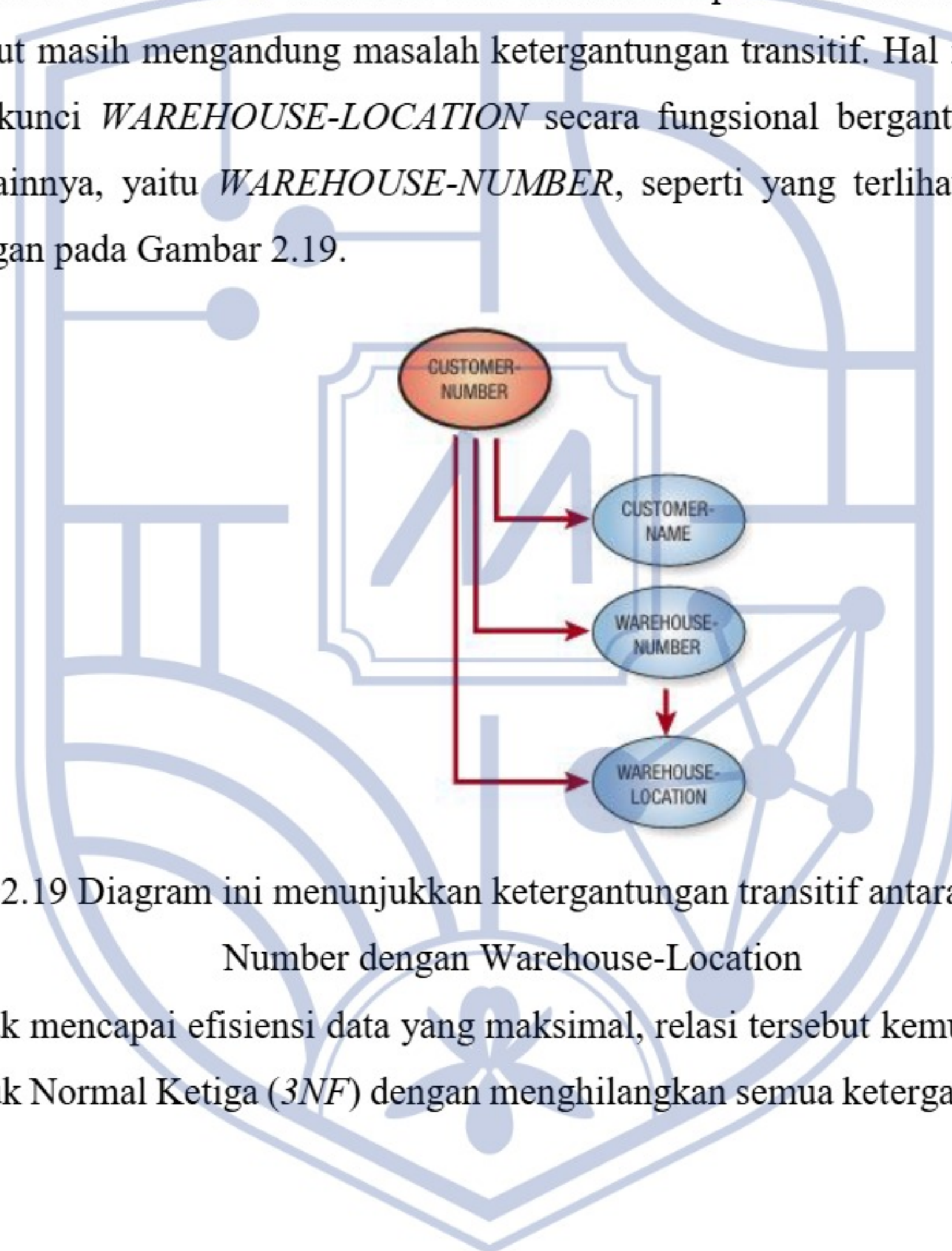
Proses normalisasi pada laporan operasional *Al S. Well Hydraulic Equipment Company* dimulai dengan mengubah bentuk yang tidak terstruktur (*unnormalized*) menjadi

bentuk normal pertama (1NF) dengan metode menghilangkan kelompok yang berulang (*repeating groups*). Berdasarkan Gambar 2.13, relasi awal *SALES-REPORT* dibagi menjadi dua relasi yang berbeda, yaitu *SALESPERSON* dan *SALESPERSON-CUSTOMER*. Relasi *SALESPERSON* menerima atribut statis yang tidak berulang dengan *SALESPERSON-NUMBER* sebagai kunci primer, sedangkan relasi *SALESPERSON-CUSTOMER* menggabungkan kunci primer tersebut dengan semua atribut yang sebelumnya merupakan bagian dari grup berulang. Meskipun relasi *SALESPERSON-CUSTOMER* sudah memenuhi kondisi 1NF, masih diperlukan penggunaan kunci komposit yang terdiri dari kolom *SALESPERSON-NUMBER* dan *CUSTOMER-NUMBER* agar data dapat diakses secara unik. Namun, hubungan ini masih dinilai tidak sempurna karena adanya anomali fungsional; atribut seperti *CUSTOMER-NAME*, *WAREHOUSE-NUMBER*, dan *WAREHOUSE-LOCATION* hanya bergantung fungsional pada *CUSTOMER-NUMBER*, bukan pada seluruh kunci komposit tersebut.



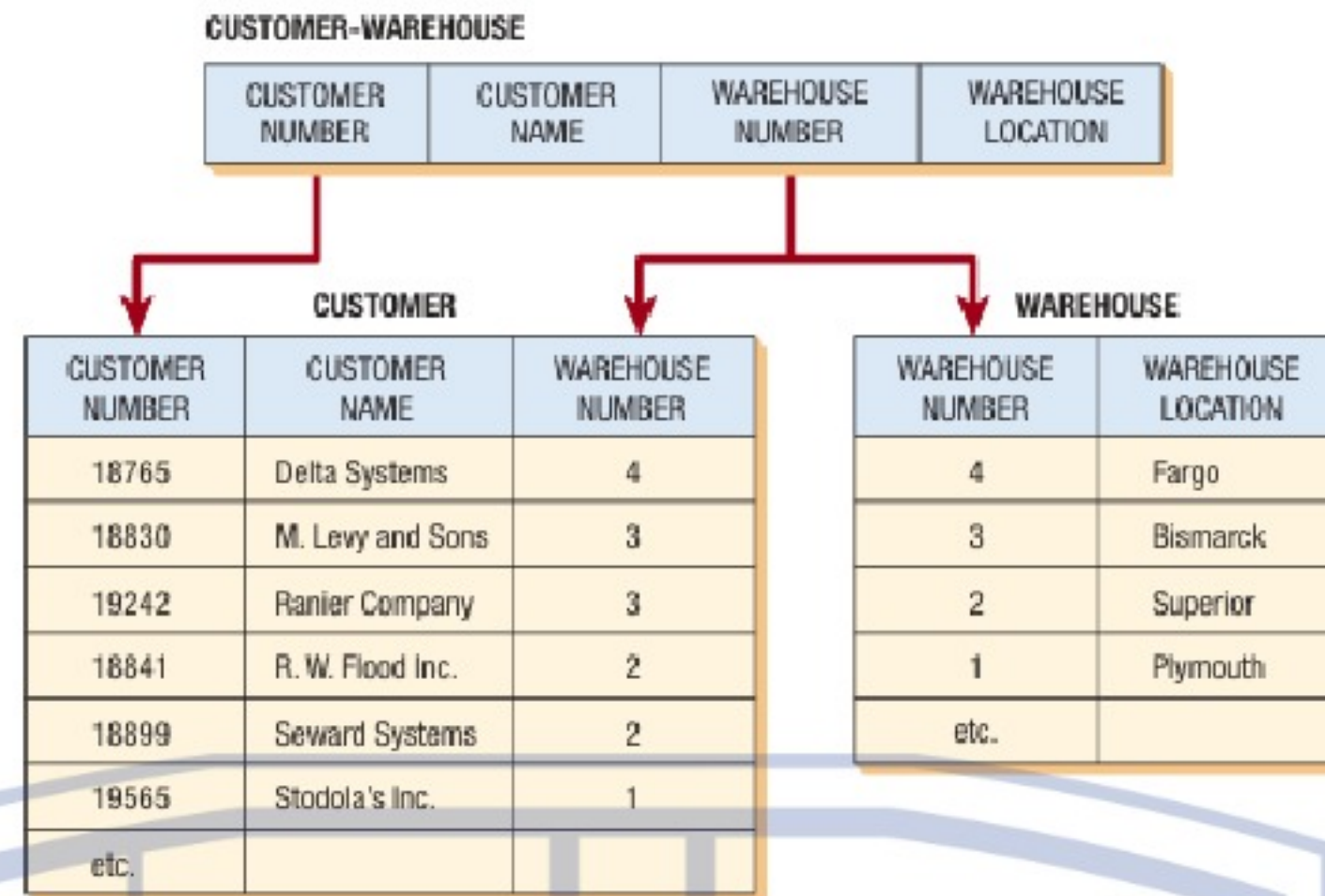
Gambar 2.18 Relasi dari Salesperson-Customer dipisah menjadi relasi bernama Customer-Warehouse (2NF) dan relasi bernama Sales (1NF)

Tahap berikutnya adalah proses transformasi ke Bentuk Normal Kedua (*2NF*), yang mengharuskan setiap atribut yang bukan kunci memiliki ketergantungan fungsional yang lengkap terhadap kunci primer. Seperti yang terlihat dalam Gambar 2.14, relasi *SALESPERSON-CUSTOMER* dibagi kembali menjadi dua relasi baru, yaitu *SALES* dan *CUSTOMER-WAREHOUSE*. Langkah ini bertujuan untuk menghilangkan ketergantungan sebagian, di mana atribut seperti nama pelanggan dan data gudang sebelumnya hanya bergantung pada nomor pelanggan, bukan pada kunci komposit secara keseluruhan. Meskipun relasi *CUSTOMER-WAREHOUSE* telah mencapai bentuk normal kedua (*2NF*), relasi tersebut masih mengandung masalah ketergantungan transitif. Hal ini terjadi karena atribut non-kunci *WAREHOUSE-LOCATION* secara fungsional bergantung pada atribut non-kunci lainnya, yaitu *WAREHOUSE-NUMBER*, seperti yang terlihat dalam diagram ketergantungan pada Gambar 2.19.



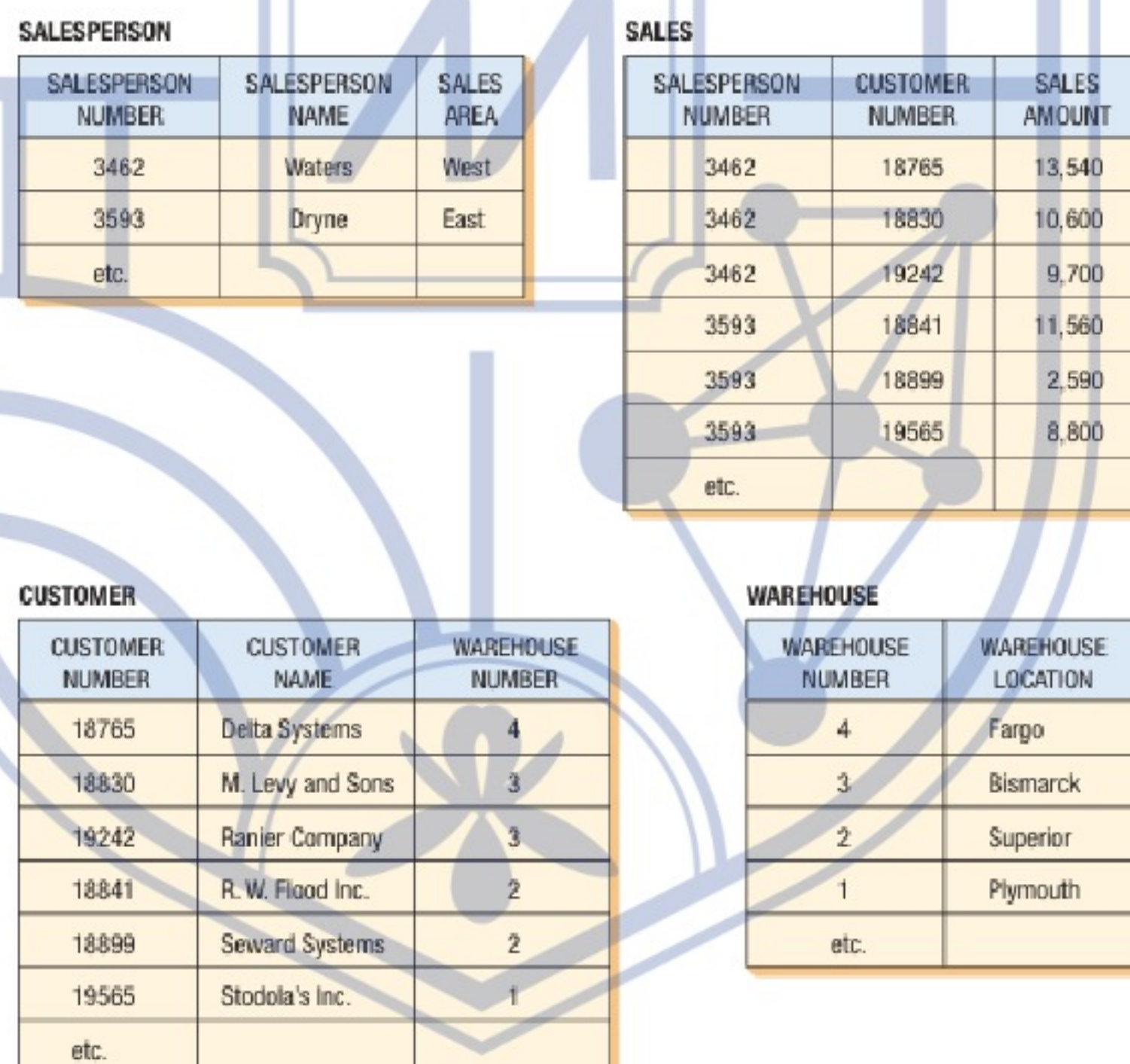
Gambar 2.19 Diagram ini menunjukkan ketergantungan transitif antara Warehouse-Number dengan Warehouse-Location

Untuk mencapai efisiensi data yang maksimal, relasi tersebut kemudian diproses ke dalam Bentuk Normal Ketiga (*3NF*) dengan menghilangkan semua ketergantungan transitif.



Gambar 2.20 Relasi Customer-Warehouse dipisah menjadi relasi Customer (1NF) dan Warehouse (1NF)

Berdasarkan Gambar 2.20, relasi antara *CUSTOMER* dan *WAREHOUSE* dibagi menjadi dua relasi yang terpisah, yaitu relasi *CUSTOMER* dan relasi *WAREHOUSE*.



Gambar 2.21 Struktur Akhir

Dalam struktur akhir ini (Gambar 2.21), laporan asli *SALES-REPORT* telah berhasil diuraikan menjadi empat relasi yang stabil, yaitu *SALESPERSON*, *SALES*, *CUSTOMER*, dan *WAREHOUSE*. Dalam relasi *CUSTOMER*, kolom *WAREHOUSE-NUMBER* berfungsi sebagai kunci asing yang menghubungkan informasi pelanggan dengan lokasi gudang

penyedia barang tersebut. TransFormasi ke bentuk normal ketiga (*3NF*) sangat penting dalam desain basis data karena dapat menghilangkan anomali yang terjadi pada proses penyisipan, penghapusan, dan perbaruan data, serta memastikan ketepatan dan konsistensi data melalui hubungan antar entitas yang lebih sederhana dan masuk akal.

2.9 Basis Data

Basis data (*database*) teknologi yang sangat berguna bagi organisasi dalam menyimpan, mengakses, dan mengambil data. Basis data dirancang untuk digunakan oleh banyak pengguna, dan cara pengguna melihat serta menggambarkan data tersebut disebut user view. Tujuan utamanya adalah untuk mengatur data untuk memperoleh kemudahan, kecepatan dan ketepatan dalam pengambilan informasi yang diinginkan [11].

Manfaat basis data [11]:

1. Kemudahan dan kecepatan, yaitu kemudahan dalam memilih data secara berurutan secara efisien dan cepat.
2. *Multi-user*, yaitu memberikan kemudahan akses bagi banyak pengguna secara bersamaan dalam satu waktu.
3. Keamanan data, yaitu melindungi data melalui sistem keamanan melalui kata sandi (*password*) sehingga akses hanya terbatas untuk pihak yang diizinkan.
4. Penghematan biaya perangkat, yaitu dengan mengurangi kebutuhan akan penyimpanan data lokal (*hard disk*) pada setiap komputer.
5. Kontrol data terpusat, yaitu memudahkan pengelolaan dan akses data dari berbagai Lokasi atau divisi melalui satu server pusat.

Jenis dan model basis data beserta fungsinya [11]:

1. *Operational Database* memiliki fungsi untuk mengelola data dinamis secara langsung dan *real-time* sehingga memungkinkan pengguna untuk melihat, memanggil dan memodifikasi data.
2. *Database Warehouse* merupakan repositori terpusat untuk data terintegrasi dari berbagai sumber yang digunakan untuk analisis dan pembuatan laporan historis.
3. *Distributed Database* adalah basis data yang perangkat penyimpanannya tersebar di beberapa komputer yang berbeda. Komunikasi sistemnya terdistribusi melalui suatu situs-situs yang tergabung.
4. *Relational Database* atau basis data relasional adalah basis data yang mengorganisir data berdasarkan model data relasi, di mana data diatur dalam bentuk tabel yang saling berelasi.

Sistem dalam basis data terdiri dari 4 komponen pokok [6]:

1. Basis data (*database*) memiliki ciri-ciri data dapat disimpan secara terintegrasi (data terkumpul dari berbagai macam file yang disusun dengan menghilangkan file yang berulang (*redudansi*) dan data dapat dipakai secara bersama-sama oleh *user* dalam waktu yang bersamaan (*shared*).
2. Perangkat keras (*hardware*), yaitu semua perangkat keras komputer yang digunakan untuk pengelolaan sistem basis data tersebut. Perangkat keras yang terdapat dalam sistem basis data, terdiri dari komputer, perangkat untuk menyimpan data permanen (*harddisk*), dan perangkat komunikasi untuk sistem jaringan.
3. Perangkat lunak (*software*) berfungsi sebagai perantara antara *user* dengan data fisik pada basis data yang dapat berupa *Database Management System (DBMS)* dan program-program aplikasi.
4. Pengguna (*user*) yaitu orang yang berinteraksi secara tidak langsung dengan basis data melalui program aplikasi basis data (*DBMS*).

Attribute adalah karakteristik spesifik dari suatu entitas, seperti pelanggan atau produk, dan setiap entitas dapat memiliki beberapa attribute. Secara teknis, *attribute* sering disebut sebagai item data atau kolom dan merupakan unit terkecil dalam file maupun *database*. *Attribute* ini memiliki nilai, seperti nama depan, alamat jalan, kota, dan tanggal kunjungan terakhir. Kunci adalah salah satu data item dalam sebuah *record* yang digunakan untuk mengidentifikasi *record* tersebut.

Berikut beberapa jenis kunci yang terdapat dalam basis data:

1. Kunci Utama (*Primary Key*)

Kunci yang secara unik dapat mengidentifikasi sebuah *record*. Contohnya, ORDER-# bisa menjadi kunci utama karena hanya memiliki satu nomor untuk setiap pesanan.

Kunci utama harus terbatas dan tidak mencakup atribut yang tidak diperlukan untuk mengidentifikasi *record*.

2. Kunci Kandidat (*Candidate Key*)

Atribut atau gabungan atribut yang dapat digunakan sebagai kunci utama.

3. Kunci Gabungan (*Concatenated/Composite Key*)

Kunci yang dibentuk dengan menggabungkan dua atau lebih data item, karena satu data item saja tidak cukup untuk mengidentifikasi *record* secara unik.

4. Kunci Sekunder (*Secondary Key*)

Kunci yang tidak mampu mengidentifikasi *record* secara unik, tetapi dapat digunakan untuk memilih sekelompok *record* tertentu, seperti semua pesanan dari suatu wilayah.

5. Kunci Asing (*Foreign Key*)

Kunci asing adalah *field* data dalam suatu file yang merupakan kunci utama dari file master lain. Kunci asing digunakan untuk menghubungkan tabel-tabel dalam basis data relasional.

2.10 Penjualan

Penjualan adalah proses pertukaran produk dan jasa kepada pelanggan dengan tujuan mendapatkan keuntungan bisnis. Penjualan memerlukan interaksi antar penjual dengan pembeli, dimana biasanya terjadi kegiatan tawar menawar atau negosiasi harga. Penjualan dapat dikatakan terjadi jika pembeli dan penjual telah mencapai kesepakatan yang disepakati oleh kedua belah pihak [12]. Penjualan merupakan aktivitas perusahaan untuk menukar produk atau jasa dengan konsumen sehingga menghasilkan pendapatan dan mendukung keberlangsungan usaha. Penjualan melibatkan proses komunikasi, penawaran nilai produk, dan penyelesaian transaksi yang terencana serta terukur. Efektivitas penjualan sangat dipengaruhi oleh perilaku tenaga penjual, khususnya kemampuan menyesuaikan pendekatan dengan karakteristik pelanggan. Perilaku penjualan adaptif terbukti meningkatkan kinerja penjualan karena membantu tenaga penjual merespons kebutuhan konsumen secara tepat [12]. Keterampilan penjualan seperti komunikasi, pemahaman produk, dan kemampuan membangun hubungan berpengaruh signifikan terhadap hasil penjualan. Tenaga penjual dengan kompetensi tinggi cenderung mencapai target penjualan yang lebih baik. Selain faktor individu, strategi pemasaran seperti promosi dan penetapan harga juga memengaruhi volume penjualan secara nyata. Kombinasi strategi penjualan yang tepat dan pengelolaan

tenaga penjual yang efektif menjadi faktor kunci dalam meningkatkan kinerja penjualan perusahaan [13].

2.11 Pembelian

Pembelian merupakan pertukaran produk dan jasa kepada pemasok dan perusahaan dengan menggunakan uang sebagai alat transaksi sah. Pembelian dilakukan untuk mendapatkan bahan baku, perlengkapan, atau produk untuk mendukung kegiatan transaksi pada perusahaan [14]. Pembelian didefinisikan sebagai tindakan membeli produk atau jasa dengan tujuan memenuhi kebutuhan atau keinginan tertentu dalam sebuah organisasi atau perusahaan. Dalam konteks bisnis, pembelian lebih menekankan pada transaksi untuk mendukung kelangsungan operasional bisnis, bukan sekadar pembelian konsumtif pribadi. Pembelian merujuk pada proses memperoleh produk dan jasa yang diperlukan perusahaan, yang melibatkan tahapan mulai dari identifikasi kebutuhan, seleksi pemasok, pemeriksaan produk/jasa yang diterima, hingga pembayaran [15]. Pengelolaan pembelian yang efektif dapat menekan biaya, meningkatkan kualitas input, mempercepat operasional, serta menjaga hubungan baik dengan pemasok. Sebaliknya, pengelolaan yang buruk berpotensi menyebabkan pemborosan, kelebihan stok, atau kekurangan pasokan yang mengganggu kelancaran bisnis. Pembelian tidak hanya merupakan aktivitas transaksional, melainkan elemen strategis yang memengaruhi daya saing dan efisiensi perusahaan secara keseluruhan [15].

2.12 Persediaan

Persediaan merupakan salah satu elemen vital yang menentukan kelancaran operasional perusahaan. Persediaan didefinisikan sebagai stok bahan baku, barang dalam proses, atau barang jadi yang disimpan untuk memenuhi kebutuhan produksi atau permintaan pelanggan pada periode tertentu. Pengelolaan persediaan yang tepat berdampak langsung pada efisiensi biaya dan kepuasan pelanggan. Jika persediaan terlalu sedikit, perusahaan berisiko kehilangan penjualan akibat kehabisan stok. Sebaliknya, jika persediaan terlalu banyak, perusahaan menanggung biaya penyimpanan yang tinggi dan risiko

kerusakan barang. Oleh karena itu, pemilihan metode pencatatan dan pengelolaan persediaan menjadi keputusan strategis yang wajib dipahami oleh pelaku usaha [15].

Terdapat berbagai metode yang digunakan untuk menentukan nilai persediaan serta urutan barang keluar dari gudang. Pemilihan metode ini akan mempengaruhi keuntungan dan efisiensi operasional. Metode-metode tersebut adalah [15]:

1. Metode *FIFO* (*First In, First Out*)

Metode *FIFO* merupakan metode yang menerapkan asumsi bahwa barang yang pertama kali masuk ke gudang adalah barang yang akan pertama kali keluar untuk dijual atau diproduksi. Dalam metode ini, aliran barang fisik diatur secara kronologis. *FIFO* sangat cocok diterapkan pada barang-barang yang memiliki sifat mudah rusak atau memiliki umur simpan terbatas, seperti makanan segar atau bahan baku kimia tertentu. Dengan *FIFO*, perusahaan dapat meminimalisir risiko kerusakan barang karena stok lama akan diproses terlebih dahulu.

2. Metode *LIFO* (*Last In, First Out*)

Metode *LIFO* mengasumsikan bahwa barang yang terakhir kali masuk ke gudang adalah barang yang pertama kali keluar. Dalam konteks perekonomian dengan tingkat inflasi yang tinggi, *LIFO* sering digunakan untuk keperluan pencatatan akuntansi karena beban pokok penjualan akan tercatat lebih tinggi (menggunakan harga terbaru), sehingga laba kena pajak bisa ditekan. Namun, dari sisi operasional fisik di gudang, metode ini jarang diterapkan pada barang-barang yang memiliki tenggat waktu kedaluwarsa karena berisiko menyebabkan stok lama menumpuk dan rusak.

3. Metode *Average* (rata-rata)

Metode *Average* adalah pendekatan yang menghitung nilai persediaan dengan cara membagi total biaya barang yang tersedia untuk dijual dengan total unit yang tersedia. Setiap kali ada pembelian barang baru dengan harga yang berbeda, sistem akan menghitung ulang rata-rata biaya per unit. Metode ini sering digunakan ketika harga bahan baku mengalami fluktuasi yang signifikan dan perusahaan ingin menstabilkan nilai pencatatan persediaan. Metode ini memberikan gambaran nilai persediaan yang lebih stabil dibandingkan *FIFO* atau *LIFO* yang fluktuatif mengikuti harga pasar.