

BAB II

KAJIAN LITERATUR

2.1 Konsep Sistem Informasi

2.1.1 Sistem

Sistem adalah kesatuan yang terorganisir dari manusia, mesin, dan metode yang dirancang untuk menjalankan fungsi pemrosesan data tertentu. Setiap sistem memiliki tujuan besar (*goal*), sementara setiap subsistem di dalamnya memiliki sasaran yang lebih spesifik (*objectives*). Tujuan dan sasaran ini merupakan target positif yang mengarah pada perbaikan untuk memenuhi kebutuhan dan keinginan, dengan tolok ukur keberhasilannya adalah kepuasan pengguna. Sebuah sistem memiliki beberapa karakteristik atau komponen pembangun, yaitu [5]:

1. Subsistem (*Subsystems*)

Sistem terdiri dari elemen-elemen yang saling berinteraksi dan bekerja sama untuk mencapai tujuan. Setiap subsistem memiliki fungsi khusus yang memengaruhi kinerja sistem secara keseluruhan. Aktivitas dalam subsistem meliputi masukan, pengolah, dan keluaran.

a. Masukan (*input*)

Masukan adalah segala bentuk energi yang dimasukkan ke dalam sistem. Terdapat dua jenis masukan: *maintenance input* dan *signal input*. *Maintenance input* adalah energi yang diperlukan agar sistem dapat berjalan, sementara *signal input* adalah energi yang akan diproses untuk menghasilkan keluaran. Sebagai contoh, pada sistem komputer, program adalah *maintenance input* yang berfungsi mengoperasikan komputer, sedangkan data adalah *signal input* yang diolah untuk menjadi informasi.

b. Pengolah (*process*)

Pengolah merupakan bagian dalam sistem yang bertugas mengubah masukan menjadi keluaran yang diinginkan. Contohnya, dalam sistem produksi, unit pengolah mengubah bahan baku menjadi produk jadi. Dalam sistem akuntansi, data transaksi diolah menjadi laporan keuangan dan laporan lain yang diperlukan manajemen.

c. Keluaran (*output*)

Keluaran adalah hasil dari pemrosesan masukan. Keluaran ini dapat dikelompokkan menjadi hasil yang bermanfaat dan sisa atau produk buangan yang tidak digunakan. Sebuah keluaran dari satu subsistem dapat menjadi masukan bagi subsistem lainnya.

Contohnya, pada sistem komputer, informasi adalah keluaran yang berguna, sementara data yang tidak relevan atau usang dapat dianggap sebagai sisa pembuangan.

2. Batasan sistem (*boundary*)

Batasan sistem adalah garis pemisah yang mendefinisikan wilayah kerja suatu sistem dengan sistem lainnya atau dengan lingkungan luarnya. Batasan ini menegaskan ruang lingkup (*scope*) sebuah sistem, yang memungkinkannya dilihat sebagai satu kesatuan utuh. Sering kali, nama sistem itu sendiri sudah mencerminkan batasannya, misalnya sistem pendidikan memiliki batasan pada penyelenggaraan pendidikan.

3. Lingkungan luar sistem (*environments*)

Lingkungan luar adalah segala sesuatu yang berada di luar batasan sistem, tetapi memengaruhi operasional sistem tersebut baik secara langsung maupun tidak langsung. Lingkungan luar ini bisa bersifat menguntungkan ataupun merugikan. Sebagai contoh, bagi sebuah sistem perbankan, nasabah, kebijakan pemerintah, dan kondisi ekonomi merupakan bagian dari lingkungan luar yang sangat berpengaruh.

4. Penghubung sistem (*interface*)

Penghubung sistem adalah media yang menyambungkan satu subsistem dengan subsistem lainnya. Penghubung ini memungkinkan aliran sumber daya di antara subsistem, di mana keluaran dari satu subsistem menjadi masukan bagi subsistem lain. Dengan demikian, penghubung berfungsi untuk mengintegrasikan seluruh subsistem menjadi satu kesatuan yang solid. Contoh dari penghubung adalah kurir, memo, telepon, atau jaringan komunikasi lainnya.

5. Tujuan sistem (*goal*)

Tujuan adalah hasil akhir yang ingin dicapai oleh sistem untuk memenuhi kebutuhan (*need*) dan keinginan (*want*), sehingga seluruh aktivitasnya diarahkan untuk memberikan manfaat atau nilai guna. Sementara itu, sasaran (*objectives*) ditetapkan pada setiap subsistem. Pencapaian sasaran ini sangat ditentukan oleh kesesuaian antara masukan yang diproses dan keluaran yang dihasilkan. Sebagai contoh, jika tujuan utama sistem adalah meningkatkan produktivitas, maka sasaran pada setiap subsistem harus dirancang untuk mendukung pencapaian tujuan tersebut.

Sistem merupakan kesatuan elemen yang saling berinteraksi untuk mencapai tujuan tertentu, yang meliputi *input*, proses, *output*, batasan, lingkungan, penghubung dan tujuan sistem. Konsep sistem menegaskan bahwa aktivitas dalam suatu usaha perlu disusun menjadi serangkaian proses yang saling berkaitan sehingga operasi dapat berjalan secara efektif dan terarah.

2.1.2 Informasi

Kata "informasi" berasal dari kata Latin "*informatio*", yang berarti gagasan atau garis besar. Kata ini juga berasal dari kata kerja "*informare*", yang berarti membentuk, menginstruksikan, atau memberi bentuk pada sesuatu. Seiring waktu, maknanya berkembang menjadi pengetahuan yang diberikan atau diterima melalui komunikasi dalam bentuk data, pesan, atau fakta. Informasi berfungsi untuk mengurangi ketidakpastian, memberi konteks, dan menambah pemahaman. Secara etimologis, informasi dapat dipahami sebagai hasil dari tindakan membentuk pikiran yang kemudian meluas menjadi pengetahuan yang disampaikan. Adapun manfaat dari informasi, di antaranya [6]:

1. Menyampaikan pengetahuan: Informasi berfungsi sebagai media untuk menyebarkan ilmu, keterampilan, dan pengalaman.
2. Menghilangkan ketidakpastian: Dengan memberikan data dan fakta yang andal, informasi membantu mengurangi ketidakpastian sehingga orang bisa membuat prediksi dan perencanaan yang lebih baik.
3. Alat hiburan: Informasi juga memiliki peran penting dalam hiburan dan rekreasi.
4. Mendukung inovasi: Informasi adalah bahan baku untuk inovasi, memungkinkan organisasi mengembangkan produk, layanan, atau proses baru yang lebih efektif.
5. Mendukung pengambilan keputusan: Informasi yang relevan dan akurat membantu individu dan organisasi membuat keputusan yang tepat.
6. Memelihara hubungan sosial: Berbagi informasi memperkuat ikatan sosial dan memfasilitasi kolaborasi yang lebih baik.
7. Mengontrol dan mengawasi: Dalam organisasi, informasi digunakan untuk memantau kinerja dan membuat penyesuaian yang diperlukan untuk mencapai tujuan.
8. Menghubungkan sumber daya: Informasi menghubungkan sumber daya yang berbeda untuk mencapai tujuan, misalnya mengarahkan sumber daya bisnis ke peluang yang paling menguntungkan.
9. Pendidikan: Dalam pendidikan, informasi adalah dasar dari pembelajaran, membantu siswa mempelajari konsep baru dan mengembangkan keterampilan.
10. Memfasilitasi transparansi: Informasi yang terbuka dan mudah diakses memungkinkan transparansi dalam berbagai proses, baik dalam pemerintahan, bisnis, maupun organisasi lainnya, yang mendukung kepercayaan dan akuntabilitas.

Informasi merupakan hasil pengolahan data yang memiliki makna serta mendukung proses pengambilan keputusan. Suatu informasi dikatakan berkualitas jika informasi tersebut akurat, tepat waktu, dan relevan. Ketersediaan informasi yang baik sangat penting dalam

mendukung aktivitas operasional yang memerlukan pengawasan dan penilaian data secara berkelanjutan. Informasi yang dihasilkan sistem akan digunakan untuk membantu pemilik Toko Kopi Sidikalang dalam mengambil keputusan, misalnya keputusan pembelian bahan baku, produksi, dan penjualan.

2.1.3 Sistem Informasi

Sistem informasi adalah gabungan dari teknologi, prosedur, manusia, dan data yang dirancang untuk mendukung operasional, manajerial, dan strategis suatu organisasi. Ini bukan hanya alat administratif, melainkan fondasi untuk pengambilan keputusan dan penyusunan strategi bisnis. Sistem ini membantu organisasi mengelola data, mengoptimalkan alur kerja, dan menciptakan keunggulan kompetitif. Secara ringkas, sistem informasi adalah sistem terintegrasi yang mengubah data menjadi informasi yang berguna untuk mendukung proses bisnis dan pengambilan keputusan. Elemen-elemen penting dari sistem informasi meliputi [7]:

1. Perangkat Keras (*Hardware*)

Perangkat keras merupakan bagian fisik dari sistem informasi yang meliputi komputer, server, perangkat *input-output* seperti *keyboard*, *mouse*, *scanner*, dan *printer*, serta perangkat jaringan. *Hardware* berfungsi untuk menjalankan perintah dari perangkat lunak, mengolah data, dan menyimpan informasi.

2. Perangkat Lunak (*Software*)

Perangkat lunak adalah kumpulan instruksi atau program yang digunakan untuk mengoperasikan komputer dan mengelola data. *Software* memungkinkan pengguna menjalankan berbagai fungsi bisnis dengan lebih efisien, akurat, dan terintegrasi.

3. Basis Data (*Database*)

Basis data merupakan tempat penyimpanan data yang tersusun secara sistematis dan dapat diakses dengan mudah. Data berfungsi sebagai bahan baku yang diolah menjadi informasi. *Database Management System* (DBMS) digunakan untuk memfasilitasi penyimpanan, pencarian, pemrosesan, dan pengelolaan data secara efektif dan aman.

4. Jaringan dan Komunikasi (*Network & Communication*)

Komponen ini memungkinkan terjadinya konektivitas antara berbagai perangkat dan sistem. Teknologi jaringan seperti LAN, WAN, dan internet memungkinkan pertukaran data dan komunikasi antara pengguna serta sistem informasi. Infrastruktur komunikasi ini mendukung kolaborasi dan mobilitas kerja.

5. Manusia (*People*)

Manusia merupakan elemen terpenting dalam sistem informasi, yang meliputi pengguna akhir (*end users*), analis sistem, administrator basis data, manajer TI, dan pengembang sistem. Manusia tidak hanya berperan dalam menjalankan sistem, tetapi juga dalam perancangan, pengembangan, dan pemeliharaan sistem informasi agar sesuai dengan kebutuhan organisasi.

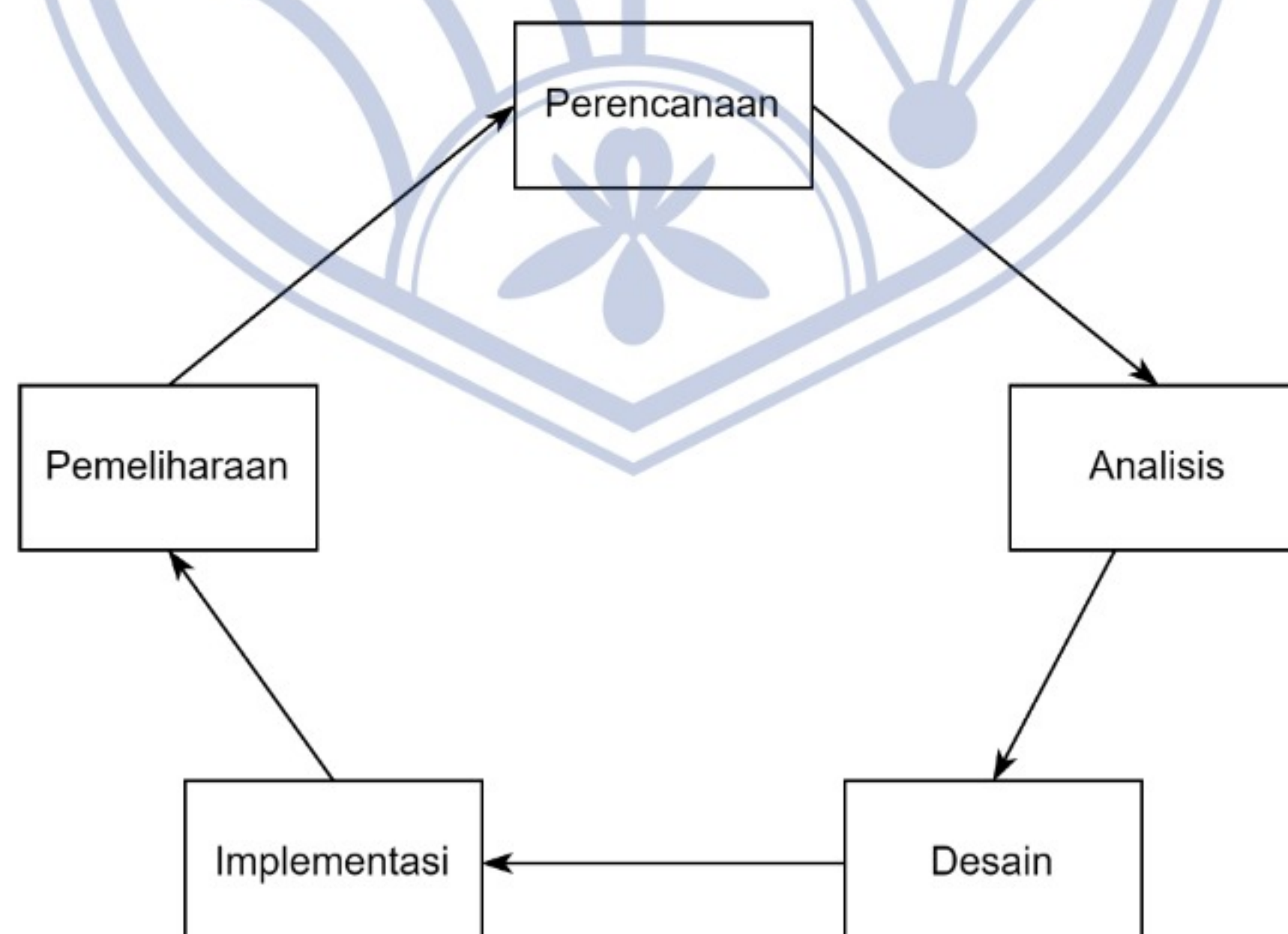
6. Prosedur (*Procedures*)

Prosedur adalah kumpulan aturan, kebijakan, dan langkah-langkah yang harus dipatuhi dalam mengoperasikan sistem informasi. Prosedur ini membantu memastikan sistem berjalan secara konsisten, aman, dan efisien.

Sistem informasi dapat dipahami sebagai gabungan dari teknologi, prosedur, manusia, dan data yang berfungsi mengumpulkan, mengolah, menyimpan, dan mendistribusikan informasi. Sistem informasi dapat membantu kegiatan operasional agar lebih cepat dan akurat. Dalam penelitian ini, sistem informasi akan menggantikan proses manual di Toko Kopi Sidikalang agar pencatatan data menjadi teratur dan mudah diakses.

2.2 *System Development Life Cycle (SDLC)*

System Development Life Cycle (SDLC) adalah tahapan yang digunakan untuk membuat dan mengembangkan sistem informasi secara teratur dari awal hingga akhir. SDLC membantu pengembang untuk menghasilkan sistem yang sesuai dengan kebutuhan pengguna. Tahapan *System Development Life Cycle (SDLC)* ditunjukkan pada gambar berikut ini yang terdiri dari [8]:



Gambar 2.1 Tahapan *System Development Life Cycle (SDLC)*

1. Fase Perencanaan: fase awal di mana tim analis sistem melakukan identifikasi masalah, peluang, dan tujuan bisnis yang ingin dicapai. Tim analis menentukan ruang lingkup proyek, jadwal kerja, sumber daya, dan biaya yang diperlukan.
2. Fase Analisis: fase di mana tim analis sistem mempelajari sistem yang ada dan mengumpulkan kebutuhan pengguna agar sistem yang dibangun sesuai dengan kebutuhan mereka. Kegiatan utamanya meliputi wawancara, observasi, dan pembuatan model proses agar kebutuhan sistem dapat dipahami dengan jelas.
3. Fase Desain: fase ini menerjemahkan kebutuhan pengguna menjadi rancangan teknis dan membuat desain antarmuka, basis data, struktur sistem, serta menentukan cara kerja sistem secara detail sebelum dibangun.
4. Fase Implementasi: fase di mana sistem mulai dibangun melalui *coding*, pengujian, dan pemasangan sistem. Setelah selesai, pengguna diberi pelatihan agar dapat mengoperasikan sistem dengan baik.
5. Fase Pemeliharaan: fase di mana setelah sistem dijalankan, akan dilakukan pemantauan dan perawatan agar tetap berjalan dengan baik. Kegiatan ini meliputi perbaikan kesalahan, penyesuaian terhadap perubahan kebutuhan, dan peningkatan fitur sistem.

System Development Life Cycle (SDLC) adalah tahapan pengembangan sistem yang terdiri dari tahap perencanaan, analisis, perancangan, implementasi, dan pemeliharaan. Tahapan ini membantu pengembangan sistem agar berjalan teratur dan sesuai kebutuhan pengguna. Tahapan SDLC ini digunakan karena dapat membantu memastikan bahwa sistem dirancang dengan mempertimbangkan kebutuhan pengguna dan sesuai dengan proses pengembangan sistem untuk mendukung kegiatan operasional toko.

2.3 Teknik Pengembangan Sistem

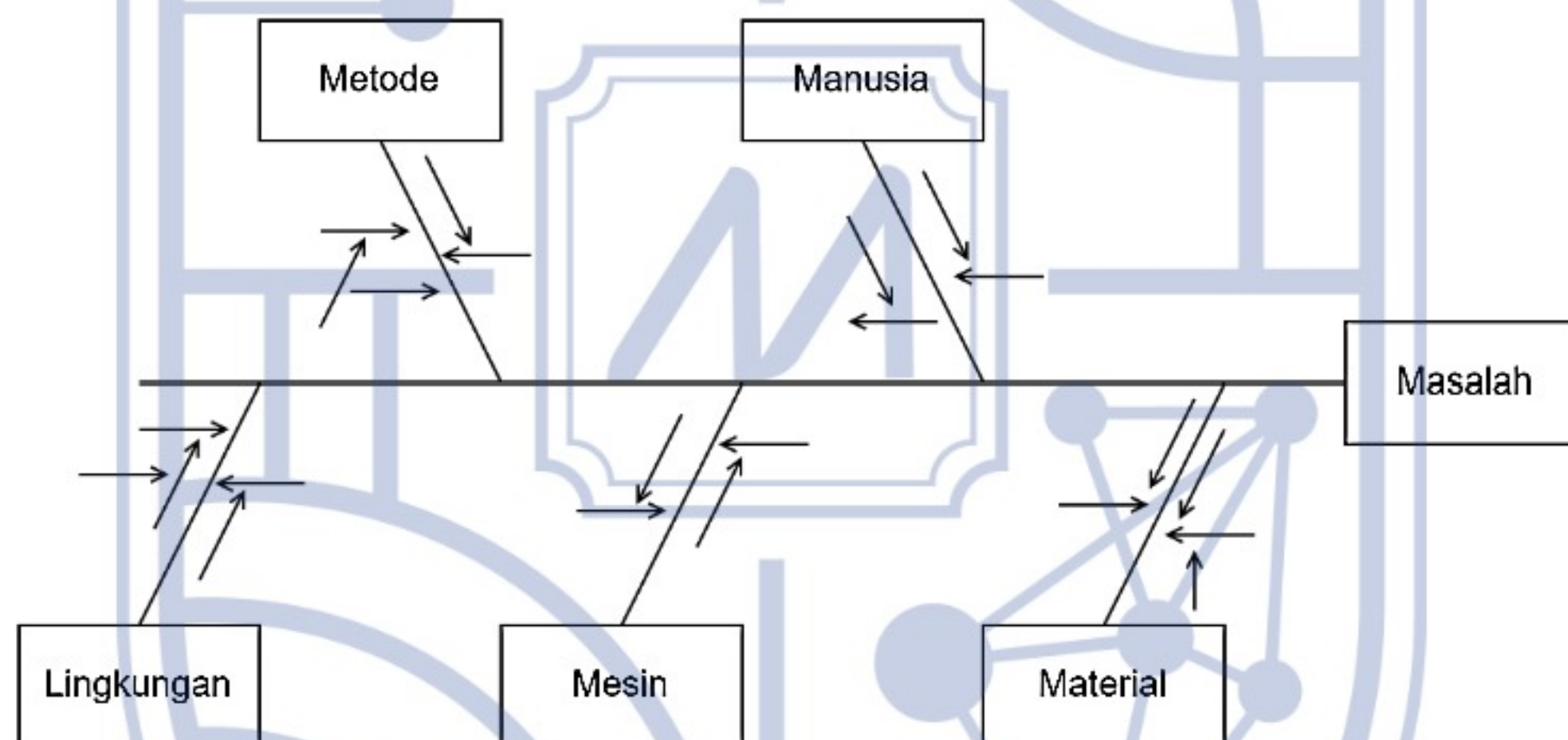
2.3.1 Diagram Fishbone

Diagram *Fishbone* atau diagram sebab-akibat, yang juga dikenal sebagai diagram *ishikawa*, merupakan salah satu dari *seven basic quality tools* yang umumnya digunakan untuk mengidentifikasi dan analisis akar penyebab dari suatu masalah. Diagram ini mengorganisir penyebab-penyebab potensial dari suatu efek spesifik kemudian memisahkannya hingga menemukan akar permasalahan [9]. Analisis ini juga memungkinkan adanya pendekatan terperinci dan terstruktur dalam penelitian.

Kata “*Fishbone*” berasal dari bentuknya yang menyerupai tulang ikan. “Kepala Ikan” mewakili masalah yang sedang dianalisis, sementara tulang-tulang utamanya merepresentasikan kategori penyebab yang dominan. Faktor masalah ini kemudian diperinci

lagi menjadi cabang-cabang masalah yang lebih kecil [10]. Untuk mempermudah identifikasi masalah, tulang utamanya terdiri dari [11]:

1. *Man* (manusia): menggambarkan orang yang terlibat atau berperan dalam sebuah proses.
2. *Method* (metode): menjelaskan metode atau bagaimana cara suatu proses dilakukan, seperti prosedur, atau peraturan.
3. *Material* (bahan): menjelaskan semua bahan yang dibutuhkan dalam menjalankan sebuah proses.
4. *Machine* (mesin): mesin, komputer, atau peralatan lain yang terlibat dalam berjalannya suatu proses.
5. *Environment* (lingkungan): lingkungan di sekitar area kerja, dapat berupa kondisi lingkungan seperti temperatur udara, kelembaban, atau pencahayaan. Berikut gambar diagram *fishbone*.



Gambar 2.2 Diagram *Fishbone*

Berdasarkan gambar diagram *fishbone* di atas, terdapat langkah-langkah dalam menyusun diagram *fishbone* meliputi [10]:

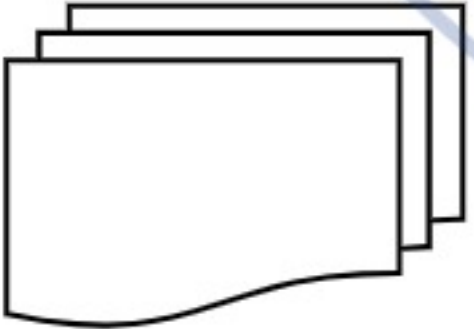
1. Mendefinisikan masalah atau kendala yang akan dianalisis.
2. Membentuk tim untuk menganalisis masalah, di mana seluruh anggota tim berpartisipasi dan mengemukakan pendapat mereka mengenai penyebab masalah.
3. Menggambarkan kerangka utama diagram, dan memberi label dari masalah yang telah didefinisikan.
4. Tentukan kategori penyebab utamanya baik *man*, *method*, *material*, *machine* atau *environment* dan hubungkan sebagai cabang yang terhubung ke garis pusat.
5. Seluruh anggota melakukan *brainstorming* untuk menemukan penyebab-penyebab spesifik dan klasifikasikan ke kategori penyebab utama yang sesuai.

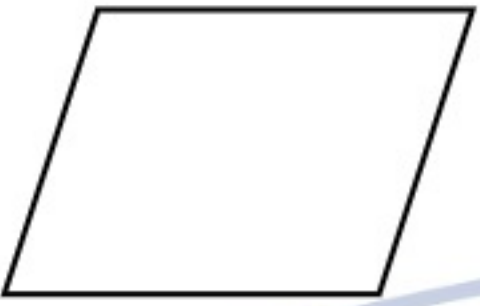
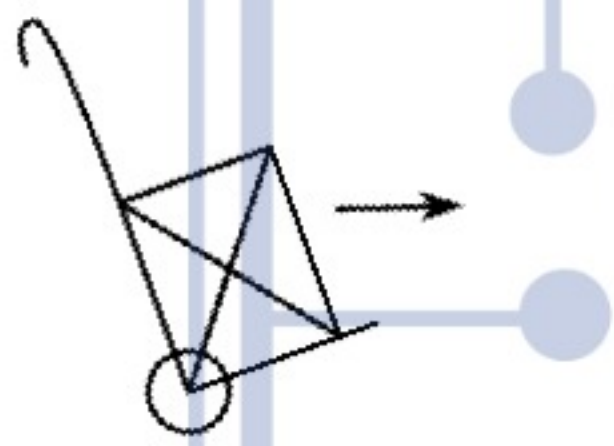
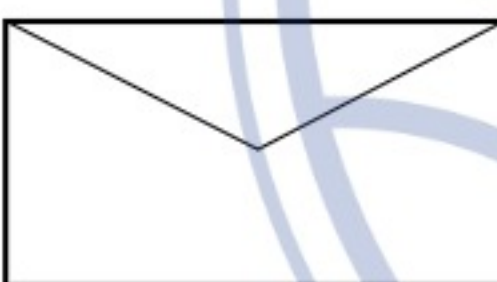
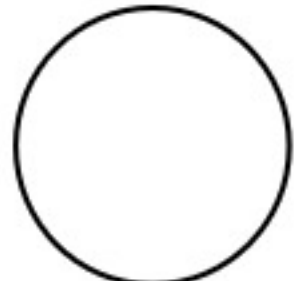
Diagram *Fishbone* merupakan alat yang digunakan untuk mencari penyebab utama suatu masalah. Diagram ini dapat membantu Toko Kopi Sidikalang untuk mengetahui penyebab masalah pada proses pembelian, persediaan, produksi, dan penjualan. Analisis tersebut berguna dalam memahami kendala operasional toko dan menyediakan dasar untuk merancang solusi yang tepat. Setelah semua akar masalah teridentifikasi, dilanjutkan dengan merencanakan tindakan perbaikan untuk mengatasi masalah yang telah didefinisikan.

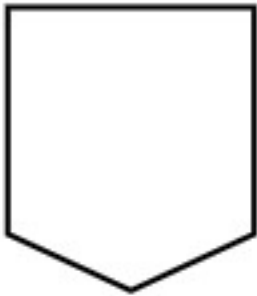
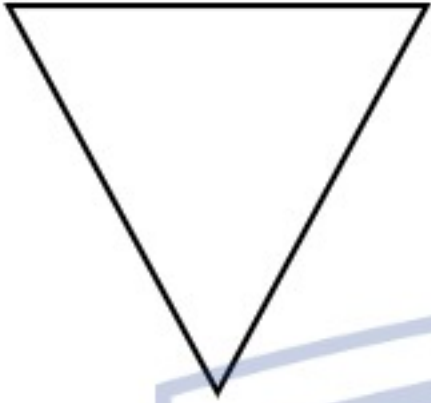
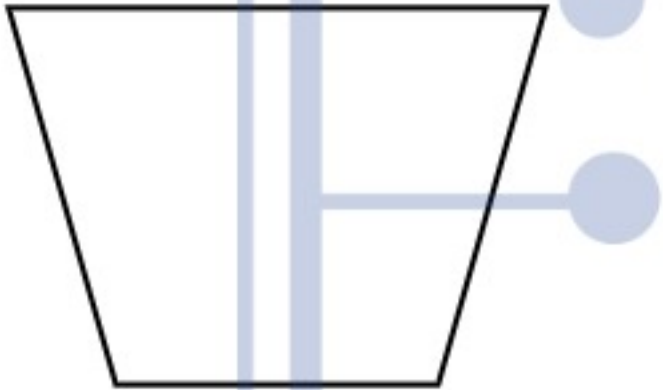
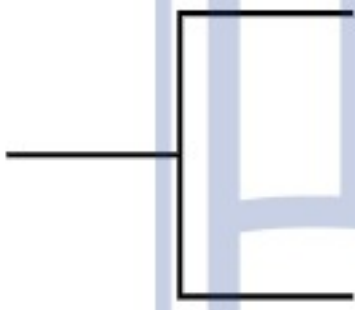
2.3.2 Flow of Document (FOD)

Flow of Document (FOD) atau bagan alir dokumen merupakan alat bantu visual yang digunakan untuk menggambarkan alur pergerakan dokumen dalam suatu organisasi, baik dalam bentuk kertas maupun elektronik. Diagram ini menunjukkan bagaimana dokumen dibuat, diproses, didistribusikan, dan diarsipkan di berbagai bagian organisasi. Melalui bentuk grafisnya, FOD mempermudah pemahaman terhadap jalannya proses administrasi, memperlihatkan pihak yang terlibat, serta hubungan antara satu kegiatan dengan kegiatan lainnya. Berikut berbagai simbol dan fungsinya, di antaranya [12]:

Tabel 2.1 Notasi Diagram Alir Dokumen

Notasi	Keterangan	Fungsi
	<i>Document</i>	Untuk menunjukkan satu lembar dokumen yang dibuat, diterima, atau digunakan dalam proses bisnis.
	<i>Multiple copies of a specific document</i>	Untuk menunjukkan bahwa dokumen memiliki beberapa salinan yang dikirim ke bagian berbeda.
	<i>Document flow</i>	Garis panah untuk menunjukkan arah pergerakan dokumen antar proses.

	<i>Information flow (without document flow)</i>	Garis putus-putus yang menunjukkan aliran informasi tanpa adanya dokumen fisik.
	<i>Journal or ledger</i>	Untuk menunjukkan buku jurnal atau buku besar tempat pencatatan transaksi dilakukan.
	<i>Flow of physical goods (inventory, etc.)</i>	Untuk menunjukkan perpindahan barang atau inventaris yang terkait dengan dokumen.
	<i>Keying operation</i>	Untuk menggambarkan kegiatan pengetikan atau <i>input</i> data ke dalam sistem komputer.
	<i>Envelope for mailing</i>	Untuk menunjukkan proses pengiriman dokumen melalui surat, faktur, atau cek ke pihak lain.
	<i>Adding machine tape</i>	Sebagai simbol totalisasi angka yang berbentuk seperti pita untuk pengendalian <i>batch</i> atau pengecekan jumlah.
	<i>On-page connector</i>	Untuk menghubungkan dua titik aliran pada halaman yang sama agar diagram lebih rapi

	<i>Off-page connector</i>	Untuk menghubungkan alur dokumen antar halaman berbeda dalam <i>flowchart</i> .
	<i>Permanent file of manual documents</i>	Untuk menunjukkan tempat penyimpanan dokumen permanen, diurutkan menurut Alfabet (A), Numerik (N), atau Tanggal (D).
	<i>Manual operation</i>	Untuk menunjukkan proses manual yang dilakukan oleh manusia, seperti mencatat, memeriksa, atau menghitung.
	<i>Annotation</i>	Untuk simbol tambahan untuk menuliskan catatan, komentar, atau keterangan penting pada diagram.
	<i>Electronic Data Communication</i>	Untuk menunjukkan komunikasi atau pertukaran data secara elektronik antar bagian atau sistem.

1. *Document*: menunjukkan satu lembar dokumen yang dibuat, diterima, atau digunakan dalam proses bisnis.
2. *Multiple copies of a specific document*: menunjukkan bahwa dokumen memiliki beberapa salinan yang dikirim ke bagian berbeda.
3. *Document flow*: garis panah untuk menunjukkan arah pergerakan dokumen antar proses.
4. *Information flow (without document flow)*: garis putus-putus yang menunjukkan aliran informasi tanpa adanya dokumen fisik.
5. *Journal or ledger*: menunjukkan buku jurnal atau buku besar tempat pencatatan transaksi dilakukan.

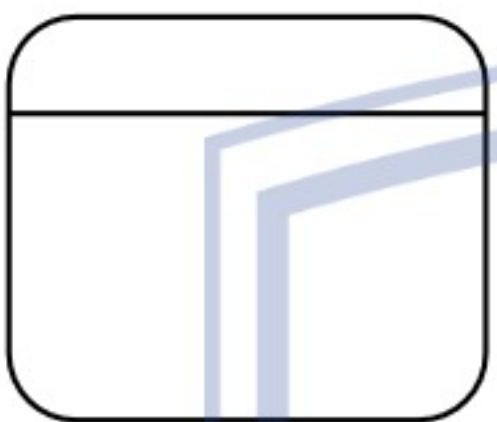
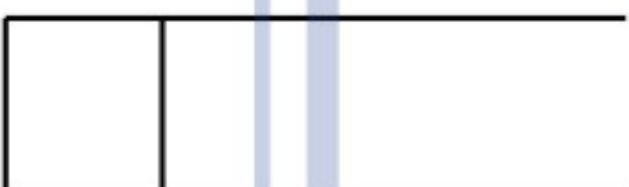
6. *Flow of physical goods*: menunjukkan perpindahan barang atau inventaris yang terkait dengan dokumen.
7. *Keying operation*: menggambarkan kegiatan pengetikan atau input data ke dalam sistem komputer.
8. *Envelope for mailing*: menunjukkan proses pengiriman dokumen melalui surat, faktur, atau cek ke pihak lain.
9. *Adding machine tape*: simbol totalisasi angka yang berbentuk seperti pita untuk pengendalian *batch* atau pengecekan jumlah.
10. *On-page connector*: digunakan untuk menghubungkan dua titik aliran pada halaman yang sama agar diagram lebih rapi.
11. *Off-page connector*: menghubungkan alur dokumen antar halaman berbeda dalam flowchart.
12. *Permanent file of manual documents*: menunjukkan tempat penyimpanan dokumen permanen, diurutkan menurut Alfabet (A), Numerik (N), atau Tanggal (D).
13. *Manual operation*: menunjukkan proses manual yang dilakukan oleh manusia, seperti mencatat, memeriksa, atau menghitung.
14. *Annotation*: simbol tambahan untuk menuliskan catatan, komentar, atau keterangan penting pada diagram.
15. *Electronic Data Communication*: menunjukkan komunikasi atau pertukaran data secara elektronik antar bagian atau sistem.

Flow of Document (FOD) menggambarkan alur pergerakan dokumen antar bagian dalam suatu proses. FOD dapat membantu menggambarkan alur dokumen pada Toko Kopi Sidikalang dari pembelian hingga penjualan sehingga sistem yang dirancang dapat mengikuti kondisi sebenarnya.

2.3.3 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah representasi visual untuk menjelaskan fungsi, aliran data, penyimpanan data, dan proses yang terjadi dalam sebuah sistem. DFD menunjukkan bagaimana suatu data berpindah melalui sistem informasi, tanpa menunjukkan logika program atau tahapan pemrosesan secara rinci. DFD memanfaatkan berbagai simbol untuk menunjukkan bagaimana sebuah sistem mengubah data *input* menjadi informasi yang bermanfaat. Berbagai simbol yang digunakan atau yang biasa disebut dengan notasi dibagi menjadi 4 simbol, di antaranya [13]:

Tabel 2.2 Notasi *Data Flow Diagram (DFD)*

Notasi	Keterangan	Fungsi
	<i>External Entity</i>	Untuk memvisualisasikan batasan sistem serta interaksinya dengan lingkungan eksternal.
	<i>Process</i>	Untuk menerima data masukan dan mengolahnya untuk menghasilkan keluaran dengan konten yang berbeda.
	<i>Data Store</i>	Untuk merepresentasikan data yang sedang tidak digunakan atau disimpan oleh sistem untuk diakses.
	<i>Data Flow</i>	Untuk merepresentasikan pergerakan satu atau lebih item data dari satu komponen ke komponen lainnya dalam diagram.

Untuk membuat sebuah DFD yang logis dan tepat, terdapat serangkaian aturan dasar yang mengatur bagaimana setiap komponen dapat berinteraksi satu sama lain. Aturan ini berfungsi untuk memastikan bahwa alur data dalam sistem tergambar dengan benar, konsisten, dan mudah dipahami. Berikut terdapat aturan-aturan yang harus diperhatikan dalam membuat DFD, yaitu [13]:

1. *External Entity*

Entitas eksternal hanya boleh terhubung dengan komponen *process* dan tidak boleh terhubung langsung dengan entitas eksternal lain atau *data store*.

2. *Process*

Aturan dalam penggunaan *process*, yaitu:

- a. Setiap proses wajib memiliki minimal satu *input* dan satu *output*.

- b. Tidak boleh ada proses yang menghasilkan *output* tanpa adanya *input* (*Spontaneous Generation*).
- c. Tidak boleh ada proses yang menerima *input* tetapi tidak menghasilkan *output* apapun (*black hole*).
- d. Tidak boleh terdapat proses *output* yang tidak logis atau tidak sesuai dengan *input* yang diterima (*gray hole*).

3. *Data Store*

Setiap aliran data yang berasal dari atau menuju ke *data store* harus melalui sebuah *process*. Komponen ini tidak boleh terhubung langsung dengan komponen lain selain *process*.

4. *Data Flow*

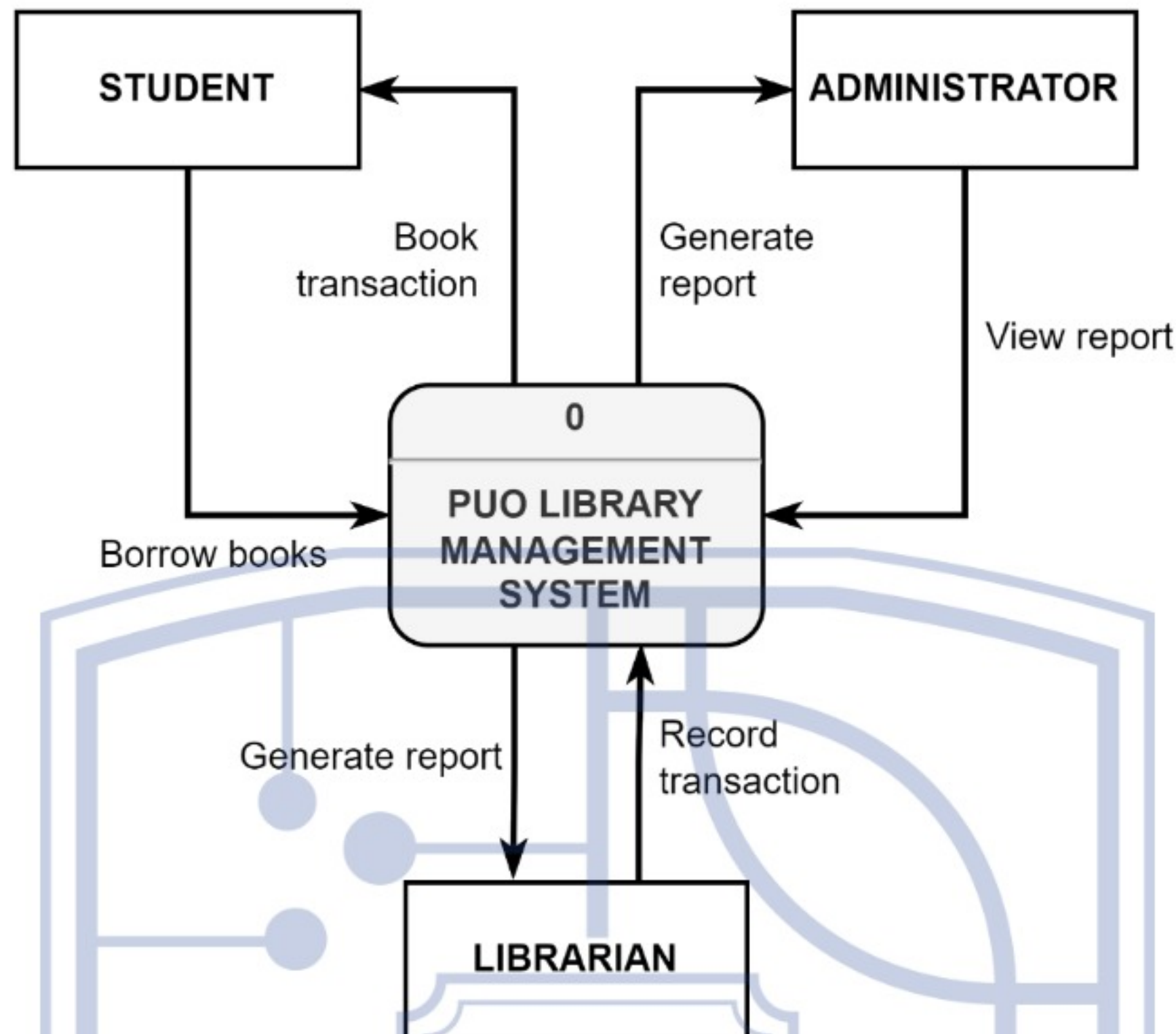
Aliran data harus berasal dari atau menuju ke sebuah *process*. Komponen ini tidak boleh menghubungkan secara langsung dengan 2 entitas eksternal, 2 *data store*, atau entitas eksternal dengan *data store*.

Pembuatan *Data Flow Diagram* (DFD) merupakan sebuah proses terstruktur yang dilakukan secara bertahap dan sistematis dengan menggunakan pendekatan *top-down* atau dekomposisi. Berikut tahapan dalam menggambarkan sebuah DFD, yaitu [13]:

1. **Membuat Diagram Konteks (*Context Diagram*)**

Diagram Konteks merupakan level tertinggi dan paling dasar dari DFD yang berfungsi untuk memberikan gambaran umum sebuah sistem informasi secara keseluruhan. Tujuan utama dari diagram ini adalah untuk menunjukkan batasan (*scope*) sistem serta hubungan interaksinya dengan pihak-pihak eksternal. Pada tahapan ini, keseluruhan sistem digambarkan sebagai satu proses tunggal.

Berikut contoh penggambaran Diagram Konteks:

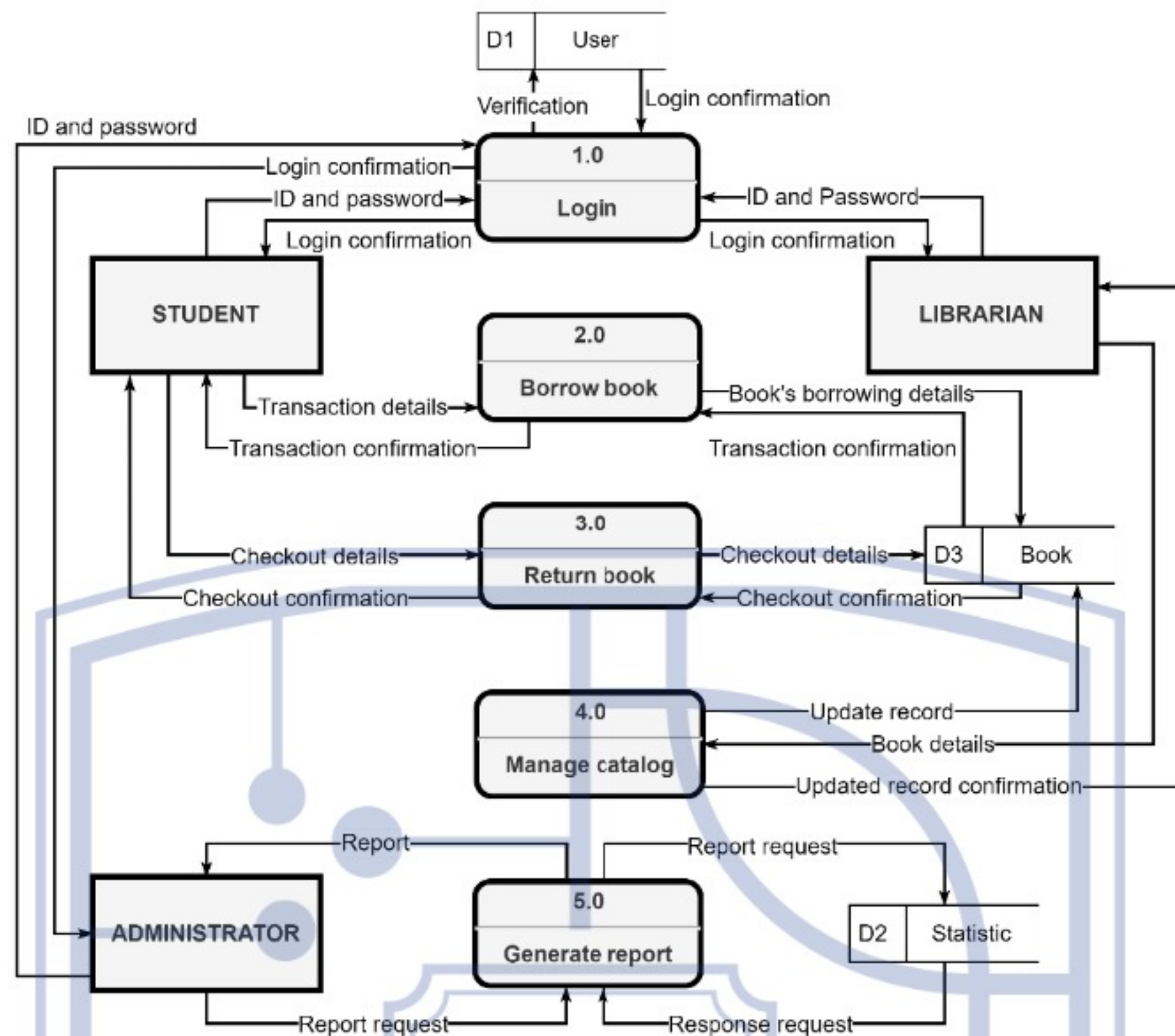


Gambar 2.3 Diagram Konteks

2. Membuat DFD Level 0

DFD Level 0 merupakan hasil dekomposisi pertama dari proses tunggal yang ada pada Diagram Konteks. Diagram ini menguraikan fungsi-fungsi utama sistem menjadi beberapa proses yang lebih spesifik. Setiap proses utama diberi nomor secara berurutan (misalnya 1.0, 2.0, 3.0) dan merepresentasikan modul atau fungsi inti dari sebuah sistem. Pada tahapan ini, komponen *data store* mulai diperlihatkan di mana data tersimpan dan diakses di dalam sistem. Hal terpenting dalam tahapan ini adalah menjaga keseimbangan prinsip di mana semua entitas eksternal, total aliran data masuk dan keluar sesuai dengan yang telah didefinisikan pada diagram konteks, serta harus tetap konsisten dan terwakili secara tepat dalam DFD Level 0.

Berikut contoh penggambaran DFD Level 0:

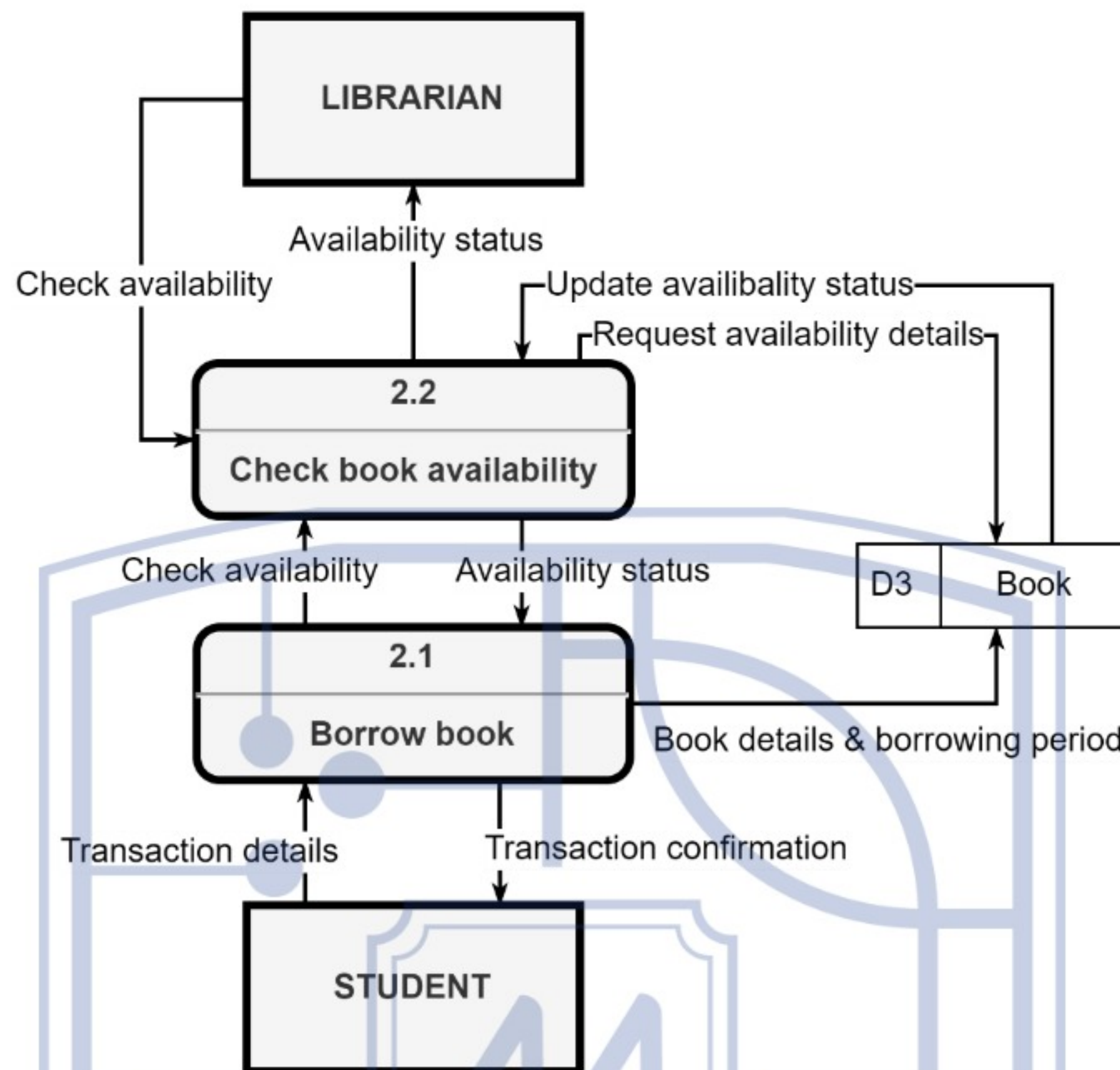


Gambar 2.4 DFD *Level 0*

3. Membuat DFD Level Turunan (*Lower-Level DFD*)

DFD Level Turunan dibuat untuk menguraikan lebih detail proses-proses yang ada pada DFD Level 0 yang masih bersifat terlalu kompleks menjadi bagian-bagian yang lebih rinci. Untuk setiap proses dari Level 0 dapat dipecah lagi menjadi beberapa subproses diagram level di bawahnya (Level 1, Level 2, dan seterusnya) yang berisi subproses yang lebih spesifik. Penomoran subproses ini mengikuti hierarki dari proses induknya. Misalnya, proses 2.0 dapat diuraikan menjadi subproses (2.1, 2.2, dan seterusnya). Seperti pada DFD Level 0, prinsip keseimbangan juga harus diterapkan pada tahapan ini, yang berarti total aliran data yang masuk dan keluar dari sebuah diagram level turunan harus sama persis dengan aliran data yang masuk dan keluar pada diagram level turunan harus sama dengan aliran data dari proses induk yang diuraikannya pada diagram level yang lebih tinggi.

Berikut contoh penggambaran DFD Level Turunan:



Gambar 2.5 DFD Level Turunan

DFD menggambarkan aliran data dari dan ke proses, entitas serta penyimpanan data dalam sistem. Dalam penelitian ini, DFD digunakan untuk menjelaskan hubungan antar proses dalam sistem informasi yang akan dibuat Toko Kopi Sidikalang sehingga memudahkan perancangan sistem informasi yang terintegrasi.

2.4 Konsep Basis Data

2.4.1 Basis Data

Basis data (*database*) adalah sekumpulan data yang tersusun secara teratur dan disimpan dalam sistem komputer. Basis data dapat diartikan sebagai himpunan data yang diorganisasi dengan rapi serta disimpan menggunakan struktur tertentu sehingga memudahkan proses akses, pengelolaan, maupun pembaruan data [14].

Secara konseptual, basis data dapat didefinisikan sebagai kumpulan data terstruktur yang saling berkaitan serta diatur secara logis untuk menggambarkan fakta-fakta dari dunia nyata. Tujuan utama dalam perancangan basis data adalah mengurangi duplikasi data, menjaga konsistensi serta keakuratan data, dan menyediakan sistem akses yang terjamin bagi berbagai pengguna maupun aplikasi [14].

Basis data berperan penting dalam arsitektur sistem informasi, terutama dalam menyediakan cara yang teratur dan efisien sebagai penyimpanan serta pengelolaan data. Selain itu, basis data juga berkontribusi langsung dalam mendukung proses pengambilan keputusan strategis. Ketersediaan data yang terintegrasi memungkinkan manajer dan pemangku kepentingan melakukan analisis secara menyeluruh, sehingga keputusan yang dihasilkan lebih tepat, akurat, dan didasarkan pada informasi faktual [14].

Suatu basis data dapat dihasilkan dengan baik apabila lingkungan terdapat komponen-komponen yang saling terintegrasi satu dengan lainnya. Komponen-komponen fundamental tersebut adalah sebagai berikut [14]:

1. Perangkat Keras (*Hardware*)

Perangkat komputer berfungsi untuk mendukung proses penyimpanan serta akses data dalam sistem basis data. Contoh: Server, *hardisk*, RAM, CPU.

2. Sistem Operasi (*Operating System*)

Sistem operasi seperti Windows maupun Linux menyediakan lingkungan untuk menjalankan DBMS dengan dukungan pemanfaatan sumber daya seperti CPU, memori, serta pengaturan izin akses. Selain itu, sistem operasi juga mengatur *background services* seperti eksekusi *query*, pencadangan (*backup*), dan proses lainnya agar berjalan lancar.

3. Basis Data (*Database*)

Suatu sistem yang berfungsi menyimpan data secara terstruktur sehingga memudahkan proses akses, kelola, serta pembaruan. Basis data mencakup data, tabel, relasi, kunci, indeks, hingga skema.

4. Sistem Pengelola Basis Data (*Database Management System/DBMS*)

Perangkat lunak (*software*) digunakan untuk membuat, mengelola, serta mengakses basis data. Contoh: *MySQL*, *Oracle*, *Microsoft SQL Server*, *MongoDB*, *PostgreSQL*.

5. Pengguna (*User*)

Pengguna adalah pihak atau individu yang berinteraksi dengan sistem basis data untuk melakukan berbagai aktivitas seperti *input* data, mencari informasi, menyusun laporan, atau mengelola sistem.

Basis data (*database*) digunakan untuk menyimpan dan mengatur data agar mudah diakses dan dikelola. Basis data diperlukan untuk menyimpan data pembelian, persediaan, produksi, dan penjualan secara terpusat sehingga data lebih aman dan terstruktur. Basis data bermanfaat untuk menyimpan data operasional Toko Kopi Sidikalang agar tidak terjadi kehilangan data seperti pada pencatatan manual.

2.4.2 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah model data konseptual yang menampilkan struktur dan hubungan antar data dalam suatu sistem untuk membantu perancangan serta pendefinisian batasan pada basis data [15]. ERD menunjukkan entitas apa yang disimpan sistem, atribut apa yang terhubung pada entitas, dan hubungan antar entitasnya. Tujuannya agar struktur data rapi, konsisten, dan bisa diterjemahkan menjadi skema basis data relasional [8]. ERD juga berfungsi sebagai alat bantu dalam perancangan basis data yang menggambarkan bagaimana cara kerja *database* yang akan dibuat [16].

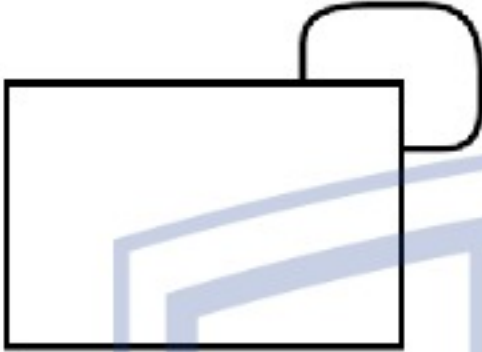
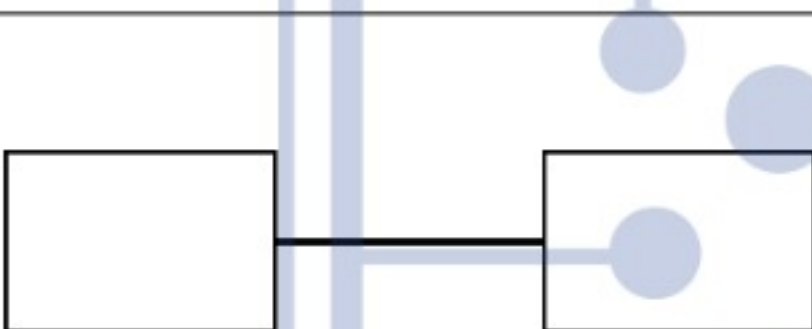
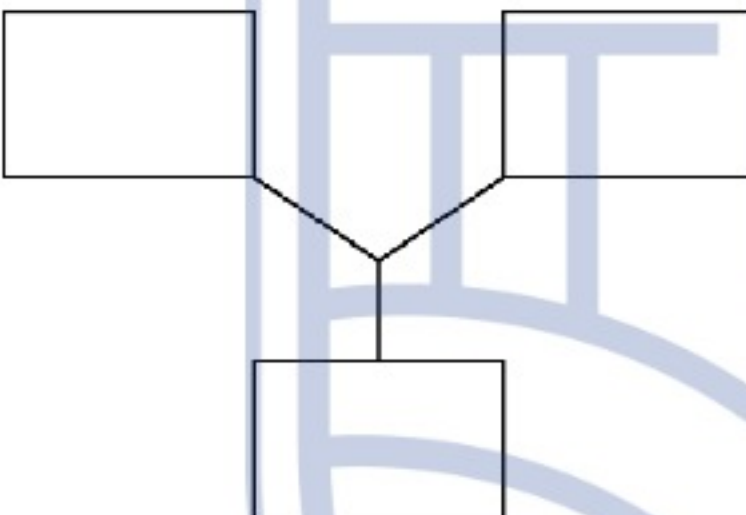
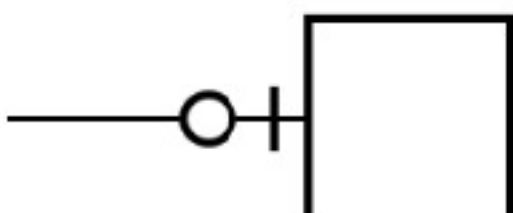
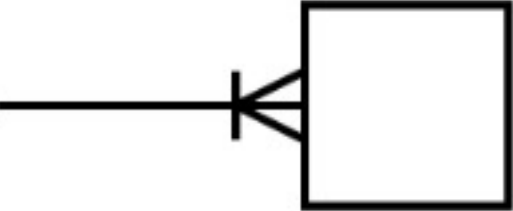
Dalam perancangan basis data relasional, penentuan identitas unik untuk setiap entitas merupakan tahapan yang penting. Terdapat dua jenis kunci utama yang digunakan dalam pemodelan data tingkat lanjut, di antaranya [17]:

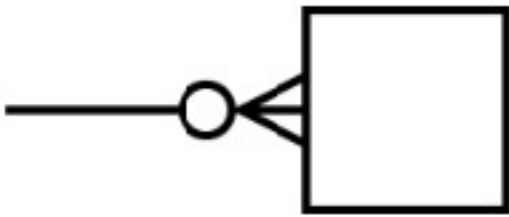
1. *Identifier (Enterprise Key)*: Satu atau lebih atribut yang secara unik mengidentifikasi setiap instansi dari suatu tipe entitas dalam lingkup organisasi. Karakteristik utama dari kunci ini adalah nilainya yang unik di seluruh relasi basis data perusahaan, bukan hanya terbatas pada satu tabel saja. Kriteria ini menjadikan *Primary Key* tersebut berperan sebagai pengenalan objek yang diakui secara luas dalam organisasi yang disebut sebagai *object identifier*. Misalnya, KodeBarang, KodePelanggan atau NomorPesanan yang memiliki makna bisnis bagi pengguna.
2. *Surrogate Key*: Nomor seri atau kunci unik lain yang ditetapkan secara otomatis oleh sistem (seperti fitur *Identify* atau *Auto-increment*) dan tidak memiliki makna bisnis bagi pengguna akhir untuk sebuah relasi. *Surrogate Key* bertujuan untuk menyederhanakan struktur kunci dan meningkatkan efisiensi sistem. Menurut Hoffer dkk. (2019), penggunaan *surrogate key* disarankan apabila memenuhi kondisi berikut:
 - a. Keberadaan *Composite Primary Key*: Menghindari penggunaan kunci gabungan yang rumit sebagai relasi ke tabel lain.
 - b. *Natural Primary Key* yang Tidak Efisien: Kunci alami yang digunakan dalam organisasi (seperti nama atau alamat) dianggap tidak efisien sebagai pengidentifikasi karena terlalu panjang, sehingga membebani kinerja perangkat lunak basis data saat digunakan sebagai *foreign key* yang merujuk ke tabel lain.
 - c. Stabilitas Kunci: Kunci alami faktanya tidak dapat dijamin unik atau stabil untuk digunakan kembali atau diulang secara berkala (misalnya ketika terjadi duplikasi data atau perubahan format kode), sementara *surrogate key* bersifat permanen dan tidak akan berubah.

Setiap kali *surrogate key* diimplementasikan, kunci alami (*enterprise key*) selalu disimpan sebagai data non-kunci (*non-key attribute*) dalam relasi yang sama karena kunci alami memiliki makna organisasi yang penting dan digunakan sebagai identitas utama dalam proses pencarian data oleh pengguna.

Tabel 2.3 Notasi ERD

Notasi	Keterangan
	<i>Strong Entity</i> : Entitas yang keberadaannya tidak bergantung pada entitas lain dan memiliki pengenalan unik (<i>identifier</i>) sendiri.
	<i>Weak Entity</i> : Jenis entitas yang kunci utamanya bergantung pada kunci utama entitas lain dan tidak memiliki kunci utama yang lengkap tanpa menyertakan kunci dari entitas pemiliknya.
	<i>Associative Entity</i> : Jenis entitas yang mengasosiasikan instansi dari satu atau lebih jenis entitas dan berisi atribut yang khusus untuk hubungan antara instansi entitas tersebut.
ENTITY NAME <u>Identifier</u> <u>Partial Identifier</u> Optional [Derived] {Multivalued} Composite(, ,)	<i>Attributes</i>: Sebuah properti atau karakteristik bernama dari suatu entitas yang menarik bagi organisasi. Terdapat beberapa jenis <i>attributes</i>, di antaranya: <i>Identifier</i>: Kunci kandidat yang telah dipilih sebagai karakteristik unik yang mengidentifikasi untuk jenis entitas. <i>Optional</i>: Sebuah atribut yang mungkin tidak memiliki nilai untuk setiap instansi entitas. <i>Derived</i>: Sebuah atribut yang nilainya dapat dihitung dari nilai atribut terkait.

	d. <i>Multivalued</i>: Sebuah atribut yang dapat memiliki lebih dari satu nilai untuk setiap instansi entitas. e. <i>Composite</i>: Sebuah atribut yang memiliki bagian komponen yang bermakna.
	<i>Unary Relationship</i>: Sebuah hubungan antara instansi dari satu tipe entitas atau disebut hubungan rekursif.
	<i>Binary Relationship</i>: Sebuah hubungan antara instansi dari dua jenis entitas. Ini adalah tipe hubungan yang paling umum ditemui dalam pemodelan data.
	<i>Ternary Relationship</i>: Hubungan simultan di antara instansi dari tiga jenis entitas.
	<i>Mandatory One</i>: Setiap instansi pada entitas A harus berhubungan dengan tepat satu instansi pada entitas B.
	<i>Optional One</i>: Instansi pada entitas A dapat berhubungan dengan maksimal satu instansi pada entitas B, atau tidak sama sekali.
	<i>Mandatory Many</i>: Setiap instansi pada entitas A harus berhubungan dengan setidaknya satu atau lebih instansi pada entitas B.

	<i>Optional Many</i>: Instansi pada entitas A dapat berhubungan dengan nol, satu, atau banyak instansi pada entitas B.
---	---

ERD memiliki 3 komponen utama, yaitu:

1. Entitas: suatu objek yang menjadi fokus dalam suatu basis data. Entitas dapat berupa orang, benda, tempat, dan kondisi terkait data yang dibutuhkan. Setiap entitas memiliki atribut yang menjelaskan karakteristiknya, di mana seluruh atribut tersebut dicatat dan disimpan dalam basis data. Entitas biasanya digambarkan menggunakan simbol persegi panjang [15].
2. Atribut: karakteristik yang dimiliki setiap entitas yang berfungsi untuk menjelaskan informasi penting yang tersimpan dalam basis data. Setiap entitas wajib memiliki *primary key* sebagai kunci unik suatu entitas dan atribut deskriptif [15]. Satu entitas juga dapat memiliki *foreign key* sebagai kolom yang merujuk pada *primary key* entitas lain untuk membentuk relasi [8]. Atribut biasanya ditempatkan dalam tabel entitas atau terpisah sesuai kebutuhan desain. Atribut biasanya digambarkan menggunakan simbol elips [15].
3. Relasi: hubungan antara dua atau lebih entitas yang saling berhubungan dalam suatu sistem. Relasi biasanya digambarkan menggunakan simbol dengan bentuk belah ketupat. Relasi memiliki beberapa jenis, yaitu [16]:
 - a. *One to One* (1:1): satu entitas hanya berhubungan dengan satu entitas lainnya.
 - b. *One to Many* (1:N): satu entitas dapat berhubungan dengan beberapa entitas lain.
 - c. *Many to Many* (M:N): beberapa entitas dapat saling berhubungan dengan beberapa entitas lainnya secara bersamaan.

Dalam pemodelan ERD, entitas biasanya dilambangkan dengan persegi panjang, atribut dengan bentuk elips, dan relasi dengan bentuk belah ketupat sebagai bagian dari notasi standar dalam perancangan basis data. Pemodelan ini digunakan untuk memetakan entitas, atribut, dan relasi antar data. ERD digunakan untuk membantu Toko Kopi Sidikalang dalam merancang struktur tabel dan relasi yang jelas seperti data pemasok, barang, pelanggan, transaksi, dan bagian lainnya serta hubungan antar data sebelum diimplementasikan pada basis data fisik.

2.4.3 Normalisasi

Normalisasi basis data adalah proses terstruktur untuk mengorganisir data dalam basis data relasional. Tujuan utamanya adalah untuk menghilangkan redundansi (pengulangan data) dan meminimalkan anomali (ketidaknormalan) yang mungkin terjadi selama pengolahan data, seperti saat melakukan operasi *UPDATE*, *INSERT*, atau *DELETE*. Dengan memperbaiki struktur tabel dan hubungan antar tabel, normalisasi bertujuan untuk menghasilkan desain basis data yang optimal dan efisien [18].

Tujuan utama normalisasi adalah untuk menghasilkan desain basis data yang baik, efisien, dan terhindar dari duplikasi data. Proses ini dilakukan dengan memecah tabel-tabel besar menjadi tabel-tabel kecil tanpa kehilangan hubungan antar data yang ada. Selain itu, normalisasi juga bertujuan untuk meningkatkan konsistensi data, mempercepat proses pencarian, serta menjaga keakuratan data di seluruh sistem [18].

Manfaat dari normalisasi basis data antara lain [18]:

1. Mengurangi Redundansi Data
Proses normalisasi membantu mengurangi pengulangan data, yang berdampak pada penghematan ruang penyimpanan dan peningkatan konsistensi data. Hal ini juga mempermudah perubahan data dan meminimalkan kesalahan atau inkonsistensi yang timbul akibat data berulang.
2. Meningkatkan Konsistensi Data
Normalisasi memastikan bahwa data yang tersimpan bersifat konsisten dan dapat diandalkan, yang penting untuk menghasilkan data yang akurat dalam aplikasi dan sistem.
3. Mempercepat Pencarian Data
Dengan mengurangi jumlah tabel dan kolom yang perlu dicari, normalisasi dapat meningkatkan kecepatan pencarian data, sehingga meningkatkan kinerja basis data dan mempercepat waktu akses.
4. Memfasilitasi Perubahan
Basis data yang dinormalisasi menjadi lebih fleksibel dan mempermudah perubahan pada struktur data tanpa memengaruhi keseluruhan basis data.
5. Menghindari Duplikasi Data
Normalisasi menjamin bahwa data hanya disimpan di satu lokasi, mencegah duplikasi, meningkatkan integritas data, dan memudahkan pemeliharaan basis data.

Terdapat beberapa kriteria umum yang harus dipenuhi dalam proses normalisasi [18]:

1. Setiap atribut dalam tabel hanya memiliki satu nilai

Setiap atribut harus bernilai tunggal (*single-value*) atau atomik, yang berarti tidak boleh mengandung kumpulan nilai atau nilai terpisah yang harus dikelompokkan bersama. Ini memastikan data dapat dikelola dengan mudah dan akurat.

2. Setiap tabel harus memiliki sebuah *primary key*

Setiap tabel wajib memiliki *primary key* yang unik dan tidak berubah-ubah untuk mengidentifikasi setiap baris dalam tabel secara khas.

3. Tidak ada atribut yang tergantung pada atribut non-*key*

Struktur tabel harus diatur sedemikian rupa sehingga tidak ada atribut non-kunci yang bergantung pada atribut non-kunci lainnya. Ini dicapai dengan menormalisasi tabel hingga mencapai minimal bentuk normal.

4. Tidak ada ketergantungan siklik

Tidak boleh ada tabel yang saling tergantung satu sama lain, sehingga urutan tabel yang tepat saat mengakses data dapat ditentukan.

Normalisasi dilakukan melalui serangkaian tingkatan atau level, yaitu:

1. **First Normal Form (1NF)**

Pada level ini, setiap kolom dalam tabel hanya boleh memiliki nilai atomik, yang berarti tidak dapat dibagi lagi. Selain itu, setiap baris dalam tabel harus unik, dan tabel tidak boleh memiliki kolom yang nilainya dapat berubah-ubah. Intinya, 1NF melarang adanya pengelompokan data atau duplikasi data.

Contoh normalisasi *database* pada *invoice Pine Valley Furniture Company* [17]:

Tabel 2.4 *Unnormalized Form (UNF)*

Order ID	Order Date	Customer ID	Customer Name	Customer Address	Product ID	Product Description	Product Finish	Product Standard Price	Order Qty
1006	10/24/2018	2	Value Furniture	Plano, TX	7	Dining Table	Natural Ash	800.00	2
					5	Writer's Desk	Cherry	325.00	2
					4	Entertainment Center	Natural Maple	650.00	1
1007	10/25/2018	6	Furniture Gallery	Boulder, CO	11	4-Dr Dresser	Oak	500.00	4

					4	Entertainment Center	Natural Maple	650.00	3
--	--	--	--	--	---	-------------------------	------------------	--------	---

Data yang disajikan di atas merupakan data mentah atau belum dinormalisasi yang selanjutnya akan diproses pada tahap Normalisasi Tingkat Pertama (1NF). Sebuah tabel dianggap memenuhi kriteria 1NF jika setiap atributnya bersifat atomik, artinya tidak mengandung data yang berulang atau pengelompokan nilai. Dengan kata lain, setiap baris data harus memuat nilai tunggal. Oleh karena itu, tabel yang masih dalam bentuk awal tersebut perlu disesuaikan strukturnya menjadi bentuk 1NF seperti yang ditunjukkan tabel berikut:

Tabel 2.5 *First Normal Form (1NF)*

Order ID	Order Date	Customer ID	Customer Name	Customer Address	Product ID	Product Description	Product Finish	Product Standard Price	Order Qty
1006	10/24/2018	2	Value Furniture	Plano, TX	7	Dining Table	Natural Ash	800.00	2
1006	10/24/2018	2	Value Furniture	Plano, TX	5	Writer's Desk	Cherry	325.00	2
1006	10/24/2018	2	Value Furniture	Plano, TX	4	Entertainment Center	Natural Maple	650.00	1
1007	10/25/2018	6	Furniture Gallery	Boulder, CO	11	4-Dr Dresser	Oak	500.00	4
1007	10/25/2018	6	Furniture Gallery	Boulder, CO	4	Entertainment Center	Natural Maple	650.00	3

Dapat diamati bahwa data telah bersih dari pengelompokan (*grouping*) dan duplikasi, menunjukkan bahwa data tersebut sudah berada dalam 1NF (*First Normal Form*). Langkah selanjutnya yang akan dilakukan adalah proses normalisasi 2NF (*Second Normal Form*).

2. *Second Normal Form (2NF)*

Syarat untuk mencapai 2NF adalah tabel harus sudah berada dalam 1NF dan tidak boleh ada *partial functional dependency* terhadap *primary key*. *Functional dependency* berarti

setiap atribut non-kunci harus bergantung secara fungsional pada *primary key*. Oleh karena itu, tabel perlu dipecah berdasarkan *primary key* untuk memastikan bahwa setiap kolom bergantung sepenuhnya pada kunci utama. Sehingga, pada data yang sudah dilakukan normalisasi 1NF perlu diubah lagi ke bentuk normalisasi 2NF dimana data akan dibagi menjadi tabel barang dan tabel transaksi untuk menghindari adanya *functional dependency* menjadi sebagai berikut:

a. Tabel *Order Line*

Tabel 2.6 *Second Normal Form (2NF) Tabel Order Line*

<i>OrderID</i>	<i>ProductID</i>	<i>OrderQty</i>
1006	7	2
1006	5	2
1006	4	1
1007	11	4
1007	4	3

b. Tabel *Product*

Tabel 2.7 *Second Normal Form (2NF) Tabel Product*

<i>ProductID</i>	<i>ProductDescription</i>	<i>ProductFinish</i>	<i>ProductStandardPrice</i>
7	Dining Table	Natural Ash	800.00
5	Writer's Desk	Cherry	325.00
4	Entertainment Center	Natural Maple	650.00
11	4-Dr Dresser	Oak	500.00
4	Entertainment Center	Natural Maple	650.00

c. Tabel *Customer Order*

Tabel 2.8 *Second Normal Form (2NF) Tabel Customer Order*

<i>OrderID</i>	<i>OrderDate</i>	<i>CustomerID</i>	<i>CustomerName</i>	<i>CustomerAddress</i>
1006	10/24/2018	2	Value Furniture	Plano, TX

1006	10/24/2018	2	Value Furniture	Plano, TX
1006	10/24/2018	2	Value Furniture	Plano, TX
1007	10/25/2018	6	Furniture Gallery	Boulder, CO
1007	10/25/2018	6	Furniture Gallery	Boulder, CO

Karena data di atas telah bebas dari masalah *functional dependency* parsial, maka tahapan selanjutnya yang akan dilakukan adalah proses normalisasi 3NF (*Third Normal Form*).

3. *Third Normal Form (3NF)*

Syaratnya, tabel harus sudah berada dalam 2NF dan tidak diperbolehkan adanya *transitive dependency* parsial. *Transitive dependency* terjadi ketika sebuah kolom non-kunci bergantung pada kolom non-kunci lainnya. Pada tahap ini, jika ada atribut yang tidak bergantung pada *primary key* melainkan bergantung pada *field* lain, atribut-atribut tersebut perlu dipisah ke tabel baru. Contohnya pada atribut *CustomerID*, kolom tersebut tidak bergantung langsung pada *primary key OrderID* melainkan bergantung pada kolom *CustomerID* itu sendiri. Sehingga setelah dinormalisasi ke 3NF akan menghasilkan tabel berikut:

a. Tabel *Order Line*

Tabel 2.9 *Third Normal Form (3NF)* Tabel *Order Line*

<i>OrderID</i>	<i>ProductID</i>	<i>OrderQty</i>
1006	7	2
1006	5	2
1006	4	1
1007	11	4
1007	4	3

b. Tabel *Product*

Tabel 2.10 *Third Normal Form (3NF)* Tabel *Product*

<i>ProductID</i>	<i>ProductDescription</i>	<i>ProductFinish</i>	<i>ProductStandardPrice</i>
7	Dining Table	Natural Ash	800.00

5	Writer's Desk	Cherry	325.00
4	Entertainment Center	Natural Maple	650.00
11	4-Dr Dresser	Oak	500.00
4	Entertainment Center	Natural Maple	650.00

c. Tabel *Order*

Tabel 2.11 *Third Normal Form (3NF)* Tabel *Order*

<i>OrderID</i>	<i>OrderDate</i>	<i>CustomerID</i>
1006	10/24/2018	2
1006	10/24/2018	2
1006	10/24/2018	2
1007	10/25/2018	6
1007	10/25/2018	6

d. Tabel *Customer*

Tabel 2.12 *Third Normal Form (3NF)* Tabel *Customer*

<i>CustomerID</i>	<i>CustomerName</i>	<i>CustomerAddress</i>
2	Value Furniture	Plano, TX
2	Value Furniture	Plano, TX
2	Value Furniture	Plano, TX
6	Furniture Gallery	Boulder, CO
6	Furniture Gallery	Boulder, CO

Tahap normalisasi hingga mencapai 3NF (*Third Normal Form*) menghasilkan pemecahan tabel baru untuk menghilangkan anomali, seperti terbentuknya tabel *Customer* secara mandiri. Tahap akhir dalam menentukan kunci relasional pada tabel hasil normalisasi adalah dengan menambahkan *surrogate Key* berupa atribut ID.

a. Tabel *Order Line*

Tabel 2.13 *Third Normal Form (3NF)* Tabel *Order Line* dengan *Surrogate Key*

<i>OID</i>	<i>OrderID</i>	<i>ProductID</i>	<i>OrderQty</i>
1	1006	7	2
2	1006	5	2
3	1006	4	1
4	1007	11	4
5	1007	4	3

b. Tabel *Product*

Tabel 2.14 *Third Normal Form (3NF)* Tabel *Product* dengan *Surrogate Key*

<i>OID</i>	<i>ProductID</i>	<i>ProductDescription</i>	<i>ProductFinish</i>	<i>ProductStandardPrice</i>
1	7	Dining Table	Natural Ash	800.00
2	5	Writer's Desk	Cherry	325.00
3	4	Entertainment Center	Natural Maple	650.00
4	11	4-Dr Dresser	Oak	500.00
5	4	Entertainment Center	Natural Maple	650.00

c. Tabel *Order*

Tabel 2.15 *Third Normal Form (3NF)* Tabel *Order* dengan *Surrogate Key*

<i>OID</i>	<i>OrderID</i>	<i>OrderDate</i>	<i>CustomerID</i>
1	1006	10/24/2018	2
2	1006	10/24/2018	2
3	1006	10/24/2018	2
4	1007	10/25/2018	6
5	1007	10/25/2018	6

d. Tabel *Customer*

Tabel 2.16 *Third Normal Form (3NF)* Tabel *Customer* dengan *Surrogate Key*

<i>OID</i>	<i>CustomerID</i>	<i>CustomerName</i>	<i>CustomerAddress</i>
1	2	Value Furniture	Plano, TX
2	2	Value Furniture	Plano, TX
3	2	Value Furniture	Plano, TX
4	6	Furniture Gallery	Boulder, CO
5	6	Furniture Gallery	Boulder, CO

Penambahan kunci buatan ini dilakukan pada tahap desain fisik karena sifatnya yang stabil, unik, dan bertipe data numerik (*integer*). Penggunaan tipe data angka sangat efisien untuk mengoptimalkan proses pengindeksan (*indexing*), pencarian data, serta penggabungan (*join*) antar tabel. Dalam implementasinya, atribut ID yang biasa disebut *ObjectID* ini ditetapkan sebagai *Primary Key* dengan fitur *Auto-Increment* dan akan ditarik sebagai *Foreign Key* untuk menghubungkan antar entitas dalam *Entity Relationship Diagram* (ERD). Penambahan *surrogate key* setelah normalisasi ini sangat membantu dalam memastikan bahwa struktur tabel tersusun secara efisien, terhindar dari duplikasi, dan terbebas dari masalah ketidaknormalan (anomali) ketika sistem menjalankan operasi data seperti *INSERT*, *UPDATE*, dan *DELETE*. Tujuannya agar basis data dapat digunakan secara stabil dalam jangka panjang.

2.4.4 Kamus Data

Kamus data dapat didefinisikan sebagai sebuah katalog yang berisi fakta-fakta terperinci mengenai data dan kebutuhan informasi dari suatu sistem. Kamus data merupakan kumpulan daftar elemen data yang alurnya terdapat dalam sebuah sistem perangkat lunak. Tujuan utamanya adalah agar setiap masukan (*input*) dan keluaran (*output*) memiliki standar penulisan yang baku, sehingga dapat dipahami secara umum dan konsisten. Kamus data menggunakan notasi atau simbol-simbol khusus untuk mendefinisikan struktur data. Simbol-simbol tersebut dapat dirincikan sebagai berikut [19]:

Tabel 2.17 Notasi Kamus Data

No.	Simbol	Keterangan
1.	=	Mengandung arti "terdiri atas" atau "disusun dari".
2.	+	Berfungsi sebagai penghubung "Dan".

3.	[]	Mengandung arti “baik” dan “atau”
4.	{ } ⁿ	Merepresentasikan pengulangan atau iterasi sebanyak -n.
5.	()	Menandakan bahwa suatu data atau elemen bersifat opsional (opsional).
6.	*...*	Digunakan untuk memberikan batas komentar.

Berdasarkan notasi kamus data di atas, berikut contoh rancangan kamus data[20]:

Tabel 2.18 Rancangan Kamus Data

No.	Nama Aliran Data	Struktur Kamus Data
1.	Data <i>supplier</i>	= kode <i>supplier</i> + nama <i>supplier</i> + no telepon + alamat
2.	Data pelanggan	= kode pelanggan + nama pelanggan + no telepon + alamat
3.	Data barang Satuan besar = [Kotak Pack] Satuan sedang = [Papan] Satuan kecil = [Butir Pcs Bungkus Botol Tube]	= kode barang + nama barang + satuan besar + konversi ke sedang + satuan sedang + konversi ke kecil + satuan kecil + { jumlah + satuan + harga ecer + harga grosir + tanggal kadaluarsa }
4.	Data pembelian Satuan besar = [Kotak Pack] Satuan sedang = [Papan] Satuan kecil = [Butir Pcs Bungkus Botol Tube]	= no pembelian + no pesanan pembelian + tanggal + kode <i>supplier</i> + nama <i>supplier</i> + { kode barang + nama barang + jumlah + satuan + harga satuan + <i>subtotal</i> + kadaluarsa } + total
5.	Data penjualan Satuan besar = [Kotak Pack] Satuan sedang = [Papan]	= no faktur + tanggal + kode pelanggan + nama pelanggan + { kode barang + nama barang + jumlah + satuan + harga + <i>subtotal</i> + kadaluarsa } + total

	Satuan kecil = [Butir Pcs Bungkus Botol Tube]	
6.	Daftar <i>supplier</i>	= { kode <i>supplier</i> + nama <i>supplier</i> + telepon }
7.	Daftar pelanggan	= { kode pelanggan + nama pelanggan + alamat + telepon }
8.	Daftar Barang	= {kode barang + nama barang + satuan besar + konversi sedang + satuan sedang + konversi kecil + satuan kecil }
9.	Laporan pembelian	= { no pembelian + tanggal + nama <i>supplier</i> + kode barang + nama barang + jumlah + satuan }
10.	Laporan penjualan	= { no penjualan + tanggal + nama pelanggan + total }
11.	Laporan Persediaan	= {kode barang + nama barang + jumlah + satuan + kadaluarsa }

Kamus data menjelaskan struktur dan arti setiap data dalam sistem. Dalam penelitian ini, kamus data membantu Toko Kopi Sidikalang agar sistem memiliki definisi data yang jelas dan tidak terjadi kesalahpahaman dalam penggunaannya, misalnya data barang, pemasok, atau penjualan.

2.5 Teknologi yang Digunakan

2.5.1 *SQL Server*

SQL Server adalah sistem manajemen basis data relasional yang digunakan untuk mengelola, menyimpan, dan mengolah data secara terstruktur menggunakan bahasa SQL (*Structured Query Language*). Sistem ini dikembangkan untuk menangani kebutuhan skala kecil hingga besar, baik untuk aplikasi lokal, jaringan internal, maupun berbasis *cloud*. *SQL Server* mendukung proses transaksi, analisis data, serta integrasi dengan berbagai platform pengembangan. Beberapa komponen penting dalam *SQL Server* antara lain [21]:

1. *Database Engine*: inti dari *SQL Server* yang berfungsi menyimpan, memproses, dan mengamankan data. *Database Engine* menangani semua perintah SQL seperti *query*, transaksi, dan kontrol akses pengguna.

2. *SQL Operating System (SQLOS)*: komponen ini bertugas mengatur penggunaan sumber daya fisik seperti memori, CPU, dan I/O. SQLOS memastikan pembagian sumber daya secara efisien di antara berbagai proses yang berjalan di dalam server.
3. *Query Processor*: komponen ini mengubah perintah SQL menjadi rencana eksekusi yang efisien. Proses ini mencakup analisis sintaks, optimasi *query*, dan pemilihan metode terbaik untuk pengambilan data.
4. *Storage Engine*: bertanggung jawab atas pengelolaan struktur fisik data di *disk*, termasuk indeks, tabel, dan halaman data. *Storage Engine* juga menangani mekanisme transaksi dan pemulihan (*recovery*).
5. *Transaction Log*: menyimpan catatan setiap perubahan data untuk menjamin integritas transaksi. Dengan log ini, sistem dapat melakukan *rollback* atau *recovery* jika terjadi kesalahan atau kegagalan sistem.
6. *TempDB*: basis data sementara yang digunakan untuk menyimpan hasil sementara dari *query*, data sementara pengguna, dan proses internal. Karena sering diakses, *TempDB* sebaiknya ditempatkan di penyimpanan dengan kecepatan tinggi.
7. *SQL Server Agent*: Komponen otomatisasi yang digunakan untuk menjalankan tugas-tugas terjadwal seperti *backup*, pemeliharaan *database*, dan *monitoring*.

SQL Server merupakan sistem manajemen basis data relasional yang digunakan untuk menyimpan dan mengolah data transaksi dan operasional secara aman, terstruktur dan mudah diakses. Sistem ini sesuai dengan kebutuhan sistem di Toko Kopi Sidikalang untuk menangani data pembelian, persediaan, produksi, dan penjualan.

2.5.2 Visual Studio

Visual Studio adalah lingkungan pengembangan terpadu yang digunakan untuk menulis, menguji, dan mengelola aplikasi perangkat lunak. *Integrated Development Environment (IDE)* ini mendukung banyak bahasa pemrograman seperti C#, C++, *Visual Basic*, *Python*, *JavaScript*, dan *HTML*. *Visual Studio* dirancang untuk mempermudah pengembang dalam membuat aplikasi berbasis desktop, web, maupun *mobile* dalam satu platform terpadu. Komponen utama dalam Visual Studio antara lain [22]:

1. *Code Editor*: bagian inti tempat pengembang menulis kode program dengan fitur pewarnaan sintaks, *auto-complete*, dan *IntelliSense* untuk membantu penulisan kode yang cepat dan akurat.

2. *Solution Explorer*: berfungsi untuk menampilkan struktur proyek dalam bentuk hierarki *folder* dan *file*. Pengembang dapat dengan mudah menavigasi antar *file* dan mengelola referensi proyek.
3. *Debugger*: digunakan untuk mendeteksi dan memperbaiki kesalahan pada program. Fitur ini memungkinkan pengguna menjalankan kode langkah demi langkah untuk memeriksa variabel, kondisi, dan logika program.
4. *Designer Tools*: komponen visual yang digunakan untuk merancang antarmuka pengguna, baik untuk aplikasi desktop, web, maupun *mobile*, melalui metode *drag and drop*.
5. *Output Window & Terminal*: menampilkan pesan sistem, hasil kompilasi, dan *error log* selama proses pembangunan proyek berlangsung.
6. *Extensions dan Integrations*: Visual Studio mendukung ribuan ekstensi yang dapat menambahkan fitur tambahan seperti dukungan bahasa baru, alat tes otomatis, hingga manajemen basis data.

Visual Studio mendukung pembuatan aplikasi berbasis *Graphical User Interface* (GUI) dan menyediakan lingkungan pengembangan yang lengkap dengan fitur antarmuka dan pengujian. Fitur tersebut memudahkan proses pembuatan aplikasi sehingga sistem lebih mudah digunakan oleh pemilik Toko Kopi Sidikalang.

2.6 Sistem Informasi Pembelian

Sistem informasi pembelian merupakan rangkaian aktivitas yang dirancang untuk memastikan kegiatan pengadaan barang atau jasa yang dilakukan memenuhi kebutuhan perusahaan, baik untuk digunakan dalam proses produksi maupun untuk dijual kembali. Proses pembelian ini melibatkan interaksi yang terstruktur antara sumber daya manusia, perangkat kerja, metode, serta pengendalian yang dirancang untuk menjalankan berbagai fungsi utama. Fungsi utama proses pembelian adalah menangani pekerjaan rutin yang dilakukan oleh bagian pembelian dan penerimaan, membantu pihak yang bertanggung jawab dalam membuat keputusan terkait pembelian dan penerimaan barang, serta mendukung penyusunan berbagai laporan untuk kebutuhan internal maupun eksternal [23]. Dengan adanya proses pembelian yang baik, organisasi dapat memenuhi kebutuhan barang dengan cepat, efektif, dan efisien [24].

Pembelian tidak hanya berfokus pada pengadaan barang atau jasa, tetapi juga memiliki tujuan strategis, seperti menjaga kualitas dan nilai barang yang diperoleh, mengendalikan perputaran modal dalam persediaan, serta memelihara kelancaran arus

barang masuk dan keluar demi mempertahankan daya saing perusahaan. Kegiatan ini meliputi pemrosesan permintaan pembelian, pencarian dan evaluasi vendor, penyusunan penawaran, pemeriksaan barang, hingga pengawasan penyimpanan dan penggunaannya. Proses ini diperkuat melalui sistem terintegrasi yang mencakup pencatatan *master data*, permintaan pembelian, seleksi vendor, pemantauan pesanan, penerimaan barang, verifikasi faktur oleh bagian keuangan, dan pelaporan pembukuan [25]. Sistem informasi pembelian yang baik diperlukan untuk mengumpulkan data dan mengolahnya menjadi laporan yang jelas sehingga dapat membantu proses pengambilan keputusan [26].

Sistem informasi pembelian membantu proses pemesanan dan penerimaan barang dari pemasok. Sistem informasi pembelian digunakan untuk memperbaiki proses pembelian pada Toko Kopi Sidikalang yang sebelumnya tidak terencana dan kurang terdokumentasi serta berperan penting dalam membantu mengelola ketersediaan bahan baku untuk mendukung proses produksi.

2.7 Sistem Informasi Persediaan

Sistem informasi persediaan adalah sistem terpadu berbasis teknologi yang dibuat untuk mendukung operasional, manajemen, analisis, dan pengambilan keputusan dalam rantai pasok. Melalui sistem ini, perusahaan dapat memantau ketersediaan dan pergerakan stok secara *real-time*, serta memfasilitasi koordinasi dan kolaborasi di antara semua pihak yang terlibat dalam rantai pasok seperti pemasok, pusat manufaktur, dan distributor dengan baik. Dengan penerapan sistem ini, perusahaan dapat menekan biaya produksi, meningkatkan efisiensi operasional, meningkatkan kualitas produk, dan mempercepat proses distribusi [27].

Selain itu, sistem informasi persediaan juga dapat menghasilkan informasi stok yang akurat dengan menggunakan metode yang tepat dalam menentukan nilai barang yang tersimpan dan barang yang terjual. Pemilihan metode ini akan memengaruhi perhitungan harga pokok penjualan, nilai persediaan akhir, serta laporan keuangan secara keseluruhan. Terdapat 2 metode persediaan, di antaranya [28]:

1. Metode Pencatatan Persediaan

Terdapat 2 metode yang digunakan dalam pencatatan persediaan, yaitu [28]:

a. Metode Pencatatan Permanen (Perpetual)

Metode di mana sistem mencatat setiap transaksi persediaan secara langsung dan berkelanjutan dalam kartu persediaan. Sistem ini memungkinkan perusahaan mengetahui jumlah persediaan dan harga pokok penjualan secara *real-time* karena

setiap perubahan stok langsung memperbarui saldo persediaan dan harga pokok penjualan. Metode pencatatan ini juga dilengkapi buku pembantu persediaan untuk setiap jenis barang, sehingga detail kuantitas dan nilai perolehan dapat dipantau kapan saja. Jika ditemukan selisih dengan stok fisik, maka dilakukan penyesuaian. Metode ini dinilai lebih akurat dan efektif untuk pengelolaan persediaan karena memberikan informasi yang cepat dan tepat dalam proses pengendalian.

b. Metode Pencatatan Fisik (Periodik)

Metode di mana sistem pencatatan persediaan hanya diperbarui pada akhir periode akuntansi. Dalam metode ini, perusahaan tidak mencatat perubahan persediaan setiap kali terjadi transaksi, sehingga selama periode berjalan tidak ada informasi saldo persediaan yang selalu diperbarui. Pembelian dicatat dalam akun khusus seperti *purchases*, dan penentuan persediaan akhir dilakukan melalui perhitungan fisik (*stock opname*) di gudang. Setelah jumlah fisik diketahui, harga pokok penjualan dihitung dengan rumus persediaan awal ditambah pembelian, kemudian dikurangi persediaan akhir. Sistem ini sederhana namun kurang akurat untuk pemantauan harian karena informasi persediaan hanya tersedia setelah stok fisik dihitung ulang.

2. Metode Penilaian Persediaan

Terdapat 3 metode yang digunakan dalam penilaian persediaan, yaitu [29]:

a. Metode FIFO (*First In First Out*)

Metode yang memiliki prinsip bahwa barang yang pertama masuk adalah barang yang pertama keluar. Sistem persediaan menganggap bahwa barang yang lebih lama akan keluar terlebih dahulu, sementara barang yang lebih baru akan tersisa sebagai persediaan akhir. Metode ini cenderung mencerminkan nilai persediaan yang lebih mendekati harga pasar terbaru dan sesuai untuk kondisi harga yang stabil atau meningkat.

b. Metode LIFO (*Last In First Out*)

Metode yang memiliki prinsip bahwa barang yang terakhir masuk ke dalam persediaan merupakan barang yang pertama kali dikeluarkan. Oleh sebab itu, persediaan akhir terdiri dari barang yang relatif lebih lama berada di gudang. Metode ini umum digunakan ketika perusahaan ingin menyesuaikan biaya penjualan dengan harga perolehan yang paling baru, sehingga pada kondisi harga meningkat, metode ini menghasilkan harga pokok penjualan yang lebih tinggi dan laba yang lebih rendah.

c. Metode Rata-Rata (*Average*)

Metode yang menentukan nilai persediaan berdasarkan biaya rata-rata dari seluruh barang yang tersedia dalam periode tertentu. Setiap terjadi penambahan persediaan, biaya rata-rata dihitung ulang dengan mempertimbangkan jumlah unit dan total biaya yang baru. Metode ini menghasilkan nilai persediaan dan harga pokok penjualan yang lebih stabil, sehingga cocok digunakan pada kondisi harga barang yang tidak menentu.

Sistem informasi persediaan berfungsi mengelola ketersediaan stok melalui pencatatan masuk dan keluar. Sistem informasi persediaan diperlukan untuk mengatasi masalah ketidakakuratan data stok bahan baku dan bubuk kopi di Toko Kopi Sidikalang.

2.8 Sistem Informasi Produksi

Sistem informasi produksi adalah sistem yang melakukan perhitungan terhadap biaya produksi, proses produksi, dan hasil produksi. Sistem ini membantu perusahaan menghitung biaya jasa produksi yang harus dikenakan kepada pelanggan. Karena setiap pesanan memiliki karakteristik tersendiri, pengusaha harus memperhitungkan biaya bahan baku, biaya operasional, dan keuntungan yang akan diperoleh [30].

Saat perusahaan menerima pesanan, sistem ini membantu menentukan bahan baku dan alat kerja yang akan digunakan. Dari persediaan yang ada, pengguna dapat menentukan bahan apa saja yang dibutuhkan untuk menyelesaikan pesanan. Selain itu, sistem ini juga membantu menentukan biaya jasa pengerjaan, sehingga perusahaan dapat menghitung total biaya secara cepat dan akurat yang harus dikeluarkan oleh pemesan [30].

Tujuan dari Sistem Informasi Produksi adalah [30]:

1. Memenuhi kebutuhan perusahaan: Sistem ini digunakan untuk memastikan produk hasil produksi tersedia agar kegiatan terkait dapat berjalan lancar. Dengan demikian, barang produksi dapat dibuat sesuai dengan pesanan, bahkan yang bersifat *custom* atau sesuai keinginan konsumen.
2. Menghitung modal: Sistem produksi dapat membantu perhitungan modal dengan mengurutkan komponen yang digunakan untuk menghasilkan produk.
3. Membuat proses produksi berjalan dengan teratur: Fungsi terpenting dari sistem ini adalah memastikan bahwa proses produksi dapat berjalan dengan lancar dan teratur.

Sistem informasi produksi membantu proses pencatatan bahan baku, proses produksi, dan hasil produksi. Sistem informasi produksi digunakan agar proses pengolahan produksi pada Toko Kopi Sidikalang dari bahan baku biji kopi menjadi bubuk kopi dapat tercatat dengan baik dan terjadwal.

2.9 Sistem Informasi Penjualan

Sistem informasi penjualan adalah serangkaian prosedur yang dirancang untuk menjalankan, merekam, menghitung, menghasilkan dokumen, dan menyediakan informasi terkait penjualan. Sistem ini berfungsi mengelola seluruh aspek penjualan, mulai dari pemesanan hingga transaksi, dengan tujuan mendukung berbagai operasi penjualan dan menyediakan data yang diperlukan oleh berbagai divisi. Peran utamanya adalah mengolah data dan transaksi yang terjadi untuk mencapai tujuan dan target organisasi [31].

Sistem Informasi Penjualan membawa beragam manfaat yang sangat berarti bagi kelancaran operasi bisnis kita, yang meliputi [31]:

1. Menyusun rencana penjualan: Sistem ini memberikan bantuan yang berarti dalam merancang strategi penjualan yang efektif untuk produk atau layanan tertentu. Dengan dukungan dari sistem ini, kita dapat mengembangkan rencana penjualan yang terperinci dan melakukan evaluasi terhadap kinerja serta hasil strategi menjadi lebih akurat.
2. Menyajikan laporan yang mendalam: Sistem ini memungkinkan kita untuk menghasilkan laporan yang menyajikan informasi penting tentang penjualan secara menyeluruh, termasuk data penjualan, status stok barang, pendapatan, pengeluaran, dan lainnya. Melalui laporan-laporan ini, kita dapat melakukan pemantauan dan evaluasi kinerja bisnis secara menyeluruh dengan lebih efektif.
3. Mengelola keuangan secara efektif: Sistem ini memungkinkan kita untuk dengan mudah menghitung dan melacak pemasukan serta pengeluaran harian. Dengan demikian, hal ini membantu dalam manajemen keuangan yang lebih efisien dan memungkinkan kita untuk membuat keputusan yang lebih tepat berdasarkan data yang lebih akurat.
4. Manajemen stok yang lebih teratur: Sistem ini memberikan dukungan dalam mengelola stok bahan baku atau produk dengan lebih efisien, sehingga kita dapat menentukan waktu yang tepat untuk melakukan *reorder* bahan baku, dan mengelola stok barang tanpa mengalami kesulitan.
5. Analisis penjualan yang mendalam: Sistem ini memudahkan dalam melakukan analisis terhadap hasil penjualan dengan menyediakan data yang terperinci. Kita dapat mengevaluasi performa penjualan dari berbagai aspek, seperti kinerja karyawan, ketersediaan stok, pembayaran, dan lainnya. Analisis yang mendalam ini membantu kita dalam mengambil keputusan yang lebih baik untuk meningkatkan kinerja bisnis secara keseluruhan.
6. Meningkatkan keuntungan secara drastis: Dengan bantuan sistem ini, kita dapat meningkatkan keuntungan dengan menyajikan laporan penjualan yang akurat dan

melakukan analisis data penjualan secara mendalam. Dengan pemahaman yang lebih baik tentang kinerja penjualan dan kondisi toko, kita dapat mengambil tindakan yang tepat untuk meningkatkan keuntungan dan merancang strategi pemasaran yang lebih efektif.

Sistem informasi penjualan mencatat transaksi penjualan dan menyusun laporan penjualan. Sistem informasi penjualan diperlukan untuk memperbaiki pencatatan transaksi dan untuk mencatat transaksi penjualan tunai dan kredit, sehingga mempermudah Toko Kopi Sidikalang dalam melacak dan memperbarui status piutang yang akan ditagih.

2.10 Kopi

Kopi merupakan minuman yang memiliki kandungan antioksidan khas berupa kafein. Kandungan kafein pada kopi dapat berbeda karena dapat ditingkatkan atau dikurangi. Perbedaan kandungan kafein pada biji kopi dapat disebabkan pengaruh intensitas cahaya matahari yang diterima oleh tanaman kopi. Selain itu, proses pengolahan seperti fermentasi juga dapat memberikan perubahan pada cita rasa kopi. Fermentasi dengan perlakuan yang berbeda dapat mengubah senyawa-senyawa kopi, termasuk kandungan antioksidan dan senyawa-senyawa lainnya [32].

Kandungan kafein dan antioksidan yang dimiliki kopi tidak hanya memengaruhi cita rasa, tetapi juga memberikan manfaat bagi tubuh. Kopi memberikan manfaat dalam meningkatkan fungsi kognitif, terutama kemampuan memori. Senyawa bioaktif di dalamnya membantu proses kerja otak sehingga konsumsi kopi dalam jumlah yang sesuai dapat memperbaiki daya ingat dan meningkatkan konsentrasi. Penelitian juga menunjukkan bahwa individu yang terbiasa mengonsumsi kopi memiliki kemampuan memori yang lebih baik dibandingkan mereka yang tidak mengonsumsinya [33].

Terdapat empat jenis kopi utama yang secara komersial dibudidayakan, yaitu [34]:

1. Arabika

Arabika adalah jenis kopi yang tumbuh optimal pada dataran yang lebih tinggi dan termasuk kopi dengan harga jual tertinggi di pasaran. Arabika memiliki rasa dan aromanya lebih halus dan kompleks dibandingkan robusta, sehingga sering digunakan untuk seduhan berkualitas premium.

2. Robusta

Robusta merupakan jenis kopi dengan produktivitas tinggi dan paling dominan dibudidayakan. Kopi ini sering digunakan untuk campuran espresso dan pembuatan kopi

instan. Robusta mendominasi produksi karena memiliki rasa yang kuat dan pahit, serta kandungan kafein yang lebih tinggi dibandingkan arabika.

3. *Excelsa*

Excelsa merupakan jenis kopi dengan ukuran buah yang lebih besar daripada arabika namun lebih kecil dari liberika. Jenis ini dikenal tahan kering dan relatif lebih tahan hama. *Excelsa* memiliki rasa yang unik karena aroma buah dan bunga yang khas, serta didukung oleh biji berukuran besar dan kandungan kafein yang seimbang.

4. Liberika

Liberika atau dikenal juga sebagai *Kapeng Barako* merupakan jenis kopi yang memiliki rasa dan aroma yang kuat serta tajam. Buah dan bijinya berukuran lebih besar, dengan aroma khas yang berbeda dari jenis kopi lain. Jenis ini memiliki kontribusi produksi paling kecil dibanding tiga jenis lainnya.

Pembahasan tentang kopi memberikan penjelasan mengenai produk utama yang dikelola oleh toko. Penjelasan mengenai jenis, karakteristik, dan pengolahan kopi diperlukan agar sistem yang dibangun menyesuaikan dengan proses bisnis yang sedang berjalan.

Berdasarkan pembahasan di atas, dapat disimpulkan bahwa pengembangan sistem informasi untuk Toko Kopi Sidikalang membutuhkan sistem yang terintegrasi agar proses pembelian, persediaan, produksi, dan penjualan dapat dikelola dengan baik. Dalam konsep sistem informasi menunjukkan bahwa data operasional Toko Kopi Sidikalang harus diolah melalui tahapan *input*, proses, dan *output* sehingga menghasilkan informasi yang berguna bagi kegiatan operasional dan mendukung proses pengambilan keputusan. Metodologi yang digunakan adalah *System Development Life Cycle* (SDLC) karena metodologi tersebut memberikan tahapan pengembangan yang jelas dan terstruktur sesuai dengan kebutuhan sistem Toko Kopi Sidikalang. Untuk analisis dan perancangan sistem, Toko Kopi Sidikalang menggunakan *fishbone*, *flow of document*, *data flow diagram*, serta perancangan basis data melalui ERD, normalisasi, dan kamus data untuk membantu dalam memahami permasalahan dan alur kerja yang ada di toko. Selain itu, pembahasan mengenai kopi dan jenis-jenis kopi memberikan gambaran mengenai objek penelitian, sehingga sistem informasi yang dirancang tidak hanya mendukung aspek teknis, tetapi juga sesuai dengan kebutuhan usaha Toko Kopi Sidikalang dalam pengelolaan data dan informasi secara lebih efektif dan teratur.