# CHAPTER 2
# LITERATURE REVIEW AND THEORETICAL FOUNDATION

## 2.1 Literature Review

The literature review was conducted using keywords related to the issue of overstock in various companies operating in the industrial sector. The literature search was conducted using search engines, namely Google Scholar and ScienceDirect.

Overstock is an inventory management dysfunction that often arises simultaneously in company operations; this problem not only occurs independently but also creates a negative cycle in company operations. Product overstock can tie up working capital and warehouse space, which can trigger stockouts in other product lines due to suboptimal resource allocation; therefore, these two issues must be analyzed holistically to find a comprehensive solution.

Based on a review of the literature, previous studies have extensively examined these two issues, though their subjects of study have been manufacturing companies operating in various industrial sectors. A variety of research methodologies were applied, resulting in diverse solutions tailored to the specific contexts of each company's unique challenges. Based on the problem analysis outlined in subsection 1.2, the identified issue at CV Andita Swarga Mahakarya is the presence of excess inventory.

The literature review focuses on tracing and analyzing various previous studies that are closely correlated with the identified problem. The steps for conducting the literature review using Google Scholar are shown in Table 2.1 below.

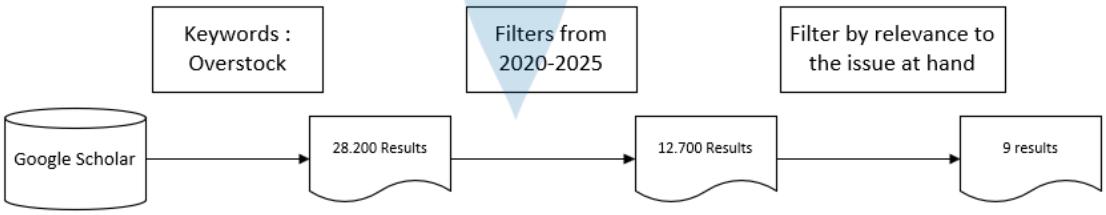



Figure 2. 1 Google Scholar Literature Review

### 2.1.1 Research on The Problem of Excess Inventory Caused by Layout Issues

A warehouse is a facility used to store items such as raw materials, semi-finished products, and finished goods that will be used for production or distribution. Warehouses play a significant role in determining the success or failure of a company's supply chain.

In addition to serving as a storage facility, warehouses have other functions, such as a distribution function that acts as a consolidation and transit point for the receipt and shipment of goods, there is a production support function used to store raw materials, work-in-progress, and finished goods that support the smooth operation of the production process, and there is also a collateral function, where the warehouse serves as a storage facility for goods that can be used as credit collateral (Pitoy et al. 2020; Astra et al. 2025). However, there are several common issues that frequently arise in warehouse management, namely.

a. Layout and Storage Issues

According to Pitoy et al. (2020), at the Paris Superstore warehouse in Kotamobagu, a suboptimal layout has led to disorderly and uneven stockpiling of goods, which hinders employees' ability to move around and increases the risk of damage to goods. Furthermore, problems with layout and storage can result in a shortage of storage space.

b. Environmental and Infrastructure Issues

A study conducted by Pitoy et al. (2020) found that rat infestations and inadequate lighting in warehouses can disrupt warehouse operations. These suboptimal environmental conditions not only affect the quality of stored goods but also lead to a decline in employee productivity.

c. Fast-Moving and Slow-Moving Products

The problem of overstocked parts piling up in the warehouse area occurs when the number of stored parts exceeds actual needs and the established storage time limit. At PT XYZ's finished goods warehouse, overstock is one of the causes of reduced storage capacity because parts that should be distributed immediately instead pile up in the warehouse area. This situation arises due to suboptimal warehouse layout, inaccurate inventory records, and a FIFO system that is not operating according to standards. The accumulation of these parts leads to wasted storage space, hinders the process of locating and retrieving items, and increases material handling time. Additionally, parts that remain in the warehouse for too long risk requiring re-inspection before being shipped to customers, thereby increasing operational activities and reducing overall warehouse efficiency Astra et al. (2025).

d. FIFO (First In-First Out) System

The FIFO (First In-First Out) system, which is not yet operating according to standards, results in items that entered the warehouse first not always being the first to leave.

According to a journal Astra et al. (2025), this situation is evident from the fact that production parts from August 1, 2023, were still found in the warehouse in May 2025. This indicates that the flow of parts storage and retrieval is not yet organized, leading to accumulation, the risk of overstock, and the need for re-inspection before goods are shipped to customers Astra et al. (2025).

### 2.1.2 Issues with Information Systems

Problems with warehouse information systems are often a major cause of inefficiencies in inventory recording and control within the warehouse. These issues are frequently caused by the fact that inventory is still recorded manually using ledgers or paper, resulting in data inaccuracies. Human errors, such as writing mistakes, can lead to invalid inventory information, and data security and reliability are low because manual systems lack data backups, and documents are prone to damage. Additionally, another issue that arises is limited access and slow information retrieval; searching for data in a manual system takes a considerable amount of time because employees must open and review each manual record one by one. (Hartono & Si, 2011;Bainal Budur et al. 2023)

A study conducted at SMK Negeri 1 Magelang revealed that requests for goods must be made by visiting the warehouse in person; the administrator then checks inventory availability using a manual ledger, which slows down the process of reporting to management Tugiyono et al. (2025). This situation is also experienced by PT. LEN Industri (Persero) in Bandung, where material checks still rely on physical archives that are not properly organized, making the processes of shipping, receiving, and storing materials in the warehouse highly ineffective and inefficient Hartono & Si, (2011).

Cost and time inefficiencies are also a problem; although this may seem simple, manual record-keeping actually incurs higher operational costs over the long term, such as expenses for purchasing books, pens, and archive storage. Research conducted at SMK Negeri 1 Magelang revealed that the cost of purchasing paper for recording data is quite high when accumulated, compounded by the wasted work time spent solely on writing and searching for data Tugiyono et al. (2025). At PT. LEN Industri (Persero), all procedures—from receiving, issuing, requesting, and ordering materials—are still manual, resulting in lengthy and convoluted workflows, which in turn reduces employee productivity.

### 2.1.3 Issues Regarding Order Timing

The main challenge in inventory management is the uncertainty involved in determining the reorder point. Setting the reorder point is a critical decision because it directly impacts

stock availability and inventory cost efficiency. If a company places an order too early before inventory reaches the required level, it will affect the amount of inventory in the warehouse, leading to overstock. According to research Nirmal et al. (2022). The primary cause of overstock lies in a conventional ordering system that relies on managers' experience, intuition, and estimates without conducting accurate demand data analysis. Consequently, ordering decisions are often made prematurely due to concerns about stock shortages. If demand is lower than estimated, incoming goods are not immediately used, leading to inventory buildup. To address this issue, the researchers recommend using the Economic Order Quantity (EOQ) method to optimize order quantities and implement more structured order scheduling. Meanwhile, according to Gilang et al. (2022), the causes of overstock are primarily Attributed to weak inventory monitoring systems and a lack of coordination among various departments within a company. Inaccurate inventory data leads to repetitive or hasty purchases, without taking into account that warehouse stock levels are still sufficient. This problem is caused by calculation errors and a lack of an information system to support decision-making. The proposed solution is to implement the Min-Max method and a more real-time monitoring system, so that orders are placed only when stock reaches the minimum threshold and inventory levels are restored to that minimum.

Table 2.1 Literature Review of Previous Studies

| No | Pustaka | Object | Issues | Methods and Tools | Solution | Results |
|---|---|---|---|---|---|---|
| **1** | Permadi et al. (2024) | PT Gunung Sejahtera Puti Pesona | Low transport unit availability (<95%) due to a high rate of Category B1 breakdowns; discrepancies between ordered and received goods; continued reliance on ineffective manual inventory control, leading to stockouts and overstock; and a lack of visualization for spare parts that would facilitate the identification process | Min-Max dan Reorder Point (ROP), Microsoft Excel 2019+ Visual Basic for Applications (VBA) | Developing a digital inventory management system for the company, including a monitoring dashboard, a comprehensive spare parts grid with images, minimum and maximum stock levels, and reorder points (ROP); creating forms for incoming and outgoing goods transactions to digitize data and implement automated inventory control | Inventory is now better controlled and monitored, enabling cost savings of up to Rp. 30,431,555 by identifying frequent instances of overstock. Order accuracy has improved thanks to visual representations and detailed part numbers, and processes are faster because data is accessed in real time without having to wait for updates from the old system. |
| **2** | Afrilia, (2020) | PT. Aneka Usaha | Overstocking often occurs because orders are still based on estimates and the accumulation of returned goods. Out-of-stock situations still frequently occur due to uncertainty in demand and a lack of proper planning. There is currently no scientific method for calculating safety stock and reorder points. High | Economic Order Quantity (EOQ), dan Microsoft Excel Macro VBA | Applying the Economic Order Quantity to determine the optimal order quantity, safety stock, reorder point, and total inventory cost (TIC) | The frequency of orders has decreased to 5 times per year, down from 12 times. Cost savings have increased significantly, from Rp. 4.05 million (2017), to Rp. 4.83 million (2018), and to Rp. 6.87 million (2019). Inventory management has also become more optimal and controlled. |

| No | Pustaka | Object | Issues | Methods and Tools | Solution | Results |
|---|---|---|---|---|---|---|
| | | | inventory costs result from inefficiencies. | | | |
| **4** | Nirmal et al. (2022) | Inventory management company | Overstocking and untimely orders occurred because the ordering system was still manual, relying on estimates and intuition, which resulted in order quantities and timing that did not match actual needs. | Metode Economi Order Quantity (EOQ), ROP, Monte Carlo Simulation | Determine the optimal order quantity and create a more efficient order schedule based on demand data | Inventory management is more under control, storage costs have decreased, and ordering is more efficient |
| **5** | Gilang et al. (2022) | The company's warehouse inventory system | Overstock occurs due to a lack of inventory monitoring, poor coordination, and insufficient inventory information, resulting in orders being placed too quickly or repeatedly | Metode, Min-Max stock Tools, Information System | Determine minimum and maximum stock levels and monitor inventory in real time so that orders can be placed as needed | Inventory is better managed, excess inventory is reduced, and the ordering process is more accurate and timely |
| **6** | Astra et al. (2025) | PT. XYZ | Fast-moving and slow-moving products are mixed together. The FIFO (First In, First Out) system is not yet operating according to standards. | Shared storage, FIFO (First In-First Out), dan ARC, ARD. | The first solution utilizes WSM software to rearrange part placement based on available empty space and standard storage dimensions. Additionally, a gravity flow rack system was implemented to separate the inbound and outbound pathways for goods, thereby speeding up | This has had a significant impact. In terms of costs, there has been a reduction in storage costs due to overstock of Rp 4,389,000, as well as a reduction in material handling costs of Rp 2,748,000,000 per year. In terms of delivery, loss time in the material handling |

| No | Pustaka | Object | Issues | Methods and Tools | Solution | Results |
|---|---|---|---|---|---|---|
| | | | | | the part transfer process. Adjustments were also made by establishing standard sizes for part storage in the warehouse, allowing parts to be moved more flexibly according to space requirements. A barcode system was also implemented through WSM to monitor part locations in real time and prevent overstocking. | process has been reduced by 61%, or approximately 2 minutes and 12 seconds. In terms of safety, the risk of workplace accidents has decreased due to adjustments to shelf heights to accommodate warehouse operators. |
| **7** | Pitoy et al. (2020) | Paris Superstore Warehouse | Layout and storage issues have caused delays in warehouse operations. | FIFO (First In-First Out), JIT (Just in Time), snowbale sampling, stock Opname | Adjust the warehouse layout to make warehousing activities more effective and efficient. Researchers recommend that stacks of goods be arranged more neatly, evenly, and in accordance with stacking capacity, so that employees have more room to move and the risk of damage to goods can be reduced. In addition, the company is also advised to implement the FEFO (First Expired-First Out) method, as many of the stored products are food and food ingredients with limited shelf lives. With the FEFO method, items with the nearest | The Paris Superstore warehouse in Kotamobagu has already implemented the FIFO system, whereby goods that enter first are removed first. Warehouse operations at this facility are divided into three main activities: receiving, handling, and shipping. |

| No | Pustaka | Object | Issues | Methods and Tools | Solution | Results |
|---|---|---|---|---|---|---|
| | | | | | expiration dates will be removed first. | |
| 8 | Tugiyono et al. (2025) | SMK Negeri 1 Magelang | Record-keeping is still done manually, resulting in imperfect data accuracy, time-consuming record-keeping, and very low data security (with no backups or access controls) | Waterfall, analysis PIECES, PHP, MySQL, black box testing, skala likert | Web-based inventory management system | Improvements to the inventory system, making it easier for administrators to accurately and efficiently record and report inventory levels; 90% success rate |
| 9 | Hartono & Si, (2011) | PT. LEN Industri (Persero) Bandung | Material tracking is still done manually, which slows down material verification and makes all procedures ineffective and inefficient, resulting in frequent errors in report generation | Waterfall, Flowmap, context diagrams, DFD, ERD, PHP, MySQL | An intranet-based warehouse management information system for material storage and inventory | Facilitates employees in data retrieval and computerized report generation, and supports procedures for receiving, issuing, requesting, and ordering materials |
| 10 | Bainal Budur et al. (2023) | UD Surya Nugraha | Manual inventory tracking meant that incoming and outgoing stock wasn't recorded, procurement estimates were consistently inaccurate, stockouts were frequent, and the owner's workload was extremely heavy | Waterfall, Observation, UML, PHP, MySQL, black box | Web-based warehouse management application | Helps manage the recording of incoming and outgoing goods, generates accurate data, stores customer and supplier data, and achieves 95% UAT compliance with user requirements |

## 2.2 Theoretical Framework

This final project was conducted using several theories taught in Industrial Engineering courses.

### 2.2.1 Theoretical Basis of Information System Design

Information system design theory encompasses the concepts, models, and methods used to analyze, design, and build a system capable of collecting, processing, strong, and presenting data to support organizational decision making. In this study, the design of the VBA-based inventory information system for CV Andita Swarga Mahakarya draws on several established theories in the fields of system analysis, database design, human-computer interaction, and software quality assurance.

a. System Development Life Cycle (SDLC)

The system development life cycle (SDLC) is a structured framework that describes the phases involved in developing an information system, from initial planning through analysis, design, implementation, and maintenance Kendall & Kendall., (2008). According to Kendall & Kendall (2008), the SDLC generally consists of several interrelated phases (1) identifying the problem, opportunities and objectives; (2) determining human and information requirements; (3) analyzing system needs; (4) designing the recommended system; (5) developing and documenting the software; (6) testing and maintaining the system; and (7) implementing and evaluating the system. Each phase produces specific deliverables-such as requirement specifications, process models, and data models-that serve as the input for the following phase, so that the resulting system can always be traced back to the actual needs identified at the outset. Kendall & Kendall (2008) further emphasize that although the SDLC can be carried out in a strictly sequential (waterfall) manner or in a more it creative fashion, the underlying logic of moving from a general understanding of the problem toward an increasingly specific and technical design remains the same regardless of the variation adopted.

In this study, the SDLC is adopted as the overarching development method because the scope of the information system-used by only three stakeholder roles within a single small enterprise-does not require complex iterative development cycles, but does require a clear and traceable sequence from requirement identification to system rollout. The SDLC approach is realized in three main stages, as elaborated in Chapter 5 and Chapter 6: (1) System Planning, in which the existing business process of receiving and issuing goods at CV Andita Swarga Mahakarya is mapped and the activities of each stakeholder are identified (2) Design, which further consist of process modelling (context diagram

and DFD Level 0-2) and data modelling (attribute identification and ERD); and (3) Implementation, which covers installation, stakeholder training, live operational trials, and User Acceptance Testing. By following this staged structure, every design decision made in Chapter 5-from the entities included in the database to the layout of each VBA from-can be traced directly back to the business process and stakeholder needs identified during the planning stage.

b. Data Flow Diagram (DFD)

A Data Flow Diagram (DFD) is a structured analysis tool used to describe, in a purely logical manner, how data moves into, through, and out of a system, without describing the physical or technical implementation of that movement Chris Gane & Thrish Sarson (1979). Introduced notation consisting of four basic symbols (10 the process, which represents an activity that transform incoming data into outgoing data; (2) the external entity, which represents a person, organization, or other system lying outside the boundary of the system being modelled but exchanging data with it; (3) the *data store*, which represents a location where data is held for later use; and (4) the *data flow*, which represents the movement of data between processes, entities, and stores. Alongside this notation, Chris Gane & Thrish Sarson (1979) proposed a top-down decomposition technique in which a system is first represented as a single process in a context diagram, and is then progressively broken down—level by level—into more detailed sub-processes, so that the internal mechanism of the system becomes increasingly explicit while the total inputs and outputs remain consistent across levels, a principle commonly referred to as balancing.

This theory underlies the entire process modelling stage of this study. At the highest level, the context diagram (Figure 5.2) represents the inventory information system as a single process exchanging data with two categories of external entities—warehouse staff/cashier on one side and the owner on the other—consistent with Chris Gane & Thrish Sarson (1979) concept of a system boundary. This is then decomposed into the Level 0 DFD (Figure 5.2), which reveals the system as a single process interacting with three stakeholders (warehouse staff, cashier, and owner), followed by the Level 1 DFD (Figure 5.3), which breaks Process 0 down into six sub-processes—master data, unit and price input, incoming goods recording, cashier sales, inventory calculation and reporting, and reporting to the owner—each receiving its own inputs and producing its own outputs that remain traceable to the higher-level diagram. Finally, the Level 2 DFDs (Figures 5.4 and 5.5) further decompose Process 2 (unit/price input), Process 4 (cashier sales), and Process 5 (inventory calculation and reporting) into more granular steps,

allowing the internal logic of unit conversion, stock validation, and ending-inventory calculation to be traced directly to the VBA macro logic implemented in Chapter 5 and Chapter 6.

c. Entity Relationship Diagram (ERD)

An Entity Relationship Diagram (ERD) An Entity Relationship Diagram (ERD) is a conceptual data modelling technique used to represent the entities that exist within a system's domain, the attributes that describe each entity, and the relationships that connect entities to one another, based on the assumption that the real world can be represented as a collection of discrete objects and the associations between them Chen, (1976). Introduced a diagrammatic notation in which entities are represented as rectangles, attributes as ovals attached to their respective entities, and relationships as diamonds connecting two or more entities, with the cardinality of each relationship—one-to-one, one-to-many, or many-to-many—indicated along the connecting lines. Chen, (1976) Further distinguished between strong entities, which can exist independently and possess their own primary key, and weak entities, which depend on another entity for their identification; this distinction becomes particularly relevant when a system contains transactional data that only has meaning in relation to a parent record, such as the line items of a single transaction.

This theory forms the basis for the data modelling stage of this study. As shown in Chapter 5 (Sections 5.3.1–5.3.2), the inventory information system is built around seven main entities—User, Product, Stock Limit, Stock Entry Header, Stock Entry Detail, Sales Transaction, and Sales Detail—each identified with a distinct primary key and a defined set of attributes (Tables 5.2–5.8). Following Chen, (1976) distinction between master and transactional data, these entities are categorized into master entities (User, Product, and Stock Limit), which describe relatively static reference data, and transaction entities (Stock Entry Header/Detail and Sales Transaction/Detail), which record events occurring repeatedly over time and depend on the master entities for their meaning—for example, a Stock Entry Detail record only has meaning in relation to its parent Stock Entry Header and to the specific Product it references. The cardinality relationships between these entities, such as the one-to-many relationship between a single Stock Entry Header and its multiple Stock Entry Detail records, are depicted in the ERD presented in Figure 5.6 and directly determine how data is organized across the Excel worksheets that serve as the system's database.

d. Database Design

Database design theory concerns the systematic process of translating a conceptual data model into a structure that can store, retrieve, and maintain data efficiently, accurately, and without unnecessary duplication, and is typically carried out through three progressive phases: conceptual design, which produces an entity-relationship model independent of any specific technology; logical design, which translates that model into a set of normalized tables or relations; and physical design, which determines how those tables are implemented within a specific database management system. Connolly & Begg, (2015) Explain that a central concern throughout these phases is normalization, a technique for organizing attributes so that each piece of data is stored only once and functional dependencies between attributes are respected, thereby minimizing update anomalies and ensuring that a change to a single fact—such as a product's price—needs to be made in only one place. Connolly & Begg, (2015) also emphasize that every entity in a well-designed database must be uniquely identifiable through a primary key, and that relationships between entities are enforced through foreign keys that reference the primary key of a related entity.

In this study, this theory is applied in the specification of the Microsoft Excel worksheet-based database described in Chapter 5. Although Excel is not a conventional relational database management system, the same conceptual and logical design principles are applied: each entity identified in the ERD (Section 5.3.2) is translated into its own worksheet-based table with a defined primary key—such as Item ID for the Product entity and Entry ID for the Stock Entry Header entity—so that records can be uniquely referenced and retrieved through VBA lookup functions. The separation between the Stock Entry Header/Sales Transaction "header" tables and their corresponding "detail" tables (Sections 5.3.1–5.3.1) reflects the normalization principle described by Connolly & Begg, (2015), in which repeating groups of line-item data are moved into a separate table rather than being stored redundantly within a single transaction record. This structure directly supports the automatic ID generation, duplicate-checking, and stock-update mechanisms implemented in the item data form and goods-out form in Chapter 5.

e. User Interface Design and Usability

User interface design theory is concerned with how the visual layout, interactive controls, and feedback mechanisms of a system are structured so that users can operate it accurately, efficiently, and with a low likelihood of error Downey & Laskowski, (1996). Downey & Laskowski, (1996) Proposed a set of usability principles—later formalized as

heuristics—that include visibility of system status, so that users always know what the system is doing; match between the system and the real world, so that terminology and workflow reflect the user's own vocabulary and mental model; user control and freedom, so that mistakes can be undone or corrected; consistency and standards, so that similar actions are represented the same way throughout the system; error prevention, which Nielsen considers even more important than well-designed error messages because it stops mistakes before they occur; and recognition rather than recall, so that users are not required to memorize information across different parts of the interface. Downey & Laskowski, (1996) further advocates for "discount usability engineering," an approach that emphasizes practical, low-cost methods—such as heuristic evaluation and iterative prototyping—that can be applied even in small-scale development projects rather than requiring large usability laboratories.

This theory underlies the design of all four VBA forms described in Chapter 5. The principle of error prevention is realized through validation routines that check for duplicate Item IDs before a new product record is saved, and through confirmation message boxes that appear before a delete action is executed (Section 5.4.2). The principle of visibility of system status is realized through real-time updates of remaining stock, unit price, and total transaction value as data is entered into the goods-out and invoice forms (Section 5.4.3), as well as through the dashboard's automatic filtering of items approaching the safety stock threshold (Section 5.4.4). The principle of consistency and standards is reflected in the use of the same set of controls—Label, Text Box, Combo Box, and Command Button—across the login form, item data form, and goods-out form, so that users who have learned to operate one form can transfer that knowledge to the others. Finally, the inclusion of a search box and refresh button in the item data form (Section 5.4.2) reflects the principle of recognition rather than recall, allowing warehouse staff to locate existing records without needing to remember exact item names or scroll through the entire dataset.

f. Role-Based Access Control (RBAC)

Role-Based Access Control (RBAC) is a security model in which permissions to perform specific system functions are assigned to roles rather than to individual users, and users are in turn assigned to one or more roles, so that a user's access rights are determined entirely by the role or roles they occupy within the organization Sandhu ' et al. (1996). Sandhu ' et al. (1996) describe RBAC through a hierarchy of reference models—ranging from a flat model, in which each role is associated with a fixed set of permissions, to more advanced models that incorporate role hierarchies and separation-of-duty

constraints—but the underlying principle common to all of them is that administering access becomes significantly simpler because permissions only need to be defined once for each role, rather than repeatedly for every individual user. Sandhu ' et al. (1996) further argue that RBAC supports the security principle of least privilege, in which each role is granted only the minimum set of permissions necessary to perform its designated tasks, thereby reducing the potential impact of an error or a compromised account.

This theory is applied directly in the design of the login form and the user access rights matrix presented in Chapter 6 (Table 6.1). In this study, three roles are defined—Owner, Inventory Staff, and Cashier—each authenticated through the login form using a username, password, and role attribute stored in the User entity (Section 5.3.1.a). Following the principle of least privilege described by Sandhu ' et al. (1996), each role is granted access only to the forms relevant to its responsibilities: Inventory Staff can access the Goods Data Form and the Dashboard Form to record incoming goods and monitor stock levels, but cannot access the Goods Outgoing Data Form, Invoice Form, or Graph Form, which are reserved for the Cashier and the Owner respectively; the Owner, as the role with the broadest oversight responsibility, is granted access to all forms in the system. This role-based structure ensures that sensitive functions—such as viewing aggregated sales and revenue graphs—are restricted to the party accountable for strategic decision-making, while operational forms remain accessible to the staff who perform the corresponding daily tasks.

g. Visual Basic for Applications (VBA) Programming

Visual Basic for Applications (VBA) is an event-driven programming language embedded within Microsoft Office applications, including Excel, that allows a standard spreadsheet to be extended into a fully interactive, custom-built application through the use of macros, UserForms, and automated procedures that respond to specific user actions. explain that VBA development is centered around the Visual Basic Editor (VBE), within which a developer can create UserForms populated with standard controls—such as labels, text boxes, combo boxes, command buttons, check boxes, list boxes, frames, and multi-page containers—and attach event-handler procedures, such as a button's "Click" event, that execute a defined sequence of VBA code whenever that event occurs. Alexander and Walkenbach (2015) further note that VBA provides direct access to the Excel object model, including Workbook, Worksheet, and Range objects, which allows a developer to read from and write to specific cells programmatically, perform calculations, and enforce validation rules that would not be possible using standard spreadsheet formulas alone.

This theory constitutes the technical foundation of the entire information system built in this study. Each form described in Chapter 5—the login form, item data form, goods-out and invoice form, and dashboard form—is constructed as a VBA UserForm using the specific control types identified by Alexander and Walkenbach (2015): Label controls to display static text and instructions; Text Box controls to capture free-form input such as usernames, passwords, and quantities; Combo Box controls to constrain input to a predefined list of products, units, or payment methods; Command Button controls to trigger procedures such as saving, editing, deleting, or printing; Check Box controls to toggle optional behavior such as showing or hiding a password; Image controls to display product visuals; Frame controls to visually group related fields; List Box controls to display tabular summaries such as the shopping cart or low-stock list; and MultiPage controls to separate the dashboard and the goods-out/invoice form into distinct functional pages. The underlying VBA code attached to each control's event procedures performs the validation, automatic ID generation, real-time recalculation, and data-saving logic described throughout Chapter 5 and Chapter 6, effectively transforming the Excel workbook into a purpose-built inventory information system without requiring a dedicated external database platform.

h. Black-Box Testing and User Acceptance Testing (UAT)

Black-box testing is a software testing technique in which a system is evaluated solely on the basis of its inputs and the outputs it produces, without any reference to or knowledge of the internal code, algorithms, or logic used to generate those outputs, so that testing is conducted entirely from the perspective of what the system is expected to do rather than how it does it. distinguish this approach from white-box testing, which examines a program's internal structure and control paths, and argue that black-box testing is particularly well suited to validating whether a system satisfies its functional specification and meets the expectations of its intended users, since it mirrors the way an actual end user would interact with the system. further note that black-box techniques—such as testing representative input scenarios and verifying that outputs match expected results—are most effective when test cases are developed directly from real operational scenarios rather than from the system's internal design, since this increases the likelihood of uncovering discrepancies between what the system was intended to do and what it actually does once deployed. User Acceptance Testing (UAT), in this context, represents the application of black-box testing principles at the final stage of the development life cycle, in which representative end users execute realistic task

scenarios to confirm that the system is fit for operational use before it fully replaces an existing process.

This theory underlies the UAT described in Section 6.1.6 of this study. The three test scenarios conducted with warehouse staff, cashier, and owner were designed around real operational tasks—recording incoming goods, processing a customer sale and printing an invoice, and monitoring stock levels and sales graphs—rather than around the internal VBA code executing behind each form. Each scenario evaluated the system purely on the basis of whether the expected output was produced given a realistic input: for example, whether entering a quantity of incoming bales correctly updated the total stock display, whether entering a sales quantity correctly reduced the remaining stock and generated an accurate invoice total, and whether the dashboard correctly flagged products approaching the safety stock threshold. Because all three scenarios were completed successfully and confirmed by the respective stakeholders, the UAT results provide evidence that the VBA-based inventory information system meets its functional requirements and is ready to replace the company's previous manual recording process, as concluded in Sections 6.1.6 and 6.1.7.

### 2.2.3 Causes of Overstock

Overstock is a condition in which a company's inventory exceeds actual market demand (Zhang et al. 2025). This condition can lead to various negative impacts, such as increased storage costs, reduced operational efficiency, and decreased profits because goods are not sold immediately. Furthermore, the accumulation of excess inventory also has the potential to cause environmental impacts due to waste from unused products (Asdecker et al. 2022). The phenomenon of overstock is generally caused by several interrelated factors. One of the main causes of overstock is the absence of effective control policies. Many companies still rely on intuition to determine the timing and quantity of orders without the support of systematic data-driven analysis. Furthermore, the absence of product classification systems—such as ABC analysis or cycle counting—results in all items being treated equally, even though each product has different turnover rates and risks (Putri et al. 2024). Another critical factor is inaccuracies in demand forecasting. Many companies still use simple methods, such as moving averages and exponential smoothing, which are unable to adapt to changes in sales patterns and seasonal trends (Poongothai et al. 2024).

### 2.3 Standards and Code of Ethics

Standards and codes of ethics encompass both the ethical and technical aspects of the overall final project research solution. In this subsection, the author explains the rules, principles, and values that will guide and ensure the validity of the solution to be produced. The elaboration of the standards and code of ethics used demonstrates the author's commitment to the impact of the final project research. The following are the standards and code of ethics used.

#### 2.3.1 Standards for Recording Income and Expenses using VBA

According to the 2018 Financial Accounting Standards for Micro, Small, and Medium Entities (SAK EMKM), simple accrual-based accounting requires that the application be capable of recording revenue when it is earned and expenses when they are incurred, even if cash has not yet been received or paid. Input forms must include columns for the transaction date and due date, if applicable. The logic in the VBA code must classify transactions into the appropriate accounts based on revenue, expenses, rent expenses, payroll expenses, etc. The application's output must be presented fairly and clearly. The application must be able to maintain a log of data changes; every entry, edit, or deletion of data must be recorded. This can be implemented with VBA code that writes the log to a worksheet. The transaction ID generated must be automatically assigned by the system as a unique identifier for each incoming or outgoing transaction. The record-keeping system must also include password protection; the Excel file must be password-protected to open the file and require a separate password to modify the VBA code. This is intended to ensure that financial data or other data recorded in Excel remains confidential, as this data is sensitive and must be kept private by the user. The recording system must also support tax compliance requirements; the application must generate transaction summaries that facilitate tax reporting (VAT, Income Tax Article 4, Paragraph 2).

SNI ISO/IEC 25010:2011 Software Quality Standard. When developing a VBA-based record-keeping system, you can refer to this SNI. This standard governs software quality, including functionality and efficiency.

SNI ISO 9001:2015 Inventory Control Clause. The VBA automation system you design serves to implement the requirements of this SNI, specifically Clause 8.5.1 (Control of Service Provision): Requires the availability of information to monitor and measure inventory. Clause 7.5 (Documented Information): Requires accurate and traceable records of incoming and outgoing goods.

### 2.3.2 Code of Ethics for Recording Revenue and Expenses using VBA

The primary reference used is the Indonesian Accountants' Code of Ethics. This code of ethics contains five fundamental principles: integrity, objectivity, professional competence and due care, confidentiality, and professional conduct (Indonesian Institute of Accountants, 2021). The principle of integrity is applied by ensuring that all revenue and expenses are recorded in accordance with the actual transactions that occurred. Data must not be altered, deleted, or manipulated solely to present a more favourable financial picture. When using VBA, this principle can be applied by creating clear input formats, avoiding unnecessary data changes, and ensuring that every transaction includes traceable details. Another key principle in the Code of Ethics is confidentiality, which plays a vital role in recording income and expenses. A company's financial data may contain sensitive information, such as sales figures, operating expenses, cash inflows, cash outflows, and information about transacting parties. The Indonesian Accountants' Code of Ethics emphasizes that information obtained through professional or business relationships must not be disclosed to unauthorized parties without a valid reason (Indonesian Institute of Accountants, 2021). In addition to accounting ethics, the use of VBA must also adhere to the ethics of technology use. The ACM Code of Ethics and Professional Conduct emphasizes the principles of avoiding harm, being honest and trustworthy, respecting privacy, and maintaining data confidentiality in the use of computing technology (Association for Computing Machinery, 2018). These principles can be applied by ensuring that VBA macros are not created to hide data, delete transaction records without a trace, or generate misleading reports. Instead, VBA should be used to speed up record-keeping, reduce calculation errors, and facilitate the reconciliation of income and expenses.