

# similarity\_138.docx

*by* Mia Roberts

---

**Submission date:** 26-Jun-2026 03:12PM (UTC+0900)

**Submission ID:** 2989721366

**File name:** similarity\_138.docx (967.88K)

**Word count:** 2583

**Character count:** 14952

# Warehouse Inventory System With Min Max Stock And QR Code Method In One Of The Printing Companies

## *Abstract*

*The swift advancement of computer-based information systems has markedly enhanced the efficacy of many operational tasks within organizations, especially in inventory management, which necessitates precision and efficiency. A printing company continues to rely on a manual system for documenting incoming and outgoing goods, retrieving item information, and overseeing stock levels, leading to frequent delays, inaccuracies, and inefficient workflows. This study formulates a comprehensive warehouse inventory system that uses the Min-Max Stock approach as a regulatory framework to establish minimum and maximum inventory thresholds, thereby averting shortages and excess stock. The system incorporates QR Code technology to enhance item identification, streamline recording, and deliver detailed item information. The deployment of this system is anticipated to improve operational efficiency, minimize recording errors, expedite data retrieval operations, and facilitate more precise decision-making in stock management at the printing company.*

**Keywords :** *Inventory System, Min-Max Stock Method, QR Code, Warehouse Management, Stock Control*

## **INTRODUCTION**

The advancement of information technology has driven substantial transformations across numerous industrial sectors, particularly in inventory management, which demands precision, speed, and efficiency. Computerized systems have emerged as a principal solution for delivering rapid and precise information in commercial operations. Inventory is essential to ensuring a company's seamless operations, as it is directly linked to storage, distribution, and meeting production requirements. Inventory management is a crucial component of the supply chain. Warehouses serve as storage facilities that require systematic organization and management to enhance material handling operations (Avuduri et al., 2025; Balaji et al., 2024).

Nevertheless, numerous organizations, especially those in the printing industry, continue to rely on manual inventory management. The procedures for documenting incoming and outgoing items, monitoring inventory levels, and retrieving product information are performed manually and maintained in distinct computer files. This scenario results in inefficiencies in

data retrieval, service delays, and possible mismatches between actual inventory and recorded data. Moreover, data on items, including prices, descriptions, and photos, is not consolidated into a unified system, complicating employees' ability to find and verify information.

An integrated inventory system is required to enhance the effectiveness of inventory management. The Min-Max Stock approach serves as a regulatory tool to establish minimum and maximum inventory thresholds, hence averting shortages and surplus stock. The adoption of QR Code technology is perceived as a means to accelerate item recognition, streamline recording, and boost data accuracy. Implementing an inventory system that integrates the Min-Max Stock approach with QR Code technology enables organizations to enhance operational efficiency, reduce recording errors, accelerate information retrieval, and facilitate improved decision-making in inventory management.

### A. Information

The term "information" originates from the French word "information," which derives from

the Latin "informationem," signifying thought, idea, or outline. The term 'information' denotes the process of knowledge transmission. It comprises a compilation of data or facts that are transformed into a valuable resource for the recipient. Information is typically processed initially to facilitate comprehension by the recipient; in other words, it is presented in a format that is valuable or relevant (Ngan Ton et al., 2024).

#### B. Inventory System

An inventory system is used to enter inventory data into a database, minimizing input and output errors while facilitating the generation of reports tailored to individual requirements. The data is organized according to specific protocols, ensuring its readiness for use and proper maintenance within the database (Hernandoko & Widyo Laksono, 2023).

#### C. Min-Max Approach

The min-max stock method is a technique for managing safety stock levels that uses minimum and maximum inventory policies. This method entails regulating minimum and maximum inventory thresholds by structuring stock ordering strategies to avert shortages (stockouts) or surplus inventory (overstock) (Hermawan et al., 2025).

#### D. QR Code

The Quick Response Code, or QR Code, is a two-dimensional barcode developed by the Japanese firm Denso Wave in 1994. This barcode type was originally used to track car spare parts inventory and has since been adopted across many business sectors for marketing and promotional purposes. The QR Code was originally designed to enable rapid content translation (Dharmakirti et al., 2024). A key feature of QR Codes is their ability to store

information horizontally and vertically, enabling greater data storage than a one-dimensional barcode. Currently, QR Codes are widely used through applications like QR Code Readers and QR Code Generators, enabling users to easily generate QR Code content and retrieve the desired information by scanning it with their smartphone camera.

#### E. Microsoft Visual Basic .NET

Microsoft Visual Basic (VB) is a programming language that provides a visual Integrated Development Environment (IDE) for developing software applications on the Microsoft Windows operating system, utilizing the Component Object Model (COM) programming model. Certain scripting languages, such as Visual Basic for Applications (VBA) and Visual Basic Scripting Edition (VBScript), resemble Visual Basic but function distinctly. Developers can create apps with the components offered by Microsoft Visual Basic. VB programs can use the Windows API, but they require additional external function declarations (Mauidzoh et al., 2025).

#### F. Unified Modeling Language (UML)

UML was established by the Object Management Group in 1997 with version 1.0 and functions as a design instrument for object-oriented systems. The inception of UML is influenced by established notions, notably Object-Oriented (OO) modeling, which parallels systems with real-world situations characterized by objects. These objects are denoted by distinct symbols, rendering OO standards-compliant and autonomous [7]. The UML categories include the Use Case Diagram, which illustrates the behavior of the information system to be developed, facilitating the identification of available

functions and the permitted users of those functions (Lailiyah et al., 2025).

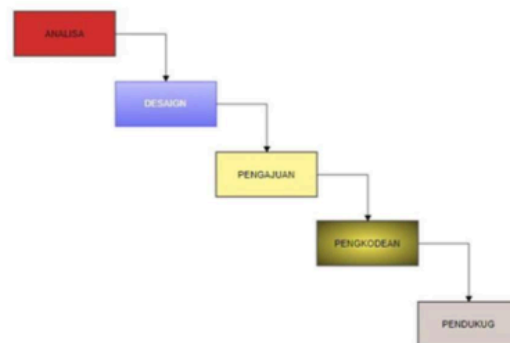
The Class Diagram delineates the system's architecture by specifying the classes that can be instantiated to construct the system. A class consists of attributes, which are its variables, and methods, which denote its functionality. The Activity Diagram illustrates the organized flow of data and operations within a system (Audina & Bakhtiar, 2021). The Sequence Diagram delineates the behavior of objects within a use case by illustrating their lifespans and the messages exchanged among them. To construct a sequence diagram, it is essential to identify the objects pertinent to a use case and the methods of the instantiated classes, with the quantity of sequence diagrams reflecting the established use cases and their message interactions (García-Chávez et al., 2022).

## METHOD

The waterfall method is a structured, linear approach to software development. The process commences with the delineation of user needs. It advances through many planning phases, encompassing planning, modeling, and system

construction, culminating in the delivery of the software to users and the requisite support. The initial phase, analysis, involves interviewing firm officials to collect data and pinpoint existing problems that necessitate research assistance. A literature survey identifies pertinent sources for review, while defining problem boundaries helps constrain the scope to the author's competencies. This step entails analyzing both hardware and software requirements for the application.

The design step, following the analysis, entails creating multiple UML diagrams, including use case, class, sequence, and activity diagrams, as well as an Entity-Relationship Diagram (ERD). The coding phase marks the apex of the development process, during which the application design is implemented in VB.NET. Upon completion of the analysis, design, and coding phases, the testing phase commences, considered the final step in system development. The author employs black-box testing to assess the program's performance. Subsequently, post-delivery, the program may be modified to accommodate new peripherals or operating systems, ensuring its continued efficacy in its environment.



**Figure 1. Waterfall Method**

## RESULT AND DISCUSSION

Developers might utilize the interview results as a reference to construct the sales system by

---

identifying the stakeholders. All actors' needs have been clearly delineated in the actor identification, as illustrated in the table below.

**Table 1. Actor Identification**

No	Actor	Information
1	Admin	This is an actor who manages applications, including suppliers, incoming goods, outgoing goods, order processing, creating invoices, delivery notes, shipping labels and reports.
2	Owner	This is an actor who can only see goods information, incoming and outgoing goods data, invoices and reports.
3	Warehouse	Is an actor who manages applications in the form of suppliers, incoming goods, outgoing goods and reports.

a. **Design**

A use case refers to a particular task, such as system login, data creation, or data deletion. In this sense, an actor is a human entity that engages with the system to do certain tasks. The use case diagram for the storage information system features three

actors: Admin, Warehouse, and Owner.

User data administration is handled by the administrator, who is responsible for adding, modifying, and removing users. The following is a detailed account of the business events and user interactions with the system.



**Figure 2. Use Case Diagram**

**Table 2. Login Use Case Narrative**

Usecase	Login
Actor	Admin, Owner, Warehouse
Description	This use case describes the process of users (Admin, Owner, Warehouse) logging into the system to access the dashboard, and changing their account password if necessary.
Pre condition	The user is already registered in the system and has a valid username and password.
Main flow of event	

Action Actor	System Response
1. User opens login page	2. The system displays the login page.
3. The user enters the username and password.	
4. Click "Login"	5. The system validates the entered data
Alternate course	Alt 5.1 If the username or password is incorrect, the system will display an error message and ask the user to repeat the login process.
Post Condition	The user successfully logs into the system and is directed to the dashboard page or the system displays an error message if the login fails.

**Table 3. Use Narrative Sales Admin**

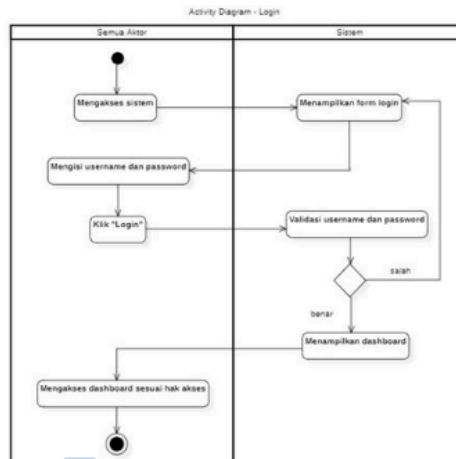
Use Narrative Sales Admin	
Objective	Sales Admin section can access the dashboard
Description	This system allows actors to process customer data, goods data, user data, sales orders, delivery orders, and invoices.
Actor	Admin
Initial Conditions	Actor opens dashboard
Actor	System Reaction
The actor selects the menu	The system will display the master data, sales transactions, and reports submenu.
The actor chooses to logout.	The system will display the process and exit from the activity.
Final Condition	If the command matches, the system will display what the actor selected.

**Table 4. Narrative of Incoming Goods Use Case**

Usecase	Incoming goods
Actor	Warehouse
Description	The warehouse can input data on newly incoming goods, change data if there is an error, delete invalid data, view a list of incoming goods, and print a report on the list of incoming goods.
Pre condition	The warehouse has been logged into the system
Main flow of event	
Action Actor	System Response
1. Warehouse opens incoming goods menu	2. The system displays the incoming goods page.
3. Warehouse selects goods input option	4. The system displays a form for adding new incoming goods data.
5. The warehouse fills in incoming goods data such as name of goods, quantity, qty, etc.	6. The system saves incoming goods data to the database.
7. Warehouse selects incoming goods list option	8. The system displays all stored incoming goods data.
Alternate course	Alt. 7.1 Warehouse can select the option to edit or delete incoming goods Alt 7.2 Warehouse can select the print incoming goods list option and the system will display a preview of the incoming goods list that will be printed.
Post Condition	Incoming goods data is successfully saved, updated, deleted, or printed according to the activities carried out.

b. Activity Diagram

The following activity diagram depicts the workflow of a menu system within the sales information system application:



12 Figure 3. Login Activity Diagram

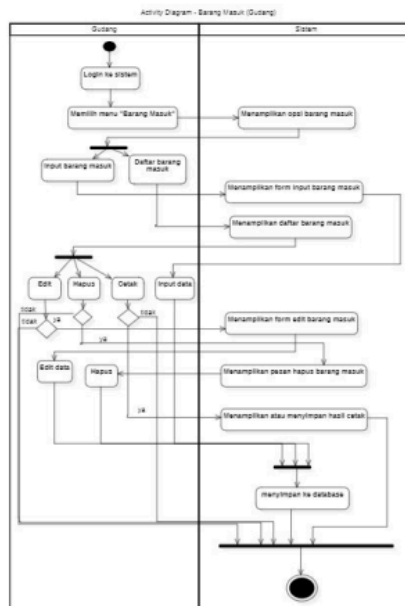


Figure 4. Activity Diagram for Incoming Goods

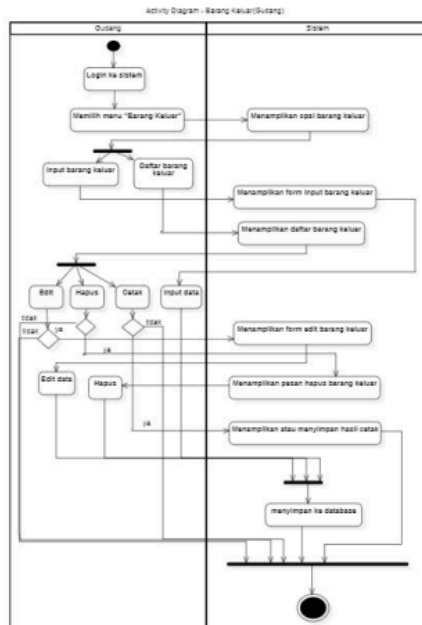


Figure 5. Activity Diagram for Outgoing Goods

c. <sup>1</sup> Sequence Diagram

The Sequence Diagram illustrates the order of messages sent between objects at particular moments during system execution. Objects are depicted as boxes,

lines with arrows represent messages, and the timeline is illustrated as a series of vertical processes. <sup>1</sup> The following is the sequence diagram for the login procedure of the storage information system:

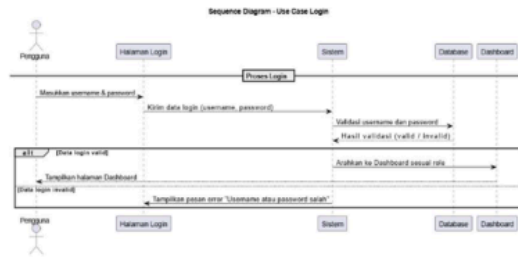
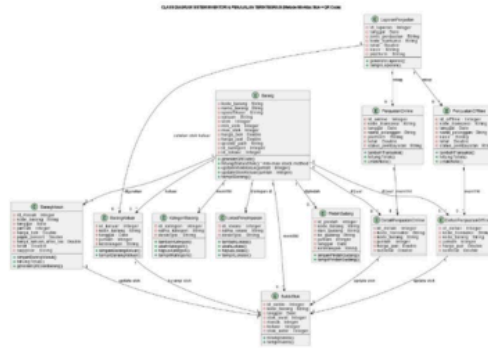


Figure 6. Login Sequence Diagram

d. Class Diagram

This class diagram delineates the objects that constitute a system and the

interrelationships among the object classes within it.

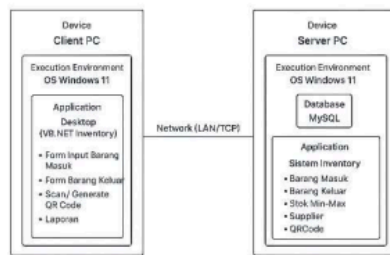


**Figure 7. Class Diagram**

e. System Architecture

The system architecture comprises the interaction between users and the web server via the internet and web browsers on user devices. The web server functions with

the XAMPP application and employs MySQL as the database server. The diagram of the system architecture has been generated below:



**Figure 8. System Architecture**

f. Database Implementation

**Figure 9. Database Table**

g. Implementation of the Min Max Stock Method

The transaction data for arriving and leaving items employs the Min-Max Stock

technique. This method aims to establish the minimum and maximum thresholds for stock storage using specified filters. The following are the commands from the Min-

Max Stock methodology on the incoming and outgoing items pages:

```

103 Cek Stok - cek stok
104
105 sub CekStok (kode, qty)
106     SELECT qty_stok, max_stok FROM barang_master WHERE kode_barang=kode
107
108     If qty_stok < qty Then
109         max_stok = qty
110     End If
111
112     If qty_stok > max_stok Then
113         max_stok = qty
114     End If
115
116     Return qty_stok
117 End Sub
118
119 sub CekStokValid (kode, qty)
120     If Not CekStokValid(kode, qty) Then
121         trans.Rollback()
122     End If
123 End Sub
124
125 sub CekStokValid (kode, qty)
126     If Not CekStokValid(kode, qty) Then
127         trans.Rollback()
128     End If
129 End Sub
130
131 sub CekStokValid (kode, qty)
132     If Not CekStokValid(kode, qty) Then
133         trans.Rollback()
134     End If
135 End Sub
136
137 sub CekStokValid (kode, qty)
138     If Not CekStokValid(kode, qty) Then
139         trans.Rollback()
140     End If
141 End Sub
142
143 sub CekStokValid (kode, qty)
144     If Not CekStokValid(kode, qty) Then
145         trans.Rollback()
146     End If
147 End Sub
148
149 sub CekStokValid (kode, qty)
150     If Not CekStokValid(kode, qty) Then
151         trans.Rollback()
152     End If
153 End Sub
154
155 sub CekStokValid (kode, qty)
156     If Not CekStokValid(kode, qty) Then
157         trans.Rollback()
158     End If
159 End Sub
160
161 sub CekStokValid (kode, qty)
162     If Not CekStokValid(kode, qty) Then
163         trans.Rollback()
164     End If
165 End Sub
166
167 sub CekStokValid (kode, qty)
168     If Not CekStokValid(kode, qty) Then
169         trans.Rollback()
170     End If
171 End Sub
172
173 sub CekStokValid (kode, qty)
174     If Not CekStokValid(kode, qty) Then
175         trans.Rollback()
176     End If
177 End Sub
178
179 sub CekStokValid (kode, qty)
180     If Not CekStokValid(kode, qty) Then
181         trans.Rollback()
182     End If
183 End Sub
184
185 sub CekStokValid (kode, qty)
186     If Not CekStokValid(kode, qty) Then
187         trans.Rollback()
188     End If
189 End Sub
190
191 sub CekStokValid (kode, qty)
192     If Not CekStokValid(kode, qty) Then
193         trans.Rollback()
194     End If
195 End Sub
196
197 sub CekStokValid (kode, qty)
198     If Not CekStokValid(kode, qty) Then
199         trans.Rollback()
200     End If
201 End Sub
202
203 sub CekStokValid (kode, qty)
204     If Not CekStokValid(kode, qty) Then
205         trans.Rollback()
206     End If
207 End Sub
208
209 sub CekStokValid (kode, qty)
210     If Not CekStokValid(kode, qty) Then
211         trans.Rollback()
212     End If
213 End Sub
214
215 sub CekStokValid (kode, qty)
216     If Not CekStokValid(kode, qty) Then
217         trans.Rollback()
218     End If
219 End Sub
220
221 sub CekStokValid (kode, qty)
222     If Not CekStokValid(kode, qty) Then
223         trans.Rollback()
224     End If
225 End Sub
226
227 sub CekStokValid (kode, qty)
228     If Not CekStokValid(kode, qty) Then
229         trans.Rollback()
230     End If
231 End Sub
232
233 sub CekStokValid (kode, qty)
234     If Not CekStokValid(kode, qty) Then
235         trans.Rollback()
236     End If
237 End Sub
238
239 sub CekStokValid (kode, qty)
240     If Not CekStokValid(kode, qty) Then
241         trans.Rollback()
242     End If
243 End Sub
244
245 sub CekStokValid (kode, qty)
246     If Not CekStokValid(kode, qty) Then
247         trans.Rollback()
248     End If
249 End Sub
250
251 sub CekStokValid (kode, qty)
252     If Not CekStokValid(kode, qty) Then
253         trans.Rollback()
254     End If
255 End Sub
256
257 sub CekStokValid (kode, qty)
258     If Not CekStokValid(kode, qty) Then
259         trans.Rollback()
260     End If
261 End Sub
262
263 sub CekStokValid (kode, qty)
264     If Not CekStokValid(kode, qty) Then
265         trans.Rollback()
266     End If
267 End Sub
268
269 sub CekStokValid (kode, qty)
270     If Not CekStokValid(kode, qty) Then
271         trans.Rollback()
272     End If
273 End Sub
274
275 sub CekStokValid (kode, qty)
276     If Not CekStokValid(kode, qty) Then
277         trans.Rollback()
278     End If
279 End Sub
280
281 sub CekStokValid (kode, qty)
282     If Not CekStokValid(kode, qty) Then
283         trans.Rollback()
284     End If
285 End Sub
286
287 sub CekStokValid (kode, qty)
288     If Not CekStokValid(kode, qty) Then
289         trans.Rollback()
290     End If
291 End Sub
292
293 sub CekStokValid (kode, qty)
294     If Not CekStokValid(kode, qty) Then
295         trans.Rollback()
296     End If
297 End Sub
298
299 sub CekStokValid (kode, qty)
300     If Not CekStokValid(kode, qty) Then
301         trans.Rollback()
302     End If
303 End Sub
304
305 sub CekStokValid (kode, qty)
306     If Not CekStokValid(kode, qty) Then
307         trans.Rollback()
308     End If
309 End Sub
310
311 sub CekStokValid (kode, qty)
312     If Not CekStokValid(kode, qty) Then
313         trans.Rollback()
314     End If
315 End Sub
316
317 sub CekStokValid (kode, qty)
318     If Not CekStokValid(kode, qty) Then
319         trans.Rollback()
320     End If
321 End Sub
322
323 sub CekStokValid (kode, qty)
324     If Not CekStokValid(kode, qty) Then
325         trans.Rollback()
326     End If
327 End Sub
328
329 sub CekStokValid (kode, qty)
330     If Not CekStokValid(kode, qty) Then
331         trans.Rollback()
332     End If
333 End Sub
334
335 sub CekStokValid (kode, qty)
336     If Not CekStokValid(kode, qty) Then
337         trans.Rollback()
338     End If
339 End Sub
340
341 sub CekStokValid (kode, qty)
342     If Not CekStokValid(kode, qty) Then
343         trans.Rollback()
344     End If
345 End Sub
346
347 sub CekStokValid (kode, qty)
348     If Not CekStokValid(kode, qty) Then
349         trans.Rollback()
350     End If
351 End Sub
352
353 sub CekStokValid (kode, qty)
354     If Not CekStokValid(kode, qty) Then
355         trans.Rollback()
356     End If
357 End Sub
358
359 sub CekStokValid (kode, qty)
360     If Not CekStokValid(kode, qty) Then
361         trans.Rollback()
362     End If
363 End Sub
364
365 sub CekStokValid (kode, qty)
366     If Not CekStokValid(kode, qty) Then
367         trans.Rollback()
368     End If
369 End Sub
370
371 sub CekStokValid (kode, qty)
372     If Not CekStokValid(kode, qty) Then
373         trans.Rollback()
374     End If
375 End Sub
376
377 sub CekStokValid (kode, qty)
378     If Not CekStokValid(kode, qty) Then
379         trans.Rollback()
380     End If
381 End Sub
382
383 sub CekStokValid (kode, qty)
384     If Not CekStokValid(kode, qty) Then
385         trans.Rollback()
386     End If
387 End Sub
388
389 sub CekStokValid (kode, qty)
390     If Not CekStokValid(kode, qty) Then
391         trans.Rollback()
392     End If
393 End Sub
394
395 sub CekStokValid (kode, qty)
396     If Not CekStokValid(kode, qty) Then
397         trans.Rollback()
398     End If
399 End Sub
400
401 sub CekStokValid (kode, qty)
402     If Not CekStokValid(kode, qty) Then
403         trans.Rollback()
404     End If
405 End Sub
406
407 sub CekStokValid (kode, qty)
408     If Not CekStokValid(kode, qty) Then
409         trans.Rollback()
410     End If
411 End Sub
412
413 sub CekStokValid (kode, qty)
414     If Not CekStokValid(kode, qty) Then
415         trans.Rollback()
416     End If
417 End Sub
418
419 sub CekStokValid (kode, qty)
420     If Not CekStokValid(kode, qty) Then
421         trans.Rollback()
422     End If
423 End Sub
424
425 sub CekStokValid (kode, qty)
426     If Not CekStokValid(kode, qty) Then
427         trans.Rollback()
428     End If
429 End Sub
430
431 sub CekStokValid (kode, qty)
432     If Not CekStokValid(kode, qty) Then
433         trans.Rollback()
434     End If
435 End Sub
436
437 sub CekStokValid (kode, qty)
438     If Not CekStokValid(kode, qty) Then
439         trans.Rollback()
440     End If
441 End Sub
442
443 sub CekStokValid (kode, qty)
444     If Not CekStokValid(kode, qty) Then
445         trans.Rollback()
446     End If
447 End Sub
448
449 sub CekStokValid (kode, qty)
450     If Not CekStokValid(kode, qty) Then
451         trans.Rollback()
452     End If
453 End Sub
454
455 sub CekStokValid (kode, qty)
456     If Not CekStokValid(kode, qty) Then
457         trans.Rollback()
458     End If
459 End Sub
460
461 sub CekStokValid (kode, qty)
462     If Not CekStokValid(kode, qty) Then
463         trans.Rollback()
464     End If
465 End Sub
466
467 sub CekStokValid (kode, qty)
468     If Not CekStokValid(kode, qty) Then
469         trans.Rollback()
470     End If
471 End Sub
472
473 sub CekStokValid (kode, qty)
474     If Not CekStokValid(kode, qty) Then
475         trans.Rollback()
476     End If
477 End Sub
478
479 sub CekStokValid (kode, qty)
480     If Not CekStokValid(kode, qty) Then
481         trans.Rollback()
482     End If
483 End Sub
484
485 sub CekStokValid (kode, qty)
486     If Not CekStokValid(kode, qty) Then
487         trans.Rollback()
488     End If
489 End Sub
490
491 sub CekStokValid (kode, qty)
492     If Not CekStokValid(kode, qty) Then
493         trans.Rollback()
494     End If
495 End Sub
496
497 sub CekStokValid (kode, qty)
498     If Not CekStokValid(kode, qty) Then
499         trans.Rollback()
500     End If
501 End Sub
502
503 sub CekStokValid (kode, qty)
504     If Not CekStokValid(kode, qty) Then
505         trans.Rollback()
506     End If
507 End Sub
508
509 sub CekStokValid (kode, qty)
510     If Not CekStokValid(kode, qty) Then
511         trans.Rollback()
512     End If
513 End Sub
514
515 sub CekStokValid (kode, qty)
516     If Not CekStokValid(kode, qty) Then
517         trans.Rollback()
518     End If
519 End Sub
520
521 sub CekStokValid (kode, qty)
522     If Not CekStokValid(kode, qty) Then
523         trans.Rollback()
524     End If
525 End Sub
526
527 sub CekStokValid (kode, qty)
528     If Not CekStokValid(kode, qty) Then
529         trans.Rollback()
530     End If
531 End Sub
532
533 sub CekStokValid (kode, qty)
534     If Not CekStokValid(kode, qty) Then
535         trans.Rollback()
536     End If
537 End Sub
538
539 sub CekStokValid (kode, qty)
540     If Not CekStokValid(kode, qty) Then
541         trans.Rollback()
542     End If
543 End Sub
544
545 sub CekStokValid (kode, qty)
546     If Not CekStokValid(kode, qty) Then
547         trans.Rollback()
548     End If
549 End Sub
550
551 sub CekStokValid (kode, qty)
552     If Not CekStokValid(kode, qty) Then
553         trans.Rollback()
554     End If
555 End Sub
556
557 sub CekStokValid (kode, qty)
558     If Not CekStokValid(kode, qty) Then
559         trans.Rollback()
560     End If
561 End Sub
562
563 sub CekStokValid (kode, qty)
564     If Not CekStokValid(kode, qty) Then
565         trans.Rollback()
566     End If
567 End Sub
568
569 sub CekStokValid (kode, qty)
570     If Not CekStokValid(kode, qty) Then
571         trans.Rollback()
572     End If
573 End Sub
574
575 sub CekStokValid (kode, qty)
576     If Not CekStokValid(kode, qty) Then
577         trans.Rollback()
578     End If
579 End Sub
580
581 sub CekStokValid (kode, qty)
582     If Not CekStokValid(kode, qty) Then
583         trans.Rollback()
584     End If
585 End Sub
586
587 sub CekStokValid (kode, qty)
588     If Not CekStokValid(kode, qty) Then
589         trans.Rollback()
590     End If
591 End Sub
592
593 sub CekStokValid (kode, qty)
594     If Not CekStokValid(kode, qty) Then
595         trans.Rollback()
596     End If
597 End Sub
598
599 sub CekStokValid (kode, qty)
600     If Not CekStokValid(kode, qty) Then
601         trans.Rollback()
602     End If
603 End Sub
604
605 sub CekStokValid (kode, qty)
606     If Not CekStokValid(kode, qty) Then
607         trans.Rollback()
608     End If
609 End Sub
610
611 sub CekStokValid (kode, qty)
612     If Not CekStokValid(kode, qty) Then
613         trans.Rollback()
614     End If
615 End Sub
616
617 sub CekStokValid (kode, qty)
618     If Not CekStokValid(kode, qty) Then
619         trans.Rollback()
620     End If
621 End Sub
622
623 sub CekStokValid (kode, qty)
624     If Not CekStokValid(kode, qty) Then
625         trans.Rollback()
626     End If
627 End Sub
628
629 sub CekStokValid (kode, qty)
630     If Not CekStokValid(kode, qty) Then
631         trans.Rollback()
632     End If
633 End Sub
634
635 sub CekStokValid (kode, qty)
636     If Not CekStokValid(kode, qty) Then
637         trans.Rollback()
638     End If
639 End Sub
640
641 sub CekStokValid (kode, qty)
642     If Not CekStokValid(kode, qty) Then
643         trans.Rollback()
644     End If
645 End Sub
646
647 sub CekStokValid (kode, qty)
648     If Not CekStokValid(kode, qty) Then
649         trans.Rollback()
650     End If
651 End Sub
652
653 sub CekStokValid (kode, qty)
654     If Not CekStokValid(kode, qty) Then
655         trans.Rollback()
656     End If
657 End Sub
658
659 sub CekStokValid (kode, qty)
660     If Not CekStokValid(kode, qty) Then
661         trans.Rollback()
662     End If
663 End Sub
664
665 sub CekStokValid (kode, qty)
666     If Not CekStokValid(kode, qty) Then
667         trans.Rollback()
668     End If
669 End Sub
670
671 sub CekStokValid (kode, qty)
672     If Not CekStokValid(kode, qty) Then
673         trans.Rollback()
674     End If
675 End Sub
676
677 sub CekStokValid (kode, qty)
678     If Not CekStokValid(kode, qty) Then
679         trans.Rollback()
680     End If
681 End Sub
682
683 sub CekStokValid (kode, qty)
684     If Not CekStokValid(kode, qty) Then
685         trans.Rollback()
686     End If
687 End Sub
688
689 sub CekStokValid (kode, qty)
690     If Not CekStokValid(kode, qty) Then
691         trans.Rollback()
692     End If
693 End Sub
694
695 sub CekStokValid (kode, qty)
696     If Not CekStokValid(kode, qty) Then
697         trans.Rollback()
698     End If
699 End Sub
700
701 sub CekStokValid (kode, qty)
702     If Not CekStokValid(kode, qty) Then
703         trans.Rollback()
704     End If
705 End Sub
706
707 sub CekStokValid (kode, qty)
708     If Not CekStokValid(kode, qty) Then
709         trans.Rollback()
710     End If
711 End Sub
712
713 sub CekStokValid (kode, qty)
714     If Not CekStokValid(kode, qty) Then
715         trans.Rollback()
716     End If
717 End Sub
718
719 sub CekStokValid (kode, qty)
720     If Not CekStokValid(kode, qty) Then
721         trans.Rollback()
722     End If
723 End Sub
724
725 sub CekStokValid (kode, qty)
726     If Not CekStokValid(kode, qty) Then
727         trans.Rollback()
728     End If
729 End Sub
730
731 sub CekStokValid (kode, qty)
732     If Not CekStokValid(kode, qty) Then
733         trans.Rollback()
734     End If
735 End Sub
736
737 sub CekStokValid (kode, qty)
738     If Not CekStokValid(kode, qty) Then
739         trans.Rollback()
740     End If
741 End Sub
742
743 sub CekStokValid (kode, qty)
744     If Not CekStokValid(kode, qty) Then
745         trans.Rollback()
746     End If
747 End Sub
748
749 sub CekStokValid (kode, qty)
750     If Not CekStokValid(kode, qty) Then
751         trans.Rollback()
752     End If
753 End Sub
754
755 sub CekStokValid (kode, qty)
756     If Not CekStokValid(kode, qty) Then
757         trans.Rollback()
758     End If
759 End Sub
760
761 sub CekStokValid (kode, qty)
762     If Not CekStokValid(kode, qty) Then
763         trans.Rollback()
764     End If
765 End Sub
766
767 sub CekStokValid (kode, qty)
768     If Not CekStokValid(kode, qty) Then
769         trans.Rollback()
770     End If
771 End Sub
772
773 sub CekStokValid (kode, qty)
774     If Not CekStokValid(kode, qty) Then
775         trans.Rollback()
776     End If
777 End Sub
778
779 sub CekStokValid (kode, qty)
780     If Not CekStokValid(kode, qty) Then
781         trans.Rollback()
782     End If
783 End Sub
784
785 sub CekStokValid (kode, qty)
786     If Not CekStokValid(kode, qty) Then
787         trans.Rollback()
788     End If
789 End Sub
790
791 sub CekStokValid (kode, qty)
792     If Not CekStokValid(kode, qty) Then
793         trans.Rollback()
794     End If
795 End Sub
796
797 sub CekStokValid (kode, qty)
798     If Not CekStokValid(kode, qty) Then
799         trans.Rollback()
800     End If
801 End Sub
802
803 sub CekStokValid (kode, qty)
804     If Not CekStokValid(kode, qty) Then
805         trans.Rollback()
806     End If
807 End Sub
808
809 sub CekStokValid (kode, qty)
810     If Not CekStokValid(kode, qty) Then
811         trans.Rollback()
812     End If
813 End Sub
814
815 sub CekStokValid (kode, qty)
816     If Not CekStokValid(kode, qty) Then
817         trans.Rollback()
818     End If
819 End Sub
820
821 sub CekStokValid (kode, qty)
822     If Not CekStokValid(kode, qty) Then
823         trans.Rollback()
824     End If
825 End Sub
826
827 sub CekStokValid (kode, qty)
828     If Not CekStokValid(kode, qty) Then
829         trans.Rollback()
830     End If
831 End Sub
832
833 sub CekStokValid (kode, qty)
834     If Not CekStokValid(kode, qty) Then
835         trans.Rollback()
836     End If
837 End Sub
838
839 sub CekStokValid (kode, qty)
840     If Not CekStokValid(kode, qty) Then
841         trans.Rollback()
842     End If
843 End Sub
844
845 sub CekStokValid (kode, qty)
846     If Not CekStokValid(kode, qty) Then
847         trans.Rollback()
848     End If
849 End Sub
850
851 sub CekStokValid (kode, qty)
852     If Not CekStokValid(kode, qty) Then
853         trans.Rollback()
854     End If
855 End Sub
856
857 sub CekStokValid (kode, qty)
858     If Not CekStokValid(kode, qty) Then
859         trans.Rollback()
860     End If
861 End Sub
862
863 sub CekStokValid (kode, qty)
864     If Not CekStokValid(kode, qty) Then
865         trans.Rollback()
866     End If
867 End Sub
868
869 sub CekStokValid (kode, qty)
870     If Not CekStokValid(kode, qty) Then
871         trans.Rollback()
872     End If
873 End Sub
874
875 sub CekStokValid (kode, qty)
876     If Not CekStokValid(kode, qty) Then
877         trans.Rollback()
878     End If
879 End Sub
880
881 sub CekStokValid (kode, qty)
882     If Not CekStokValid(kode, qty) Then
883         trans.Rollback()
884     End If
885 End Sub
886
887 sub CekStokValid (kode, qty)
888     If Not CekStokValid(kode, qty) Then
889         trans.Rollback()
890     End If
891 End Sub
892
893 sub CekStokValid (kode, qty)
894     If Not CekStokValid(kode, qty) Then
895         trans.Rollback()
896     End If
897 End Sub
898
899 sub CekStokValid (kode, qty)
900     If Not CekStokValid(kode, qty) Then
901         trans.Rollback()
902     End If
903 End Sub
904
905 sub CekStokValid (kode, qty)
906     If Not CekStokValid(kode, qty) Then
907         trans.Rollback()
908     End If
909 End Sub
910
911 sub CekStokValid (kode, qty)
912     If Not CekStokValid(kode, qty) Then
913         trans.Rollback()
914     End If
915 End Sub
916
917 sub CekStokValid (kode, qty)
918     If Not CekStokValid(kode, qty) Then
919         trans.Rollback()
920     End If
921 End Sub
922
923 sub CekStokValid (kode, qty)
924     If Not CekStokValid(kode, qty) Then
925         trans.Rollback()
926     End If
927 End Sub
928
929 sub CekStokValid (kode, qty)
930     If Not CekStokValid(kode, qty) Then
931         trans.Rollback()
932     End If
933 End Sub
934
935 sub CekStokValid (kode, qty)
936     If Not CekStokValid(kode, qty) Then
937         trans.Rollback()
938     End If
939 End Sub
940
941 sub CekStokValid (kode, qty)
942     If Not CekStokValid(kode, qty) Then
943         trans.Rollback()
944     End If
945 End Sub
946
947 sub CekStokValid (kode, qty)
948     If Not CekStokValid(kode, qty) Then
949         trans.Rollback()
950     End If
951 End Sub
952
953 sub CekStokValid (kode, qty)
954     If Not CekStokValid(kode, qty) Then
955         trans.Rollback()
956     End If
957 End Sub
958
959 sub CekStokValid (kode, qty)
960     If Not CekStokValid(kode, qty) Then
961         trans.Rollback()
962     End If
963 End Sub
964
965 sub CekStokValid (kode, qty)
966     If Not CekStokValid(kode, qty) Then
967         trans.Rollback()
968     End If
969 End Sub
970
971 sub CekStokValid (kode, qty)
972     If Not CekStokValid(kode, qty) Then
973         trans.Rollback()
974     End If
975 End Sub
976
977 sub CekStokValid (kode, qty)
978     If Not CekStokValid(kode, qty) Then
979         trans.Rollback()
980     End If
981 End Sub
982
983 sub CekStokValid (kode, qty)
984     If Not CekStokValid(kode, qty) Then
985         trans.Rollback()
986     End If
987 End Sub
988
989 sub CekStokValid (kode, qty)
990     If Not CekStokValid(kode, qty) Then
991         trans.Rollback()
992     End If
993 End Sub
994
995 sub CekStokValid (kode, qty)
996     If Not CekStokValid(kode, qty) Then
997         trans.Rollback()
998     End If
999 End Sub
1000

```

Figure 10. Check Min Max Stock of Incoming Goods

```

'Cek stok min-max
If Not CekStokValid(kode, qty) Then
    trans.Rollback()
Exit Sub
End If

```

Figure 11. Check Min Max Stock of Outgoing Goods

h. Interface Implementation

During the implementation phase, the interface is developed according to the

designs established in the preceding stage to enhance user comprehension of the system.



Figure 12. Login Page Implementation



Figure 13. Implementation of the Dashboard Page

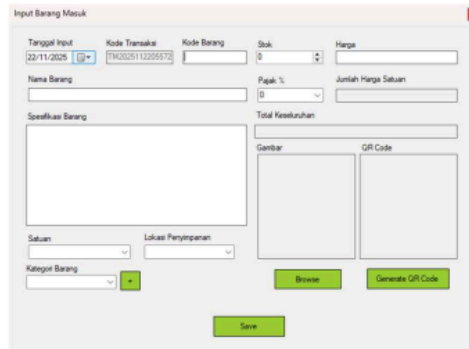


Figure 14. Implementation of the Incoming Goods Page

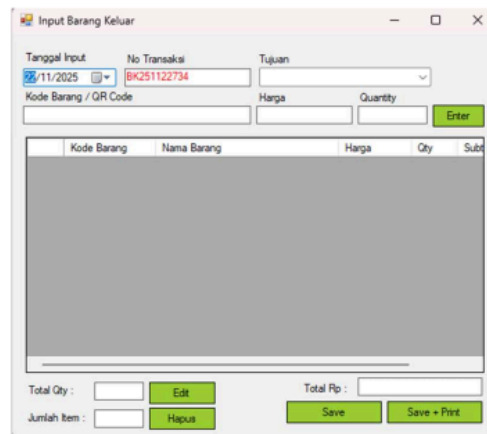


Figure 15. Implementation of the Outgoing Goods Page

i. Testing

This application testing emphasizes Black-Box Testing to assess the system's functionality and analyze program

execution outcomes. It guarantees that designated inputs will activate the appropriate processes and yield outputs consistent with the design.

Table 5. Login Testing

No	Testing Scenario	Input	Expected Output	Results
1	Login with correct username and password	username: admin, password: correct	The system accepts login and enters the dashboard.	Succeed

2	Login with correct username, password wrong	username: admin, password: wrong	The system rejects and displays the message "Wrong password"	Succeed
3	Login with unregistered username	unknown username	The system displays the message "Username not found"	Succeed
4	Login without filling in any of the fields	empty username or empty password	The system rejects login and asks you to complete the data.	Succeed
5	Logout	Click the logout button	The system returns to the login page.	Succeed

**Table 6. Testing of Incoming Goods**

No	Testing Scenario	Input	Expected Output	Results
1	Adding new item data	Complete data (name, unit, initial stock, min-max)	Complete data (name, unit, initial stock, min-max)	Succeed
2	Adding item data without filling in any of the fields	Empty field	The system rejects and displays an error message.	Succeed
3	Changing item data	Change of name/stock/unit	Data changes according to input	Succeed
4	Delete item data	Select item → delete	Data deleted from the items table	Succeed

**Table 7. QR Code Generation Testing**

No	Testing Scenario	Input	Expected Output	Results
1	Generate QR Code	Valid Item ID	QR Code is formed and appears in PictureBox	Succeed

2	Generate QR Code with empty Item Id	Empty ID	The system rejects and displays an error message.	Succeed
3	Saving QR Code	Click the save button	Saved QR Code PNG file	Succeed

## CONCLUSION

The established system, utilizing the Software Development Life Cycle (SDLC) Waterfall methodology, has been executed systematically from the analytical phase to testing. This method efficiently addresses problems that were hitherto managed manually. The system optimizes data recording, management, and presentation processes, making them swifter, more precise, and more efficient. Functions such as transaction documentation, item data administration, automated database storage, and report generation have demonstrated enhanced support for users in their daily operations. Test findings indicate that all functionalities perform well and meet user requirements, rendering the system suitable for routine operational tasks.

## REFERENCES

---

ORIGINALITY REPORT

14%  
SIMILARITY INDEX

13%  
INTERNET SOURCES

3%  
PUBLICATIONS

3%  
STUDENT PAPERS

PRIMARY SOURCES

- 1 [jurnal.stmik-mi.ac.id](http://jurnal.stmik-mi.ac.id)  
Internet Source
- 2 [rubynwa.blogspot.com](http://rubynwa.blogspot.com)  
Internet Source
- 3 [ojspanel.undikma.ac.id](http://ojspanel.undikma.ac.id)  
Internet Source
- 4 Submitted to Islington College, Nepal  
Student Paper
- 5 [jurnal.itscience.org](http://jurnal.itscience.org)  
Internet Source
- 6 José Rouillard. "Contextual QR Codes", 2008 The Third International Multi-Conference on Computing in the Global Information Technology (iccg 2008), 2008  
Publication
- 7 [umpir.ump.edu.my](http://umpir.ump.edu.my)  
Internet Source
- 8 [colomnewbie.blogspot.com](http://colomnewbie.blogspot.com)  
Internet Source
- 9 Submitted to Online Education Services  
Student Paper
- 10 Submitted to University of Technology, Sydney  
Student Paper
- 11 [www.coursehero.com](http://www.coursehero.com)  
Internet Source
- 12 [journal.admi.or.id](http://journal.admi.or.id)  
Internet Source
- 13 [pdfcoffee.com](http://pdfcoffee.com)  
Internet Source
- 14 [norma.ncirl.ie](http://norma.ncirl.ie)  
Internet Source

---

Exclude quotes

Off

Exclude matches

Off

Exclude bibliography

Off