Research Article

Implementation of Data Access Object Pattern in Bookkeeping System Development

Ahlijati Nuraminah [1*]

[1] Department of Computer Science, ESQ College of Management and Computer Science

[*] Corresponding Author: ahlijati.nuraminah@esqbs.ac.id

Abstract: The bookkeeping system is vital for managing a catering business, and incorporating design patterns in software development, specifically the Data Access Object (DAO) Pattern, can enhance structure, flexibility, and efficiency. This study focuses on implementing the DAO Pattern in the catering bookkeeping system to improve data management and overall system performance. Utilizing a software development framework, the DAO Pattern facilitates CRUD operations (Create, Read, Update, Delete) on entities, ensuring flexibility. Tests validate data integrity and proper DAO functions. The system test results through unit testing showed that 96% of the features that implemented DAO were successfully implemented. This demonstrates significant benefits, with improved data management structure and centralized changes. Implementing the DAO Pattern in catering's bookkeeping system ultimately enhances structured data management and overall system performance, providing insights for software developers working on similar business applications.

Keywords: data access object; desain pattern; bookkeeping; catering; software developers

Citation: A.Nuraminah," Implementation of Data Access Object Pattern in Bookkeeping System Development". *Iota*, **2024**, ISSN 2774-4353, Vol.04, 01. https://doi.org/10.31763/iota.v4i1.712

Academic Editor : Adi, P.D.P

Received : January, 12 2024

Accepted : January, 21 2024

Published : February, 17 2024

1. Introduction

The development of information technology has had a significant impact on life, including in the catering business established in the city of Depok since 2012 (Ammar et al., 2021). Currently, bookkeeping still uses a manual method of recording all transactions in a book, causing the bookkeeping process to be slow. Digital bookkeeping, as expressed by the catering owner in the interview, provides convenience in data management, improves financial efficiency, saves costs, maintains data security, and accelerates decision-making through reporting features. Catering service providers need an efficient accounting system. Therefore, an application to manage customer data, menus, payment transactions, etc., is required. In the development of accounting systems, the Structural Programming method is generally used (Dodi et al., 2018). Still, this method has weaknesses in modularity (O.-J. Dahl et al., 2021) and database access (Khairunnisa & Rismayanti, 2020).

The Data Access Object (DAO) design pattern is used to overcome these weaknesses, separating business logic from database access and making the accounting system more structured (Nock, 2008). The use of the DAO Pattern can also improve the performance (Aniche et al., 2014), security (Jánki & Bilicki, 2023), and data integrity (Kouli & Rasoolzadegan, 2022) of the bookkeeping system. This facilitates system maintenance and development, as the database structure or technology changes can be isolated within the DAO class without affecting the associated business logic (Matic et al., 2004).

Previous research that also discussed Data Access Objects was research (Fajar et al., 2023), which focused on developing an Android-based daily schedule reminder application by utilizing the Model View Controller (MVC) and Data Access Object (DAO) pattern approaches. In crowded cities, time management is crucial, and the use of smartphones has become a daily necessity. This application is designed to help users manage their activities and schedules. The MVC and DAO approach was chosen to facilitate future modifications to the application. The evaluation results show that using this approach increases the modification rate on various metrics, such as Attribute Hiding

Factor (AHF), Polymorphism Factor (POF), Number of Methods Inherited (NMO), and Reuse Ratio (RR) of the system. Thus, the MVC and DAO patterns proved their ability to ease application modifications in future development and maintenance. However, this research does not focus on functions to access the database, so it is necessary to investigate further the benefits of DAO in functions related to database access. Therefore, the research aims to implement the DAO Pattern in data access to the catering accounting system application database.

2. Method

Several stages need to be carried out in conducting this research. The figure represents each stage carried out in the implementation of this research. Based on the Figure, the flow process of this research is divided into three parts so that the implementation of this research can be measured and directed. The following is an explanation of each flow carried out in Figure 1.

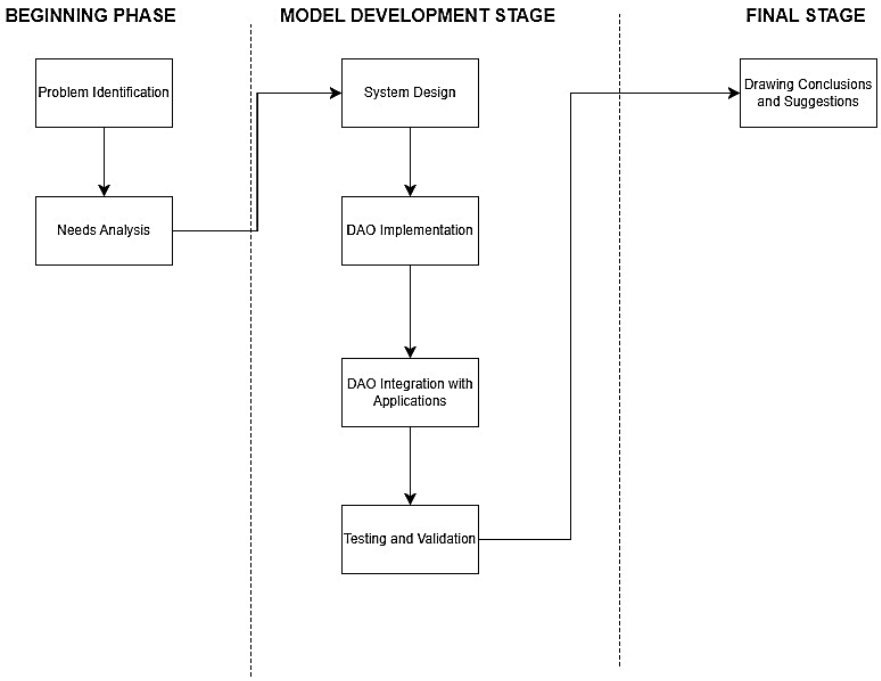

Figure 1. Research Flow

2.1 Problem Identification

The first step is to identify the problem that needs to be solved. In this context, some of the issues that may arise include the absence of separation of business logic and data access in the existing bookkeeping system, slow bookkeeping system performance due to inefficient data management, and vulnerable data security and integrity due to direct access to the database.

2.2 Needs Analysis

Next, analyze the needs of the catering bookkeeping system in depth. Identify the functional and non-functional requirements that must be met by the bookkeeping system implemented using the Data Access Object Pattern. Based on the results of interviews with catering owners, there are two needs required by the system (Aulia Aziiza & Fadhilah, 2020), namely, functional needs and non-functional needs. The following are the functional and non-functional requirements needed:

- Functional Needs contain any features that the system will carry out (Turserno & Rosihan, 2022).
  - To do bookkeeping in the system, users must be registered in the system which is divided into two account categories, namely Admin and Owner.
  - Admin and owner can add, view, change, cancel, and delete an order.
  - Admin and owner can add, view, change, cancel, and delete an order item.
  - Admin and owner can add, view, change, cancel, and delete a category.
  - Admin and owner can add, view, change, and delete income or expense transactions.
  - The owner can view income and expense reports daily, weekly, and annually.
  - The owner can view order and payment reports daily, weekly, and annually.
  - The owner can view the profit and loss report in the system.
  - The owner can view graphs of income and expenses on a daily, weekly, and annual basis.

- Non-Functional Requirements describe how the system will work (Hakim et al., 2019).
  - Availability. The system can operate seven days per week, 24 hours without fail.
  - Ergonomy. The system can be used easily by all users.
  - Portability. All browser versions and all devices can access the system.
  - Security. All stored data cannot be accessed other than by the database administrator.

2.3 System Design

Database design refers to designing the structure and organization of data that will be used in the database system (Pradipta et al., 2022). Database design involves modeling entities, attributes, and relationships between entities to meet business needs and optimize the performance and efficiency of database systems (Connoly & Begg, 2019). Good database design ensures that the database is efficient, consistent, secure, and easily accessible to support the desired information system needs (Trevy Jonatya Novella et al., 2022). The database design can be seen in Figure 2.

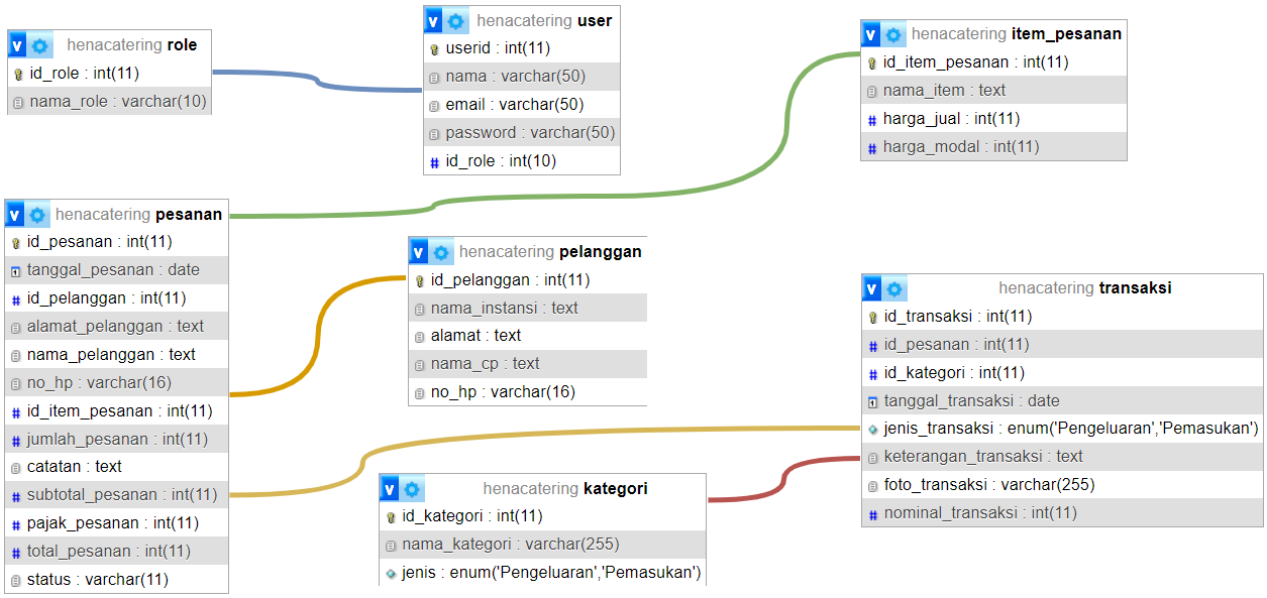

Figure 2. Database Design

The Role table aims to store data from system user roles, the User table aims to store user data, the Order table aims to store orders that have been ordered, the Order Item table stores data from order items, the Customer table holds data from customers who order, the Category table stores data on categories of orders, the Transaction table stores data from Transactions that have been made. Between tables are connected through foreign keys.

The class diagram in UML clearly illustrates the structure and description of each object's classes, attributes, methods, and relationships (Sandfreni et al., 2021). Class diagrams are static (Suharni et al., 2023), in the sense that class diagrams do not explain what happens if the classes are related (Indra Andhika et al., 2022) but rather explain what relationships occur. Class diagrams can help developers understand the overview of the application scheme and map the system structure (Nabila et al., 2021) clearly by describing classes, attributes, operations, and relationships between objects.

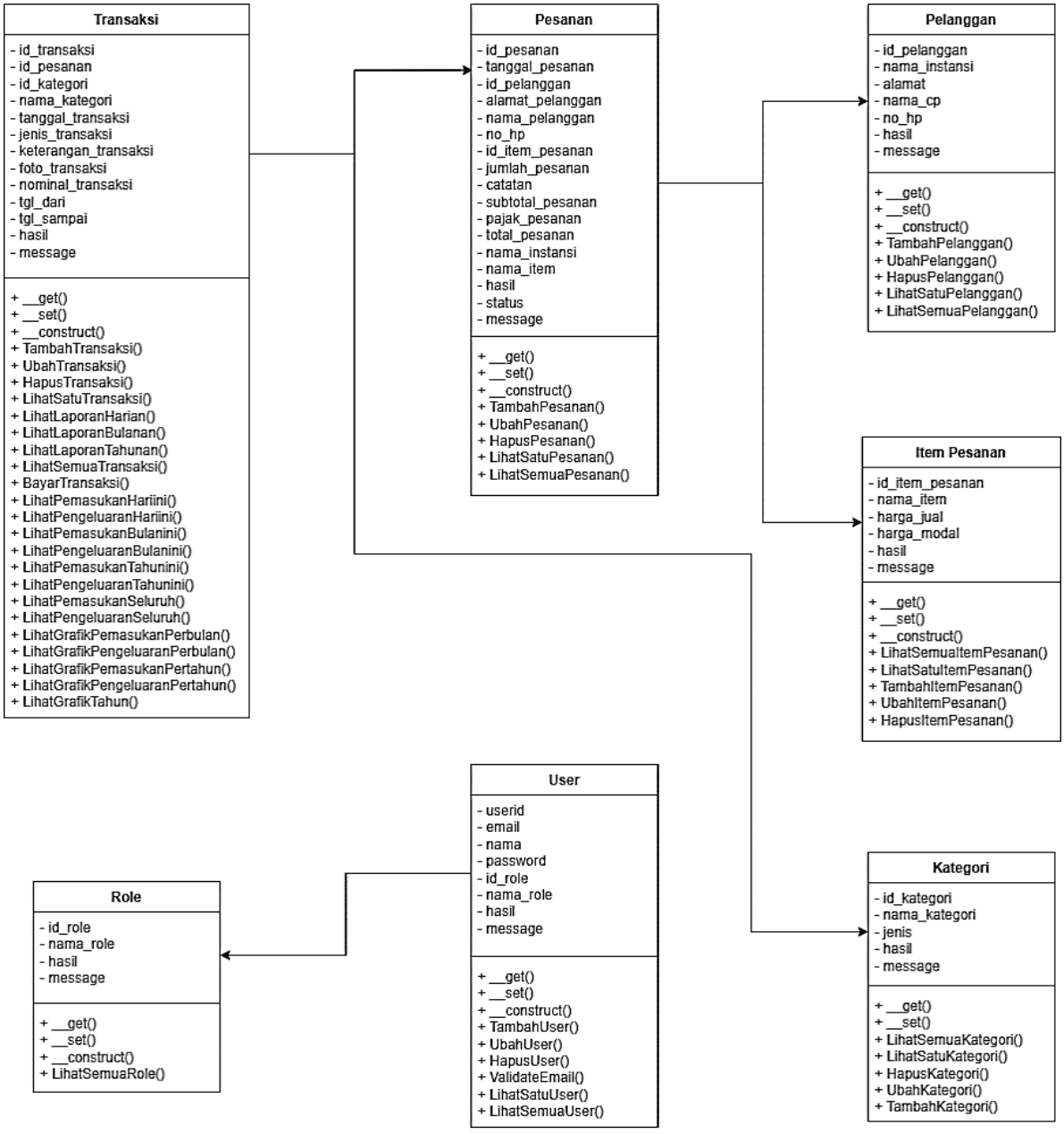

Figure 3. Class Diagram Design

From the class diagram design in Figure 3, the existing class, namely the transaction class, describes the operations that occur in the transaction table. The order class has functions related to managing orders in the system. The customer class has the function of managing customers in the system. The order item class has a function to manage order items such as order name, order type, selling price, and capital price. The category class has a function to control the category of the transaction. The user class has a function to work with users or to help users use the system. The role class has a function to manage the user's role by distinguishing the user into two, namely admin and owner.

2.4 Implementation of Data Access Object Pattern

At this stage, implement the Data Access Object Pattern in developing the catering bookkeeping system. Create DAO interfaces that define the necessary data access operations, then implement DAO classes that connect the system with databases or other data storage systems.

2.5 Integration with Bookkeeping Application

Next, integrate the DAO class into the catering bookkeeping application. Modify the existing bookkeeping application code to use the DAO interface and perform data access through the DAO class.

2.6 Testing and Validation

Perform thorough system testing to ensure the Data Access Object Pattern implementation works properly. Test the system's functionality, security, data integrity, and performance. Ensure that the catering bookkeeping system can access and manage data efficiently and accurately.

2.7 Evaluation and Improvement

After testing, evaluate the Data Access Object Pattern implementation results on the bookkeeping system. Identify strengths and weaknesses and find solutions to improve the system. Make repairs and improvements according to the evaluation results obtained.

2.8 Implementation and Maintenance

After making repairs and improvements, implement the latest version of the catering bookkeeping system. Ensure the system is running correctly and perform regular maintenance to maintain system performance and security.

3. Result and Analysis

3.1 DAO Implementation

In a DAO (Data Access Object) implementation, the "connection" class is not explicitly related to the DAO itself. However, in general, the "connection" class can be used to manage connections to data sources, such as databases (Hidayat et al., 2019). The implementation of the connection class in the "class.user.php" class is done in the following way:

a) Create variables to store server data, usernames, passwords, and system databases, and connection variables to store database connections.
b) Create a constructor to connect to the database.
c) Create a connect method that contains the mysqli_connect function call, then select the database used and save the connection data into the connection variable.

Setter and getter methods are methods used to set (set) and retrieve (get) the value of an object's property (field). Programming languages such as PHP, setter, and getter methods ensure that access to properties is done through methods, thus allowing validation or additional logic before property values are changed or retrieved. The implementation of setter and getter methods in the "class.user.php" class is as follows:

a) Create a getter function with an attribute name parameter to return the value of the attribute in question.

b) Create a setter function with an attribute name and value parameters to store the value in the attribute.

Moreover, Getter and setter functions are part of a class and are used to implement a "magic method" called "__get()." The "__get()" function is used to access properties (attributes) that cannot be accessed directly from outside the class. When a property that cannot be accessed directly is accessed from outside the class, the "__get()" method will be called automatically. In the code, the "__get()" function accepts one parameter, "$attribute," which is the name of the property to be accessed. This function checks using the "property_exists()" function to determine if the property exists in the current object ($this). If the property exists in the object, then the value of the property will be returned using the syntax "$this->$atribute", where "$atribute" is the name of the property being accessed.

The "add" method in PHP is commonly used to add data to a data source, such as a database. In the context of a DAO implementation, the "add" method can be used to add user data to the "userid" table. The implementation of the "add" method in the "class.user.php" class is done as follows:

a) Create the AddUser method.
b) Call the connect method found in the Connection class.
c) Create a SQL query to store data in the table with the INSERT command.
d) Execute the query by calling the mysqli_query command to save the execution results in the result variable.
e) If the execution result is TRUE, then create a message that the data was successfully added; otherwise, create a message that the data failed.

Furthermore, The "change" method in PHP is generally used to change data that already exists in a data source, such as updating records in a database table. In the context of a DAO implementation, the "change" method can be used to change existing user data in the "user-id" table. The implementation of the "change" method in the "class.user.php" class is as follows:

a) Create the ChangeUser method.
b) Call the connect method contained in the Connection class
c) Create a SQL query to change the contents of a row in the table with the UPDATE command
d) Execute the query by calling the mysqli_query command and store the execution result in the result variable
e) If the result of execution is TRUE, then create a successful message to change the data; otherwise, create a failed message to change the data.

The "delete" method in PHP is generally used to delete data already in a data source, such as deleting records from a database table. In the context of a DAO implementation, the "delete" method can be used to delete user data from the "userid" table. The implementation of the "delete" method in the "class.user.php" class is done in the following way:

a) Create the DeleteUser method
b) Call the connect method contained in the Connection class
c) Create a SQL query to delete a row in the table with the DELETE command
d) Execute the query by calling the mysqli_query command and store the execution result in the result variable
e) If the execution result is TRUE, then create a successful message to delete the data; otherwise, create a failed message to delete the data.

The "view one" method in PHP is commonly used to retrieve and display single data from a data source, such as displaying the details of a single record in a database table. In the context of a DAO implementation, the "view one" method can be used to get user details from the "user-id" table based on the user ID. The implementation of the "view one" method in the "class.user.php" class is done in the following way:

a) Create the ViewOneUser method
b) Call the connect method contained in the Connection class
c) Create a SQL query to retrieve the contents of a row in the table with the SELECT command
d) Execute the query by calling the mysqli_query command to save the execution results in the result variable
e) Insert the query result data into the object using mysqli_fetch_assoc method.

The "view all" method in PHP is commonly used to retrieve and display all data from a data source, such as displaying all records in a database table. In the context of a DAO implementation, the "view one" method can be used to get user details from the "user-id" table. The implementation of the "view all" method in the "class.user.php" class is as follows:

a. Create the ViewUser method
b. Call the connect method found in the Connection class
c. Create a SQL query to retrieve all row contents in the table with the SELECT command
d. Execute the query by calling the mysqli_query command to save the execution results in the result variable
e. If there is data in the result variable, loop through all existing data
f. Insert the query result data into the data object array using the mysqli_fetch_array method.

3.2 System Implementation Results

The system was created using PHP programming language and MySQL database. The display of the system implementation results for the add data feature can be seen in Figure 4. The form consists of data input for name, email, password, and role.

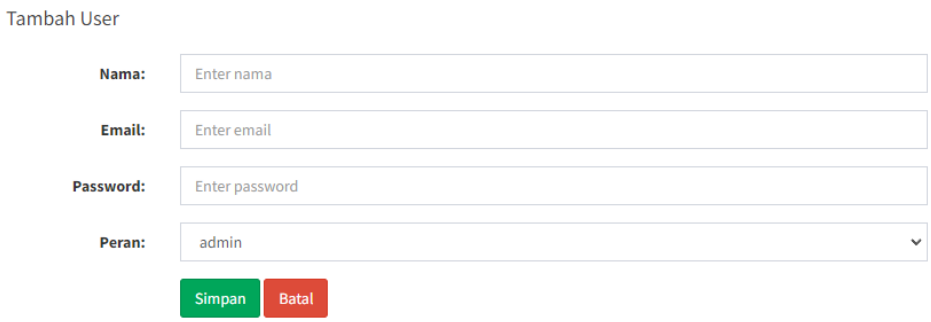

Figure 4. Add Data Display

Moreover, the system implementation results for the data change feature can be seen in Figure 5. The data changed is the name, email, password, and user role.

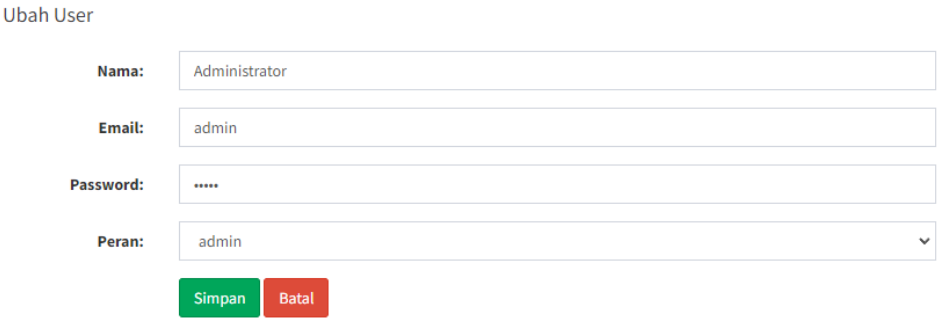

Figure 5. Change Data View

Moreover, the system implementation results for the data viewing feature can be seen in Figure 6. The data displayed are userid, name, email, and role, with action columns to change and delete data in each row.

No.	User Id.	Nama	Email	Peran	Aksi
1	1	Administrator	admin	admin	Ubah | Hapus
2	2	Pemilik Hena Catering	owner	owner	Ubah | Hapus
3	25	zul	zulfikri.maulana16@gmail.com	admin	Ubah | Hapus

Figure 6. View All Data

3.3 DAO Testing

DAO testing is carried out using unit testing. Unit testing on DAO (Data Access Object) refers to testing the operation and functionality of DAO in applications. DAO is responsible for accessing and manipulating data from data sources such as databases.

3.3.1 Testing Stages

1) Create a test class for each class you want to test. For example, to test the 'Transaction' class, create a test class 'TransactionTest'. Within the test class, define test methods that test the various features and methods within the class.
2) Within the test method, create scenarios that cover various situations and conditions in a catering bookkeeping system.
3) Prepare appropriate test data for each test case.
4) Run unit tests using PHPUnit. Run the vendor/bin/phpunit command through the terminal and view the results.
5) Check the test results to see if all test methods are successful or if there are failures. If there are failures, analyze the error messages and check the code to fix the issues found.
6) Keep repeating tests and improvements while developing the catering bookkeeping system. Make sure to test a variety of test cases, including successful cases as well as unexpected situations.

3.3.2 Testing Results

Unit testing using PHPUnit on PHP is an approach to testing each program unit in isolation, including order items, categories, customers, orders, roles, and users. Table 1 is the result of unit testing using PHPUnit for each component mentioned.

Table 1. Unit Testing Results

ID	Method	Test Data	Expected Result	Actual Result	Status
1	TambahItemPesanan()	nama_item = 'Test Tambah Item harga_jual = '10000' harga_modal = '5000'	OK	OK	PASS
2	UbahItemPesanan()	nama_item = 'Test Tambah Item harga_jual = '10000' harga_modal = '6000'	OK	OK	PASS
3	HapusItemPesanan()	Id_item_pesanan = '5'	OK	OK	PASS
4	LihatSatuPesanan()	Id_item_pesanan = '1'	OK	OK	PASS
5	LihatSemuaItemPesanan()	$obj -> LihatSemuaItemPesanan()	OK	OK	PASS
6	TambahKategori()	nama_kategori = 'Test Tambah' jenis = 'pengeluaran'	OK	OK	PASS
7	UbahKategori()	id_kategori = '5' nama_kategori = 'TestLagi' jenis = 'Pengeluaran'	OK	FAIL	FAIL
8	HapusKategori()	id_kategori = '5'	OK	OK	PASS
9	LihatSatuKategori()	id_kategori = '1'	OK	OK	PASS
10	LihatSemuaKategori()	$obj -> LihatSemuaKategori()	OK	OK	PASS
11	TambahPelanggan()	nama_instansi = 'Pelayanan' alamat = 'Margonda' nama_cp = 'Pragos' no_hp = '0821'	OK	OK	PASS
12	UbahPelanggan()	nama_instansi = 'Pelayanan' alamat = 'Margonda' nama_cp = 'Pragos' no_hp = '08211'	OK	OK	PASS
13	HapusPelanggan()	id_pelanggan = '5'	OK	OK	PASS
14	LihatSatuPelanggan()	id_pelanggan = '1'	OK	OK	PASS
15	LihatSemuaPelanggan()	$obj -> LihatSemuaPelanggan()	OK	OK	PASS
16	TambahPesanan()	tanggal_pesanan = '12-12-2002' id_pelanggan = '1'	OK	OK	PASS

ID	Method	Test Data	Expected Result	Actual Result	Status
		alamat_pelanggan = 'Depok' nama_pelanggan = 'Alya' no_hp = '0821' id_item_pesanan = '2' jumlah_pesanan = '20' catatan = 'HOHO' subtotal_pesanan = '1234' pajak_pesanan = '12' total_pesanan = '12345' tanggal_pesanan = '12-12-2003' id_pelanggan = '1'			
17	UbahPesanan()	alamat_pelanggan = 'Depok' nama_pelanggan = 'Alya' no_hp = '0821' id_item_pesanan = '2' jumlah_pesanan = '20' catatan = 'HOHO' subtotal_pesanan = '1234' pajak_pesanan = '12' total_pesanan = '12345'	OK	OK	PASS
18	HapusPelanggan()	id_pesanan = ‘5’	OK	OK	PASS
19	LihatSatuPesanan()	id_pesanan = ‘1’	OK	OK	PASS
20	LihatSemuaPesanan()	$obj -> LihatSemuaPesanan()	OK	OK	PASS
21	LihatSemuaRole()	$obj -> LihatSemuaRole()	OK	OK	PASS
22	TambahUser()	nama = 'Alya' email = 'AlyaGmail' password = 'qweasd' id_role = '3'	OK	OK	PASS
23	UbahUser()	userid = '3' nama = 'Alyaa' email = 'AlyaGmail' password = 'qweasd' id_role = '3' userid = '3'	OK	OK	PASS
24	HapusUser()	userid = ‘5’	OK	OK	PASS
25	ValidateEmail()	inputemail = ‘AlyaGmail’	OK	OK	PASS
26	LihatSatuUser()	userid = ‘1’	OK	OK	PASS
27	LihatSemuaUser()	$obj -> LihatSemuaUser()	OK	OK	PASS

Unit testing shows that most of the features in the system have successfully passed the test and run according to predetermined specifications. However, one feature did not pass the test and failed. The failure indicates that there is a problem in the implementation of the feature that needs to be fixed. Through the results of this test, the development team can identify areas that require more attention to make improvements and ensure that the entire system can function properly and as expected. The failed method test result is on the ChangeCategory method. The change category method is a method to change an order category in the catering opening system, but this method cannot be executed due to an error from the front end. To anticipate this, the user can delete the category you want to change, then create a new category you want to create.

4. Conclusions

Based on the results of the analysis and experiments that have been carried out, it can be concluded that the DAO pattern allows a clear separation between business logic and data access, makes the code more structured, easy to test, and allows high scalability and flexibility in the catering bookkeeping system. Thus, the DAO pattern allows a clear separation between business logic and data access, makes the code more structured and easy to test, and allows high scalability and flexibility in the catering bookkeeping system.

Acknowledgments: We would like to thank all parties in the Department of Computer Science, ESQ College of Management and Computer Science, and the lecturers and supervisors who have helped the author so that this work can be completed properly.

Author contributions: The author is responsible for building Conceptualization, Methodology, analysis, investigation, data curation, writing—original draft preparation, writing—review and editing, visualization, supervision of project administration, funding acquisition, and has read and agreed to the published version of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Al Karim, H.; Nuraminah, A.; P, Pulungan. Pengembangan Whistleblowing System pada Lingkup Perguruan Tinggi. *JUISIK* **2021**, 1(3). P-ISSN: 2827-8135, E-ISSN: 2827-7953. [CrossRef]
2. Ammar, A.; Zulfikri, M. ; Hawari, S. ; Novella, T. J.; Nuraminah. A. Perancangan Dan Implementasi Catering Ordering System "Hena Catering" Menggunakan Rapid Application Development. *Seminar Nasional Mahasiswa Ilmu Komputer Dan Aplikasinya (SENAMIKA)* **2021**, April, 31–39. [CrossRef]
3. Aniche, M. F.; Oliva, G.A.; Gerosa, M. A. Are the methods in your Data Access Objects (DAOs) in the right place? A preliminary study. *Proceedings - 2014 6th IEEE International Workshop on Managing Technical Debt, MTD* **2014**, 47–50. [CrossRef]
4. Aulia Aziiza, A., & Fadhilah, A. N. Analisis Metode Identifikasi dan Verifikasi Kebutuhan Non Fungsional. Applied Technology and Computing Science Journal **2020**, 3(1). [CrossRef]
5. Connoly, T., & Begg, C. Database System, A Practical Approach to Design, Implementation and Management Sixth Edition. **2019** Pearson Education Limited.
6. Dodi; Wardah, E.; F.A, Maki. Pengembangan Media Pembelajaran pada Pemrograman Terstruktur dan Pemrograman Berorientasi Objek dengan Visualisasi Bangun Datar Menggunakan Processing. *A Journal of Language, Literature, Culture, and Education POLYGLOT*, **2018**, Vol.14 (No.1). [CrossRef]
7. Fajar, M.; Ciuandi, F.; Munir, A; Informatika, T.; Kharisma Makassar, S. Desain Aplikasi Daily Remainder Menggunakan Model-View Controller Dan Data Access Object Daily Remainder Application Design Using Model-View Controller and Data Access Object. *In JTSI* 4, 2, **2023**. [CrossRef]

8. Hakim. L.; Rochimah.S.; Fatichah. C. Klasifikasi Kebutuhan Non-Fungsional Menggunakan FSKNN berbasis ISO/IEC 25010. *JUTI: Jurnal Ilmiah Teknologi Informasi,* 2019, 17(2), 107 [CrossRef]
9. Hidayah, A.; Yani, A.; Studi Sistem Informasi, P.; Mahakarya, S. Membangun Website SMA PGRI Gunung Raya Ranau menggunakan PHP dan MySQL, *Jurnal Teknik Informatika Mahakarya* 2019, 2, 2. [CrossRef]
10. Indra Andhika, D.; Muharrom, M.; Prayitno, E.; Siregar, J. Rancang Bangun Sistem Penerimaan Dokumen pada PT. Reasuransi Indonesia Utama. *Jurnal Informatika dan Teknologi Komputer* 2022, 2(2), 136–145.
11. Jánki, Z. R., & Bilicki, V. (2023). The Impact of the Web Data Access Object (WebDAO) Design Pattern on Productivity. *Computers,* 12(8), 149. [CrossRef]
12. Khairunnisa, & Rismayanti. Perancangan Intelligent Tutoring System Sebagai Upaya Inovatif Pada Pembelajaran Pemrograman Terstruktur. *QUERY: Jurnal Sistem Informasi.* 2020, 4, 1 [CrossRef]
13. Kouli, M., & Rasoolzadegan, A. A Feature-Based Method for Detecting Design Patterns in Source Code. *Symmetry* 2022, *14(7), 1491.* [CrossRef]
14. Matic, D.; Butorac, D., ; Kegalj, H. Data Access Architecture in Object Oriented Applications Using Design Patterns. *Proceedings of the 12th IEEE Mediterranean Electrotechnical Conference (IEEE Cat. No.04CH37521)* 2014, 595–598. [CrossRef]
15. Nabila, S.; Putri, A. R.; Hafizhah,A.; Rahmah, F. H.; Muslikhah, R. Pemodelan Diagram UML Pada Perancangan Sistem Aplikasi Konsultasi Hewan Peliharaan Berbasis Android (Studi Kasus: Alopet*). Jurnal Ilmu Komputer Dan Bisnis 2021*, 12(2), 130–139. [CrossRef]
16. Nock, C. (2008). Data Access Patterns: Database Interactions in Object-Oriented Applications. Addison-Wesley Professional.
17. O.-J. Dahl, E. W. Dijkstra, & C. A. R. Hoare. (2021). Structured Programming. Academic Press.
18. Olsson, M. (2021). Magic Methods. In: PHP 8 Quick Scripting Reference. Apress, Berkeley, CA.
19. Pradipta, R. A., Wintoro, P. B., & Budiyanto, D. (2022). perancangan pemodelan basis data sistem informasi secara konseptual dan logikal. *Jurnal Informatika Dan Teknik Elektro Terapan*, 2022, 10(2). [CrossRef]
20. Pramushinto, S.; Jaya, N. A. ; Azhaar, A.; Noviadih, M.; Saifudin, A. Unit Testing Pada Aplikasi Web (Studi Kasus Bisnis Jasa Laundry). *TEKNOBIS : Teknologi, Bisnis Dan Pendidikan 2023, Vol. 1(No. 1).* [CrossRef]
21. Rizkyana, M. A., Yunanto, Y., Yoga, Y., & Widianto, S. R. (2021). Implementasi Unit Testing menggunakan metode Test-First Development. *Multinetics, 2021, 7(1),* 37–47. [CrossRef]
22. Sandfreni, S.; Ulum, M. B.; Azizah, A. H. Analisis Perancangan Sistem Informasi Pusat Studi pada Fakultas Ilmu Komputer Universitas Esa Unggul. *Sebatik, 2021, 25(2), 345–356.* [CrossRef]
23. Suharni, Susilowati, E., & Pakusadewa, F. Perancangan Website Rumah Makan Ninik sebagai Media Promosi Menggunakan Unified Modelling Language. *Jurnal Rekayasa Informasi,* 2023, 12(1). [CrossRef]
24. T. J, Novella; A, Nuraminah; A.D., Goenawan. Perancangan Database dan Website Loker Rental System Pada Kampus ESQ Business School. *jurnal teknik mesin, industri, elektro dan informatika, 2022, 1(3), 111–119*. [CrossRef]
25. Turserno, A., & Rosihan, R. I. Analisis Kebutuhan Fungsional Sistem Informasi Manajemen Gudang dengan Metode Pieces (Studi Kasus CV Karya Bangsa). *Journal Industrial Manufacturing, 2022, 7(1), 01.* [CrossRef]