# Rifka-EBMTJ-TURNITIN

*by* Rifka Fitriyah

**Submission date:** 25-Sep-2023 02:35AM (UTC-0700)
**Submission ID:** 2176266565
**File name:** Hasil_Ringkasan_Rifka-eng.docx (3.61M)
**Word count:** 3723
**Character count:** 20836

**Land and Building Tax Calculation Application (E-Pbb)**
**(Case Study in One of The Villages in West Java)**

Rifka Fitriyah
STMIK Mardira Indonesia
Email: rifkafitriyah@gmail.com

**Abstract**
Introducing an efficient land and building tax computation system can have favorable outcomes for the rural community and its inhabitants. The management and authenticity of data about the distribution of Property Tax Notification Letters (SPPT) have historically posed issues within the tax management process.
The e-PBB program for calculating land and building taxes in the Tribaktimulya Village case study is implemented as a web-based platform. The application utilizes quantitative data collection methods, including observation, interviews, and literature reading. The system is developed with the Object-Oriented Analysis and Design (OOAD) methodology, which encompasses the stages of analysis, design, and system implementation centered around objects.
The design of this system is anticipated to enhance the ability of village authorities to input and compute land and building taxes that are either outstanding or paid. Automating the calculation process will enhance efficiency while simplifying tax report generation will contribute to a more effective and streamlined operational workflow.

Keywords: Application, Processing, Calculation, Design, PBB, Report, Recapitulation and Quantitative Method

## Introduction

The Land and Building Tax is a form of direct taxation imposed by the central government, with collection taking place every year. Managing the central tax lies with the Directorate General of Taxes, which operates through the Land and Building Tax Office. The generated money is allocated, with 10% allocated to the central government and the remaining 90% distributed among local governments.

The Land and Building Tax tax collecting mechanism is commonly called the Official Assessment mechanism. According to the current regulations, taxpayers are exempt from the duty to pay Land and Building Tax until they have received a tax assessment from the treasury. The assessment above was issued in the year 1994. Given the lofty objectives of this country, it is imperative to direct attention toward factors that can bolster the government's efficacy in safeguarding the well-being of the Indonesian populace. One such factor is the Land and Building Tax (PBB), which is explicitly about land situated inside the confines of Indonesia's territorial jurisdiction.

This statement highlights the importance of thoroughly assessing and appraising the efficiency of Land and Building Tax collection to attain favorable outcomes, ultimately benefiting the populace of Indonesia using streamlined tax-collecting processes. One of the strategies the government employs to accomplish this objective is the collection of taxes, as taxes serve as a crucial income source that plays a substantial role in procuring funds for government spending.

One example of an income source is the Land and Building Tax (PBB). The Land and Building Tax has the potential to be employed for a multitude of policy considerations about land and buildings. Determining the Land and Building Tax relies on the Taxable Selling Value, established within a range of 20% to 100% of the Taxable Object Selling Value (NJOP) (Arianto, 2016). The Land and Building Tax is a significant potential

revenue generator for regions, serving as a direct tax. The tax in question can be classified as a central tax due to its focus on items within specific regions, hence enabling these regions to obtain a more significant portion of the tax revenue (Soemitro & Manurung, 2018).

One of the intricate matters in Indonesia, precisely in Tribaktimulya Village, pertains to the annual escalating population growth. The rise in population substantially influences the augmentation of Land and Building Tax revenue, as viewed through a tax lens. *Taxation* is a process through which funds are gathered from community members, commonly referred to as taxpayers. The tax collected is determined by assessing the income or wages these individuals obtain in exchange for their services. The money generated from Land and Building Tax can be observed through the annual escalation in land and building prices, with the growing demand for land and buildings attributed to population expansion. Hence, the revenue generated from Land and Building Tax also demonstrates an upward trend due to the progress in diverse amenities and infrastructure, particularly in residential areas that cater to the populace's needs (Surahman, 2018).

The advantages associated with paying Land and Building Tax (PBB) for development and societal well-being are substantial, with numerous conveniences available for facilitating the payment process. Despite the convenience of payment and its tangible rewards, there remains a noticeable need for more community engagement in fulfilling their responsibilities as taxpayers of the PBB. The general perception among the community is that taxes are perceived as an onerous financial obligation, ultimately leading to a decrease in individual well-being.

Hence, by implementing a system update in tax collection and providing convenient payment options for PBB, the inhabitants of Tribaktimulya Village will actively engage in meeting this responsibility. The design of the "LAND AND BUILDING TAX CALCULATION APPLICATION (e-PBB) (CASE STUDY OF TRIBAKTIMULYA VILLAGE)" is developed by the prevailing conditions and through interviews conducted with the village personnel. There is an expectation that the acceptance of this application will yield benefits in facilitating payment processes. In order to achieve effective adoption by the initial objectives, it will be imperative to provide adequate guidance on the functionality and operation of the application during its implementation phase.

**Method**

*Descriptive research* is an investigation that aims to ascertain the magnitude or characteristics of independent variables, either alone or in combination, without establishing comparisons or associations with other variables (Nana & Elin, 2018).

The interview process involves gathering information through a structured conversation with an individual or group to gain insights or obtain data.

In order to gather comprehensive information for this Final Project, the author employed a question-and-answer methodology to explore various aspects of the Village. This encompassed examining the Village profile, reviewing Village reports, assessing the presence of non-computerized systems, and considering the systems the author intends to develop.

The act of observing.

To identify potential solutions, the researcher conducted firsthand observations of the local environment in the Village that pertained to the topic or issue under investigation.

The topic of interest for this investigation is field study.

Engaging in many performance activities organized by a village in West Java presents an avenue for acquiring additional knowledge.

The present section comprehensively reviews the existing literature on the subject matter.

In conjunction with conducting interviews and making observations, the author also undertook a comprehensive literature review by consulting references available in libraries

and online sources. This approach was employed to augment understanding of the subject matter, various classifications, and methodologies of tax calculations.

**System Development Methods**

Object Oriented Programming (OOP) is the approach employed for creating systems. Object Oriented Programming (OOP) is a software development methodology that arranges software components as a cohesive ensemble of objects, each encompassing data and the corresponding actions that may be performed on that data. Object-oriented programming (OOP) is a programming paradigm that emphasizes objects and their interactions.

Based on the current body of knowledge, Object Oriented Programming (OOP) is a contemporary methodology or paradigm for software development or system design that considers the concept of objects. Object-oriented methodologies are frequently preferred due to the inherent issues associated with older methodologies. These issues include seamlessly transitioning outcomes from one development stage to the subsequent stage. For instance, in the structured approach method, the nature of applications developed in contemporary times diverges from those developed in the past.

**Result and Discussion**

**Sequence Diagrams**

A Sequence Diagram is a diagram that explains object interactions and shows (gives signs or instructions) communication between these objects.

1. Login Sequence Diagram
   The image below explains the flow of an action that has been designed in a sequence diagram, namely the action in the form of logging in (the process of accessing the system). It can be seen as follows:

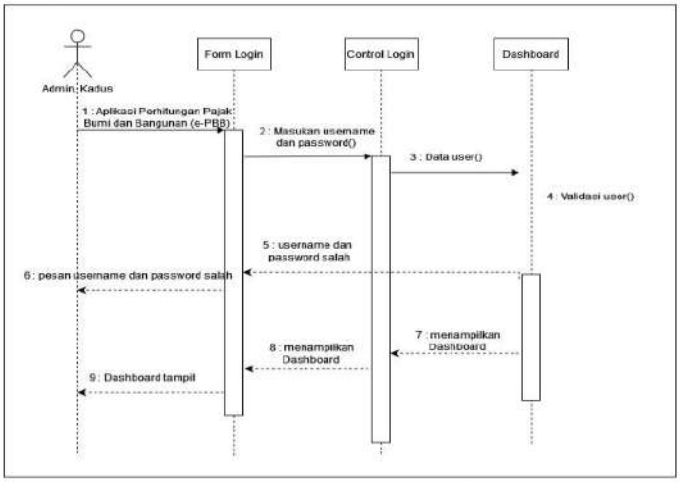


Figure 1. Login Sequence Diagram

2. Sequence Diagram for Managing SPPT Data
   The image below explains the flow of an action that has been designed in a sequence diagram, namely the action of managing SPPT data. It can be seen as follows:

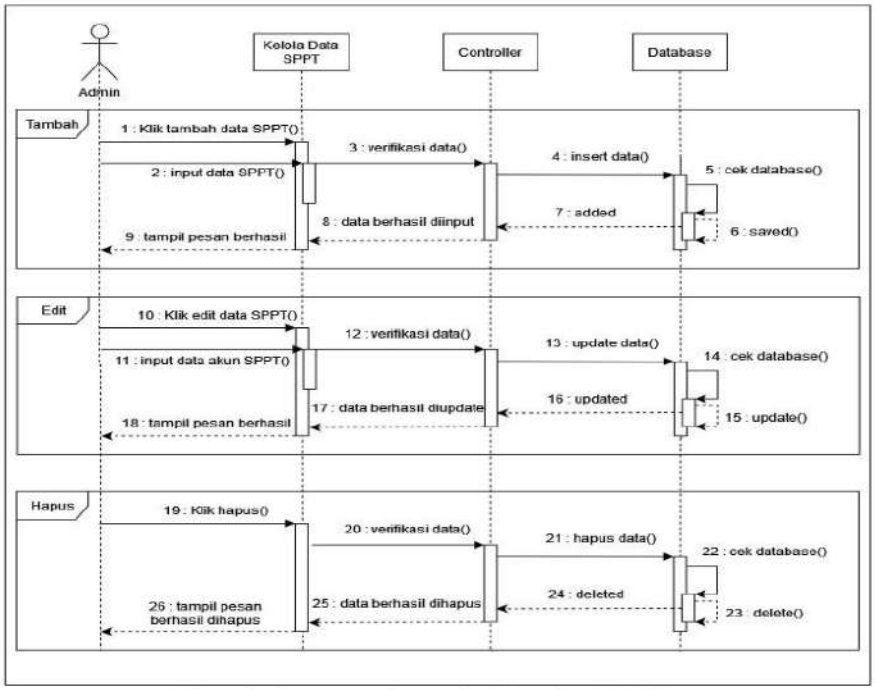


Figure 2. Sequence Diagram for Managing SPPT Data

3. Diagram for Managing Officer Data

The image below explains the flow of an action that has been designed in a sequence diagram, namely the action in the form of displaying data on related officers. It can be seen as follows:

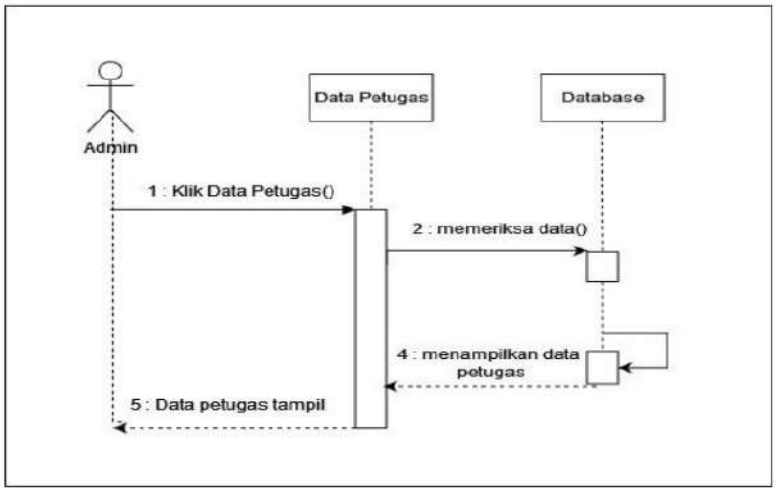


Figure 3. Sequence Diagram for Managing Officer Data

4. Sequence Diagram Managing Profile Data

The image below explains the flow of an action that has been designed in a sequence diagram, namely the action of managing job data. It can be seen as follows:

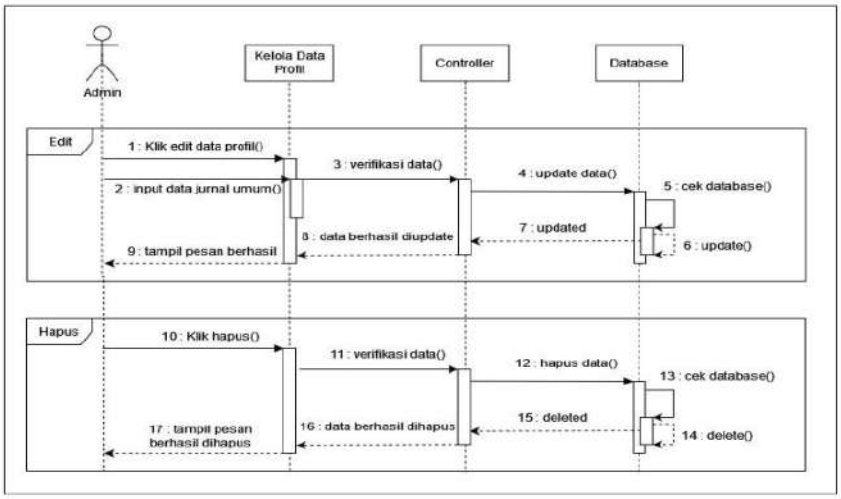


Figure 4. Sequence Diagram for Managing Profile Data

5. Sequence Diagram for Managing Outstanding PBB Data

The image below explains the flow of an action that has been designed in a sequence diagram, namely the action of managing outstanding PBB data. It can be seen as follows:

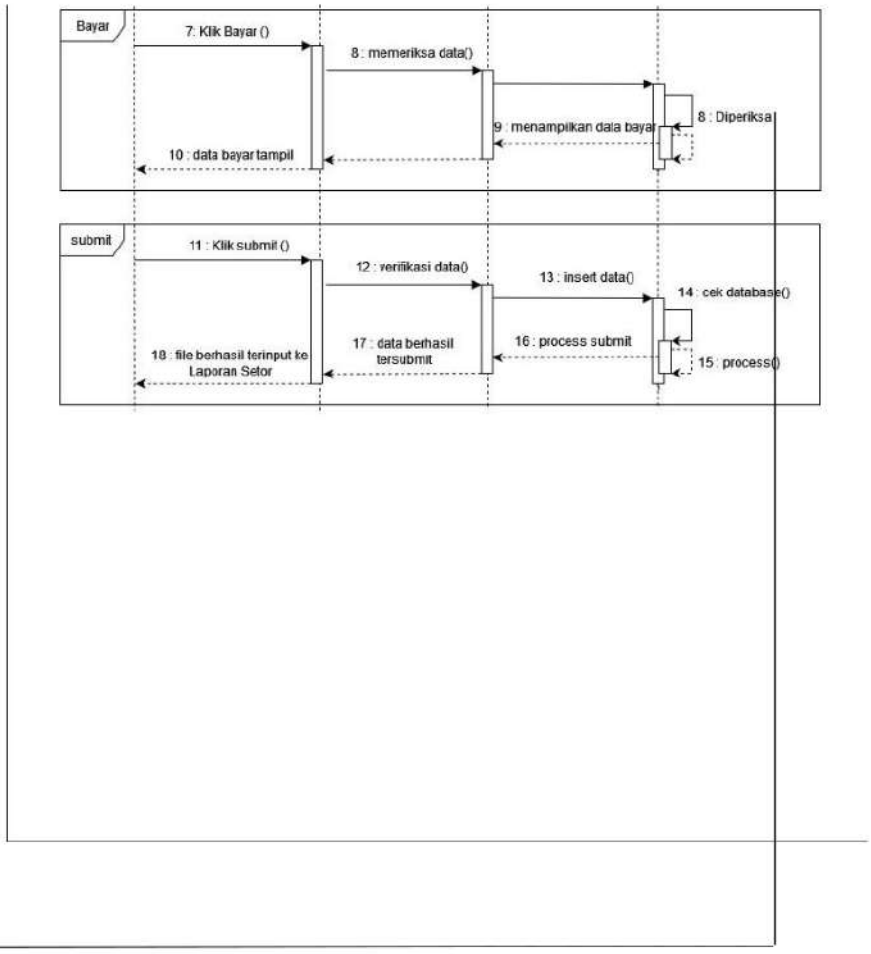

Figure 5. Sequence Diagram for Managing Outstanding PBB Data

6. Sequence Diagram for Managing PBB Deposits
   The image below explains the flow of an action that has been designed in a sequence diagram, namely the action of managing PBB deposits. It can be seen as follows:

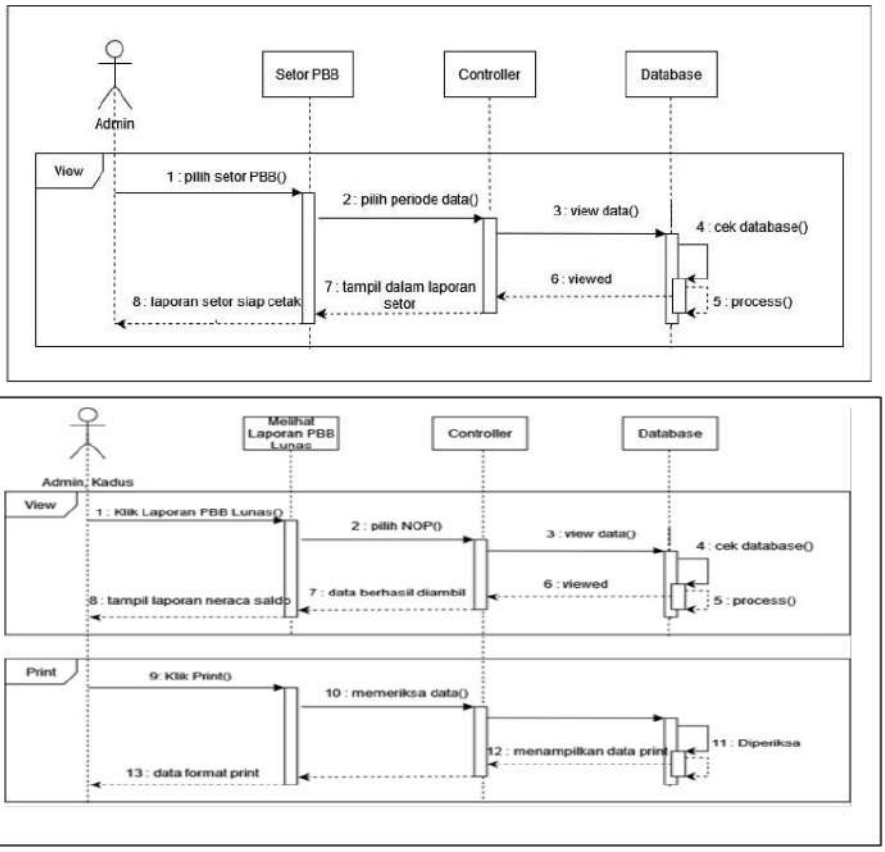


Figure 6. Sequence Diagram for Managing PBB Deposits

7. Sequence Diagram for Managing UN Recapitulation
   The picture below explains the flow of an action that has been designed in a sequence diagram, namely the action of managing the UN recapitulation. It can be seen as follows:

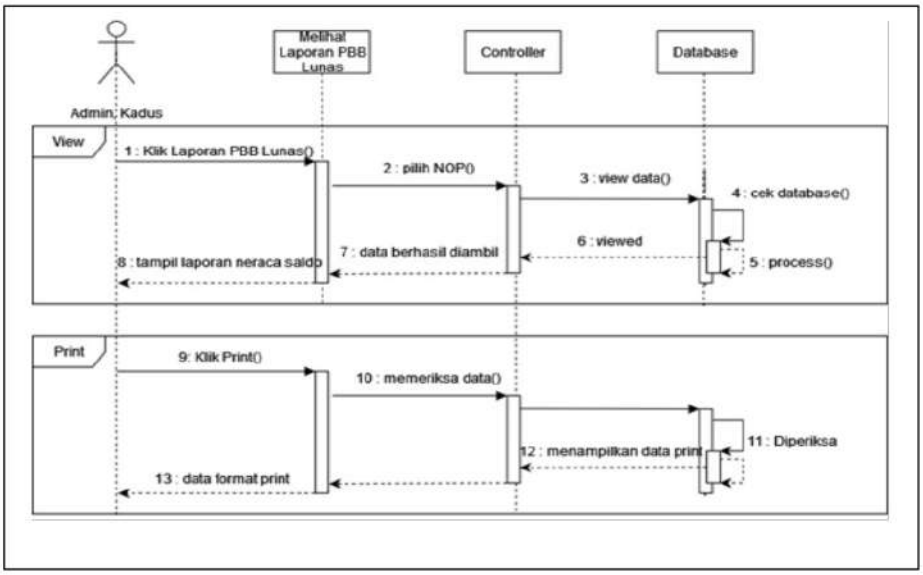


Figure 7. Sequence Diagram for Managing UN Recapitulation

8. Sequence Diagram Viewing Report
   image below explains the flow of an action that has been designed in a sequence diagram, namely the action of viewing a report. It can be seen as follows:

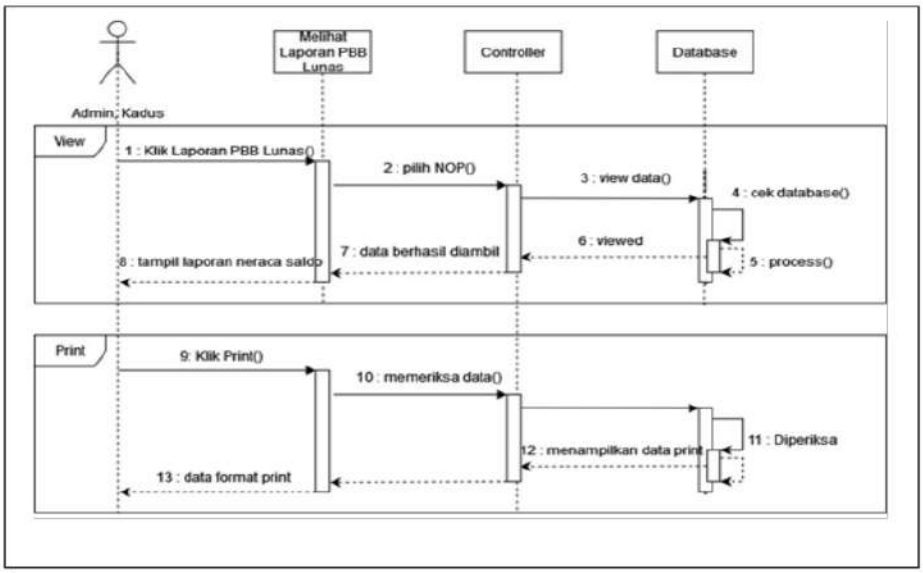


Figure 8. Sequence Diagram Viewing Reports

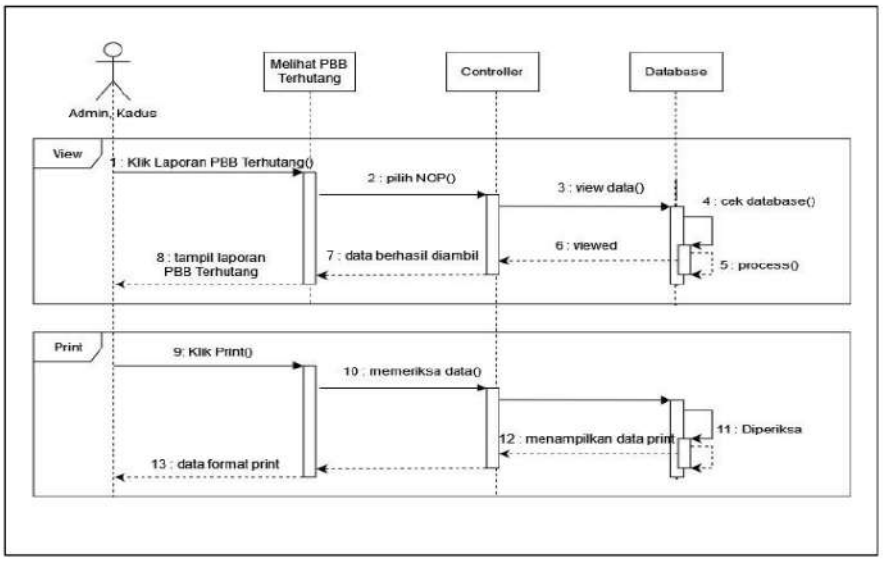


Figure 9. Sequence Diagram Viewing Reports

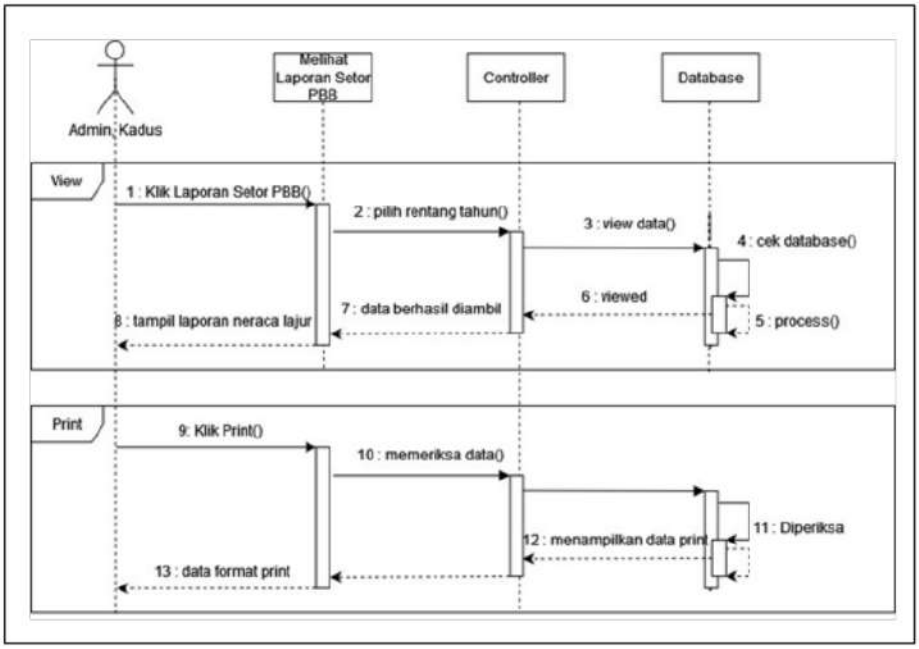


Figure 10. Sequence Diagram Viewing Reports

9. Logout Sequence Diagram

The image below explains the flow of an action that has been designed in a sequence diagram, namely the action in the form of logging out (the system access process). It can be seen as follows:

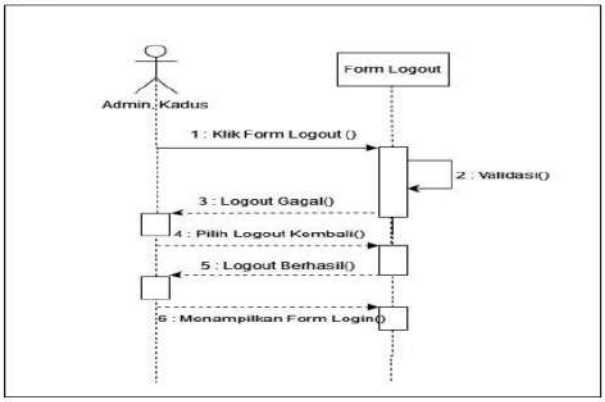


Figure 11. Logout Sequence Diagram

**Class *Diagrams***

Class diagrams are utilized to depict the structure of classes, including their attributes and relationships, within the software system under development. These graphs exhibit interconnections or associations between different classes. The subsequent picture illustrates the suggested class diagram for the building tax calculation system.

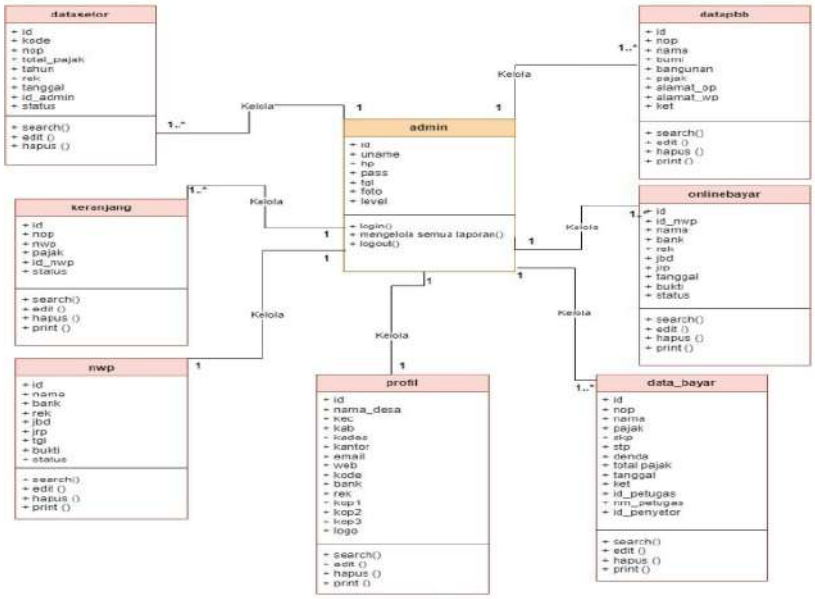


Figure 12. Class Diagram

**Database Design**

Database Design In making this program, the database used is the MySQL database, which is SQL database management system software. This software functions as a database query maker and so on. MySQL is a relation database management system (RDBMS) distributed free of charge under the GPL (General Public License).

This database is suitable because it can run in various information systems.
The following are table designs from the property tax database:

1. Table tb_admin

Table 1. tb_admin

| No | Field | Type | Primary |
|---|---|---|---|
| 1 | id | int (11) | |
| 2 | uname | varchar (30) | |
| 3 | hp | varchar (100) | |
| 4 | pass | varchar (70) | |
| 5 | date | Date | |
| 6 | picture | varchar (150) | |
| 7 | level | varchar (15) | |

2. Tabel tb_ pay data

Table 2. tb_ pay data

| No | Field | Type | Primary |
|---|---|---|---|
| 1 | id | int (11) | |
| 2 | nop | varchar (20) | |
| 3 | name | varchar (100) | |
| 4 | tax | int (12) | |
| 5 | skp | int(12) | |
| 6 | stp | int (12) | |
| 7 | fine | int(12) | |
| 8 | total_tax | int (12) | |
| 9 | date | Date | |
| 10 | information | varchar (100) | |
| 11 | id_officer | int (4) | |
| 12 | name_officer | varchar (100) | |

| | | | |
|---|---|---|---|
| 13 | id_ depositor | int (4) | |

3. Table tb_ UN data

Table 3. tb_ UN data

| No | Field | Type | Primary |
|---|---|---|---|
| 1 | id | int (11) | |
| 2 | nop | varchar (20) | |
| 3 | name | varchar (100) | |
| 4 | earth | varchar (5) | |
| 5 | building | varchar (5) | |
| 6 | tax | int (12) | |
| 7 | address _op | varchar (200) | |
| 8 | address _wp | varchar (200) | |
| 9 | information | varchar (50) | |

4. Table tb_ seto data

Table 4. tb_ seto data

| No | Field | Type | Primary |
|---|---|---|---|
| 1 | id | int (11) | |
| 2 | code | varchar (4) | |
| 3 | nop | varchar (20) | |
| 4 | total_tax | int (12) | |
| 5 | year | varchar (2) | |
| 6 | rek | varchar (20) | |
| 7 | date | Date | |
| 8 | id_admin | int (4) | |
| 9 | status | int (2) | |

5. Table tb_ basket

Table 5. tb_ basket

| No | Field | Type | Primary |
|---|---|---|---|
| 1 | id | int (4) | |
| 2 | nop | varchar (17) | |
| 3 | nwp | varchar (50) | |
| 4 | pajak | int (6) | |
| 5 | id_wp | int (4) | |
| 6 | status | varchar (15) | |

6. Table tb_nwp

Table 6. tb_nwp

| No | Field | Type | Primary |
|---|---|---|---|
| 1 | id | int (4) | |
| 2 | name | varchar (150) | |
| 3 | bank | varchar (125) | |
| 4 | rek | varchar (25) | |
| 5 | adress | varchar (250) | |
| 6 | pass | varchar (100) | |
| 7 | date | Date | |
| 8 | picture | varchar (2000) | |
| 9 | level | varchar (15) | |

7. Table tb_ pay online

Table 7. Tabel tb_ pay online

| No | Field | Type | Primary |
|---|---|---|---|
| 1 | id | int (4) | |
| 2 | id_nwp | int (4) | |
| 3 | name | varchar (50) | |
| 4 | rek | varchar (25) | |
| 5 | jbd | int (4) | |
| 6 | jrp | int (6) | |
| 7 | date | Date | |

| | | | |
|---|---|---|---|
| 8 | proof | varchar (200) | |
| 9 | status | varchar (15) | |

8. Table tb_profile

Table 8. tb_profile

| No | Field | Type | Primary |
|---|---|---|---|
| 1 | id | int (4) | |
| 2 | name_ desa | varchar (50) | |
| 3 | kec | varchar (50) | |
| 4 | kab | varchar (75) | |
| 5 | kades | varchar (45) | |
| 6 | office | varchar (250) | |
| 7 | email | varchar (150) | |
| 8 | web | varchar (150) | |
| 9 | code | varchar (4) | |
| 10 | bank | varchar (50) | |
| 11 | rek | varchar (18) | |
| 12 | Kop1 | varchar (200) | |
| 13 | Kop2 | varchar (200) | |
| 14 | Kop3 | varchar (200) | |
| 15 | Kop4 | varchar (200) | |

**Menu Structure**

The menu structure design that will be created is as shown in the image below:

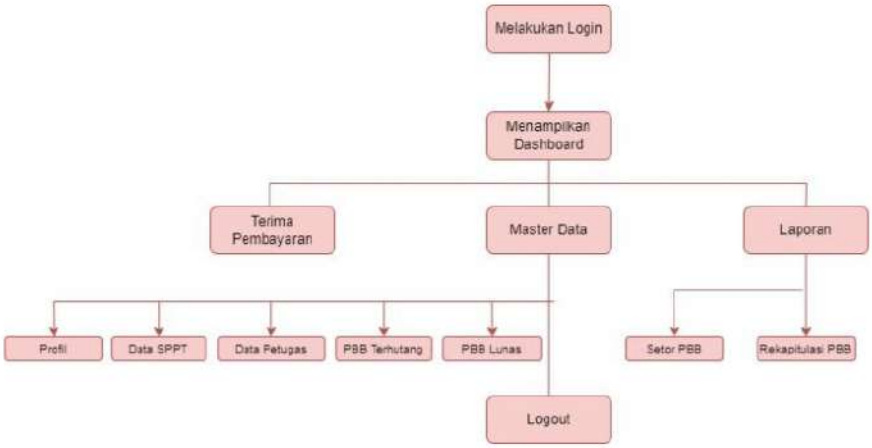


Figure 13. Admin Menu Structure

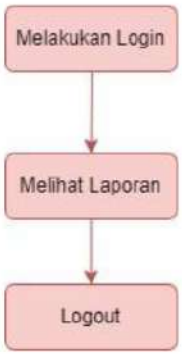


Figure 14. Kadus Menu Structure

In designing this interface, it will describe the display design of what is in the system. The following are the interface design views:

1. Login Form Display


Figure 15. Login Form Display

2. Dashboard View


Figure 16. Dashboard display

3. Display of the Receive Payment Data Form

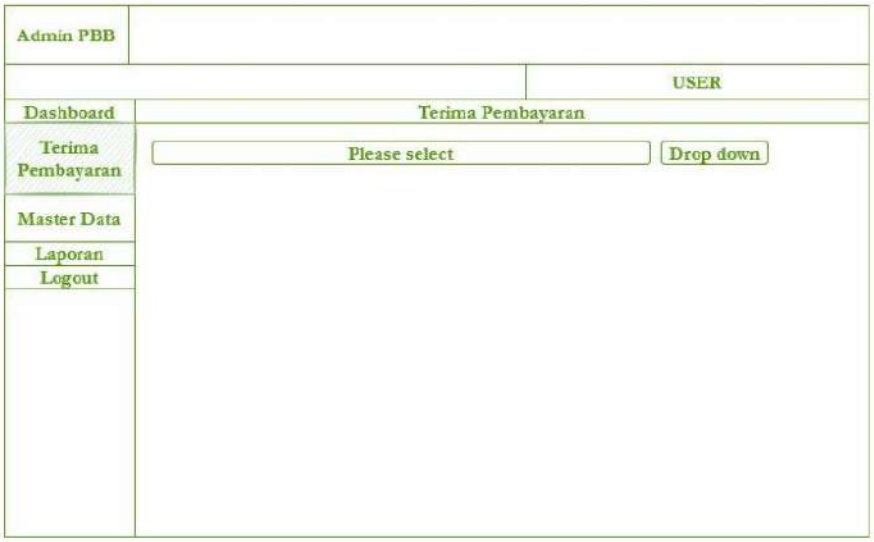


Figure 17. Display of the Receive Payment Data Form

4. Display Master Data Form

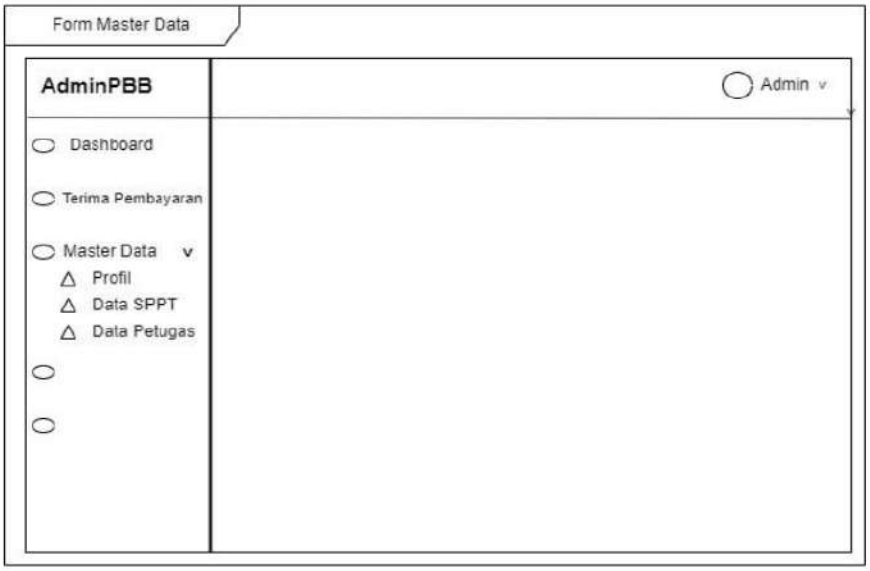


Figure 18. Master Data Form Display

5. Report Form Display

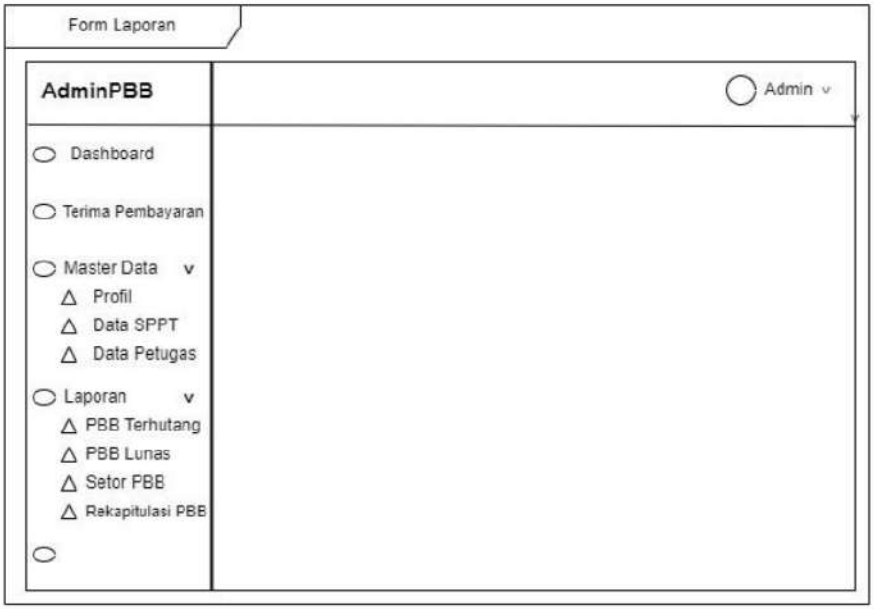


Figure 19. Report Form Display

6. Logout Form Display

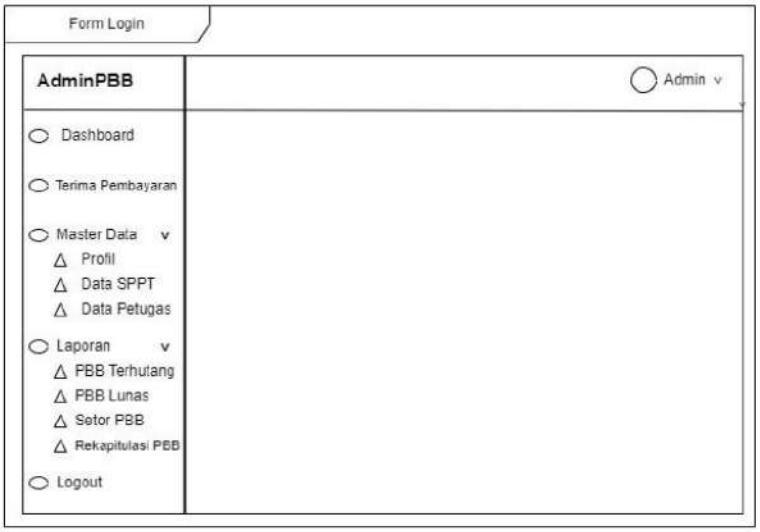


Figure 20. Logout Form Display

## SYSTEM IMPLEMENTATION

The etymological derivation of the term "implementation" pertains to executing, putting into effect, and carrying out a specific action or plan. In the realm of terminology, implementation refers to the process of executing activities and actions in order to accomplish a predetermined goal. Implementation refers to applying a system to be executed after thorough study and strategic planning.

### Implementation Needs

Once the system's thorough analysis and detailed design have been completed, it will progress to the subsequent implementation stage. The implementation phase involves deploying the system ensuring its readiness for operational use. The primary objective of implementation is to validate the design modules, thereby enabling users to contribute their insights to the system's creators. The suggested scope of implementation encompasses both software and hardware environments. The prerequisites for application design serve as the primary foundation to ensure that the design aligns with pre-established objectives. The recommended breadth of implementation encompasses both software and hardware environments.

### Hardware

The hardware needed to support the implementation of a computerized system. Hardware applications that can support the system applications that will be created include:

1) Minimum dual core 94 processor

2) RAM 2048 MB

3) Space hard disk 10 GB

4) Monitor, keyboard. dan mouse

### Software

The software used to run the system created. In this case, the software used by the author to apply this system:

1) Windows 10

2) Bahasa pemograman PHP

3) Database My SQL

4) XAMPP

5) Web server

**Brainware**

Brainware is one of the important components in implementing a system and is usually in the form of human objects as the implementing staff consisting of:

1) Admin, namely the person in charge of operating the system and carrying out the process of entering data, changing data or deleting data.

**Selection of Programming Language**

PHP is an open-source programming language. The availability of PHP open-source code at no cost facilitates continuous improvement, development, and problem identification among the developer community. The concept of open source entails the absence of reliance on a specific corporate entity, such as Windows or Microsoft, to release subsequent versions in case of unforeseen complications. Additionally, there is no requirement to incur expenses for purchase and upgrade costs, typically of a considerable magnitude. The concepts of stability and compatibility are paramount in various academic disciplines. Currently, PHP demonstrates stable performance across multiple operating systems, encompassing various iterations of UNIX (including Linux), Windows, and MacOS. PHP is compatible with various web servers, notably the widely utilized IIS and Apache servers. PHP is additionally furnished with diverse supplementary functionalities, including direct support for multiple prominent databases, an adaptable architecture, and a processor that consumes minimum resources on the host system compared to its counterparts and exhibits rapid rendering of web pages.

**Program Testing**

System testing is a testing process to find out where the deficiencies in the system are and whether they are in accordance with what was described in the previous design. Testing of this system is carried out carefully so that the results obtained can provide benefits to users.

**Maintenace**

During the execution of application programs, modifications to the application frequently transpire and, in some instances, result in detrimental effects. This occurrence may arise due to a programming malfunction or a user-initiated request. Maintenance activities are essential to ensure the program's stability and equilibrium, tailored to the specific environment in which it is utilized. These maintenance efforts aim to sustain the program's ability to generate the necessary information for the user.

**Implementation Activities**

*Programming* is a highly beneficial endeavor that significantly facilitates the successful implementation of novel systems. This is primarily because a well-designed and organized program can generate information by specific requirements and demands. Before implementing the program, it is imperative to do thorough testing to ensure its error-free functionality.

The testing process can be conducted individually for each program module, followed by comprehensive testing of all modules to verify the proper and accurate integration of all modules. This part describes the visual interface of the application software, which has been developed based on the design specifications established during the system design phase. The author has categorized the present system design into multiple display forms.

**Database**

The database table structure display is part of the implementation of the tables in the database that will be accessed by the user.

1. Admin_Database

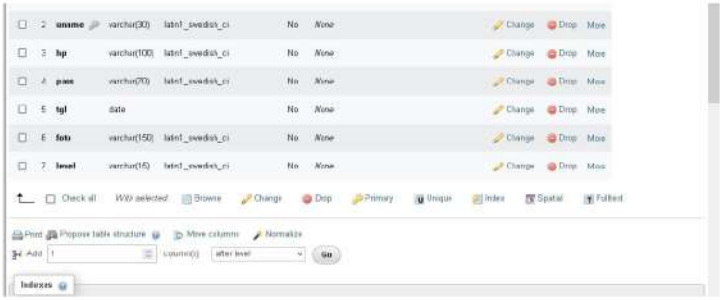

Figure 21. Admin Database

2. Pay Data_Database

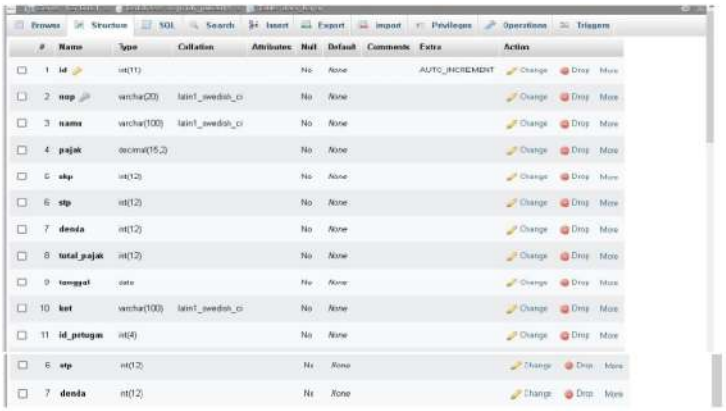

Figure 22. Payment data database

Figure 23. Payment data database

3. Data_setor database

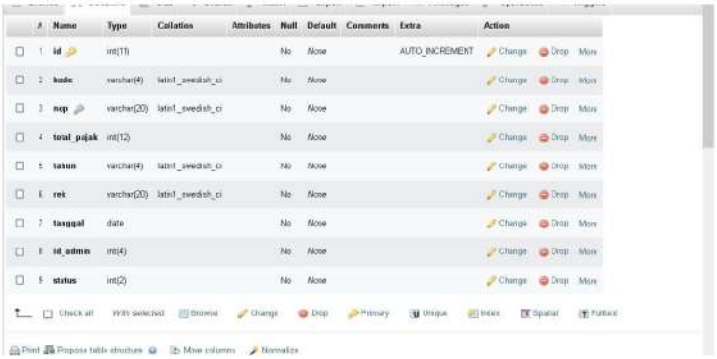


Figure 24. Data_setor database

4. Database Data_cart

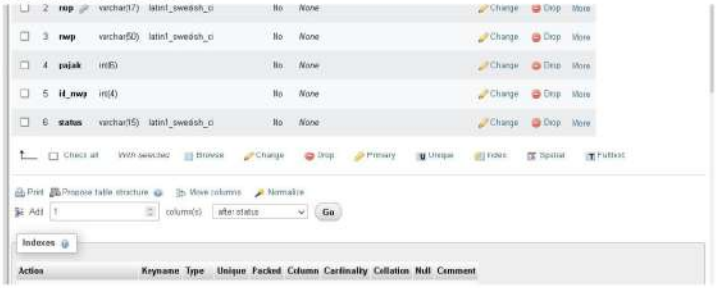


Figure 25. Database Data_cart

5. Database Data_nwp

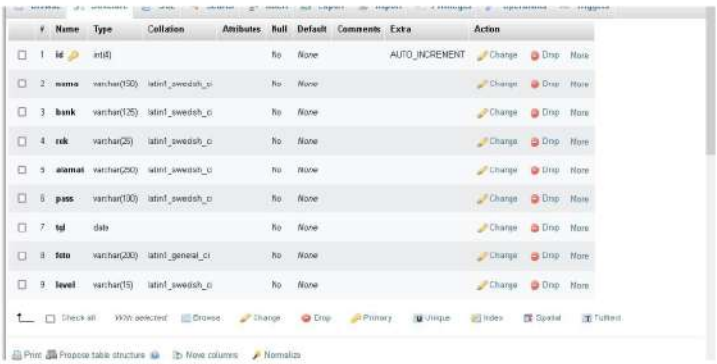


Figure 26. Data_nwp database

6. Database online_bayar

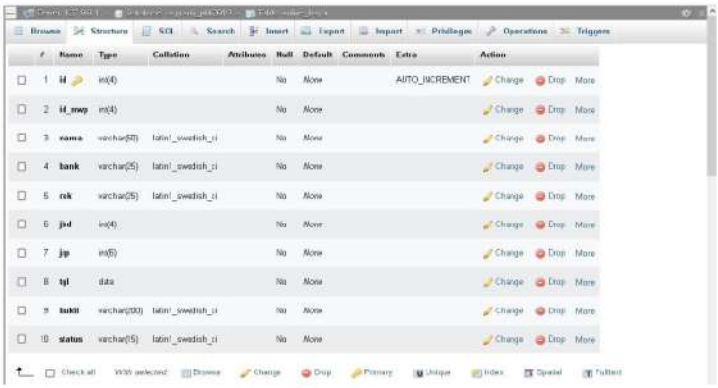

Figure 27. Online_paid database

7. Profile Database

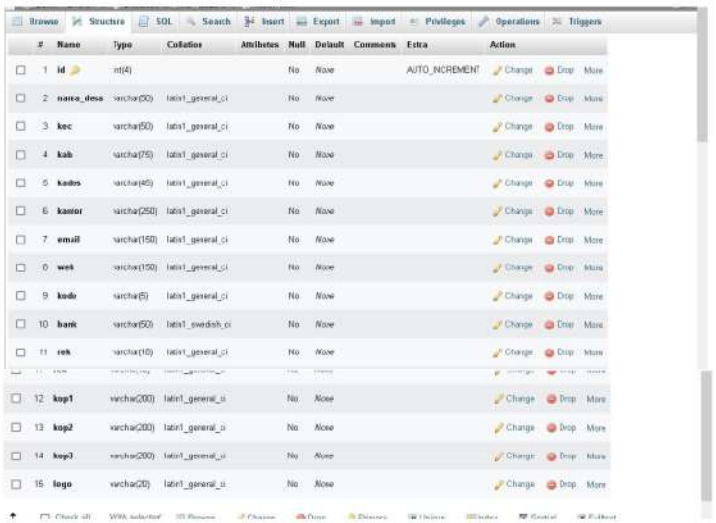

Figure 28. Profile database

**Homepage**

Upon initiating the software, the user will be presented with the initial menu display, encompassing the following options: Home, input, process, report, and exit. The initial presentation will exhibit the system's title, the company's name, and the emblem. The

subsequent output is the visual representation generated by the application program.


Figure 29. Login Display

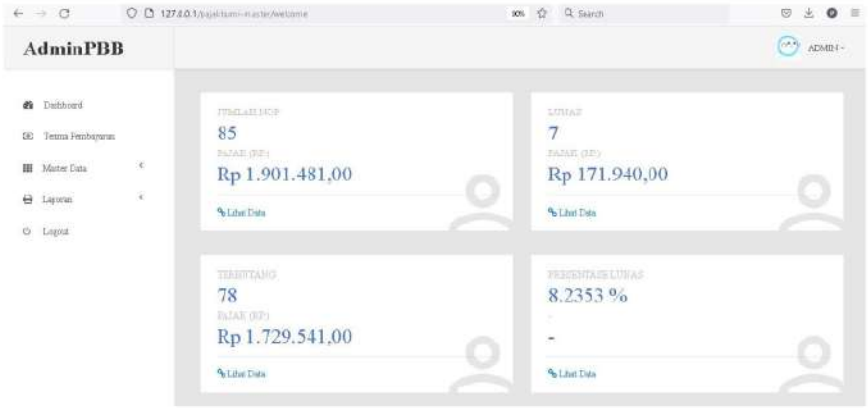


Figure 30. Dashboard display

**Input Page**

The concept of input design refers to creating a structured format, typically in the form of a form, that facilitates data entry. The design and implementation of input/output systems serve as a valuable resource for developers in creating and constructing applications. Input design refers to designing the mechanism by which users provide information to a system, which is subsequently saved in a database.

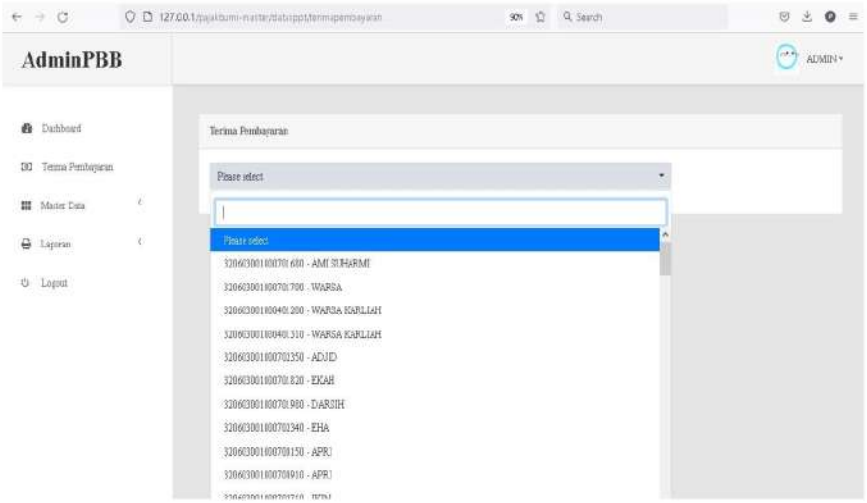


Figure 31. Receive Payment display

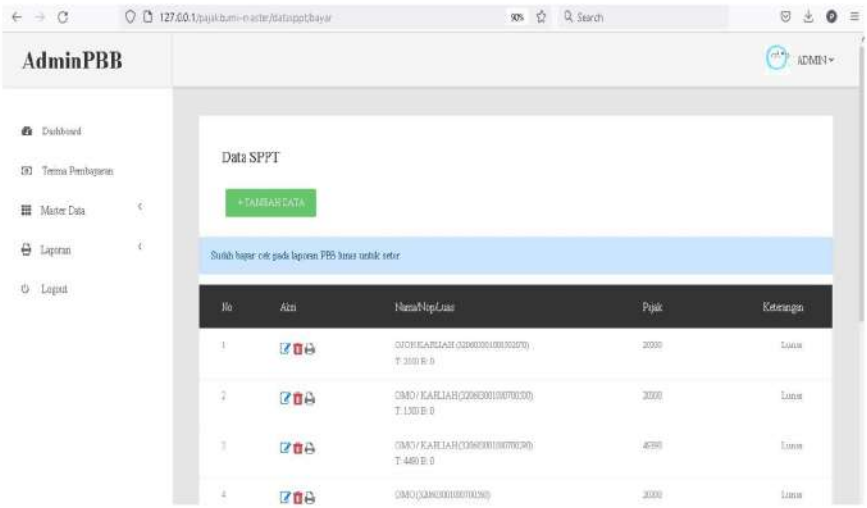


Figure 32. Successful Data Input Display

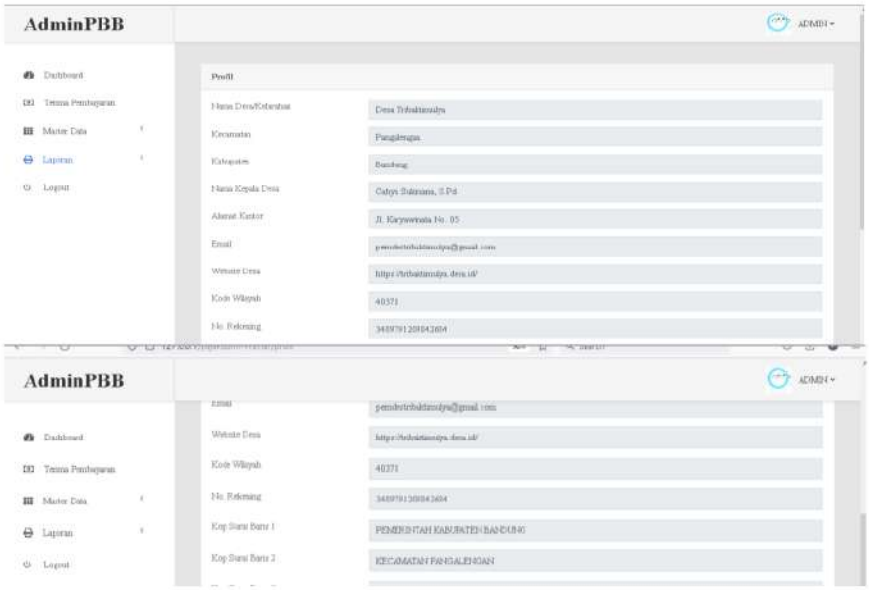


Figure 33. Village Profile View

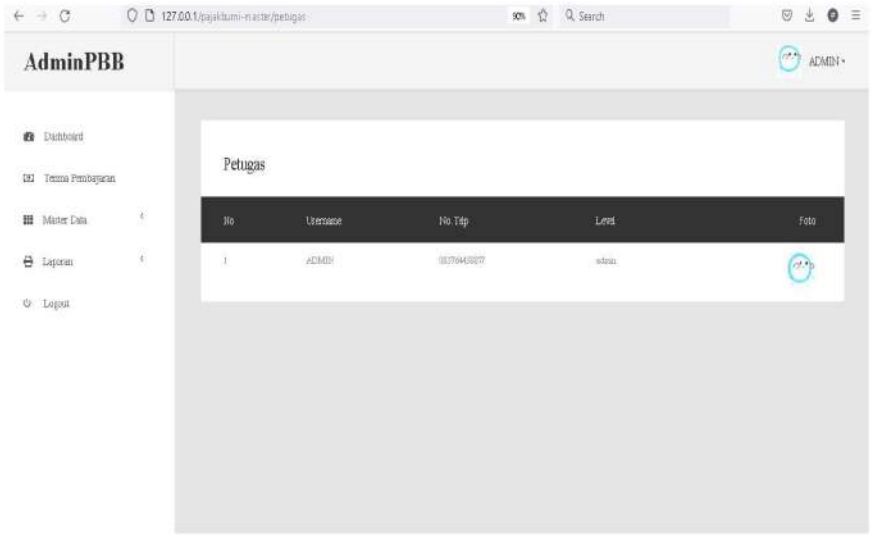


Figure 34. Officer Data Display

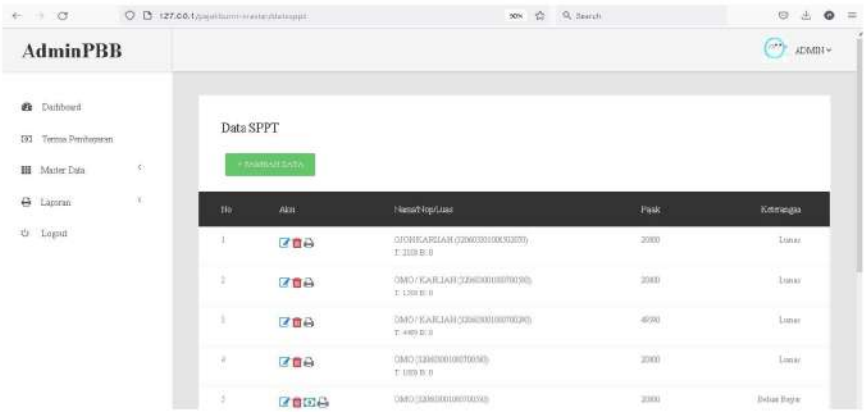


Figure 35. Display of SPPT Data List

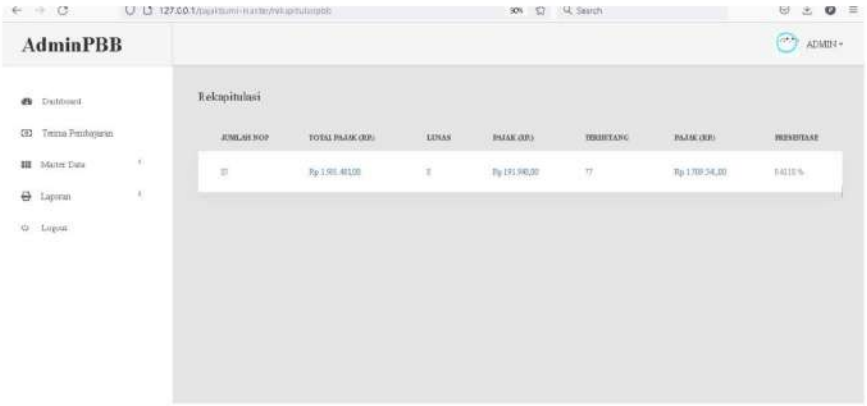


Figure 36. UN Recapitulation Display

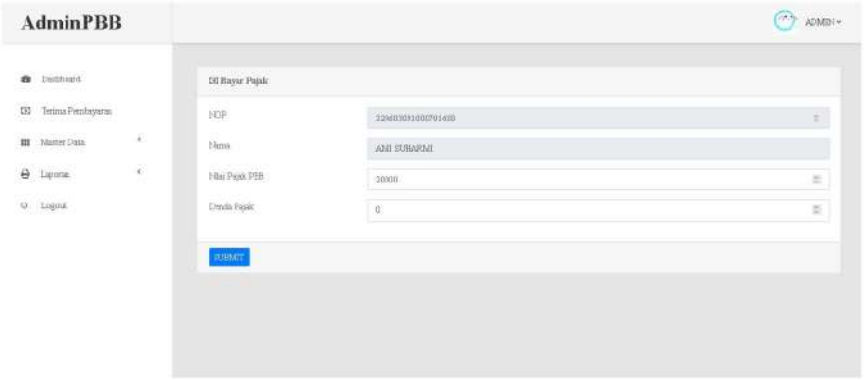


Figure 37. Tax Payment Input Display

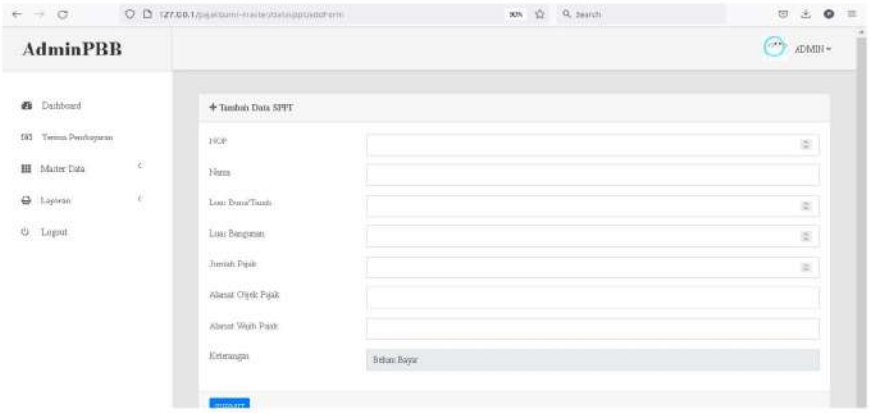


Figure 38. SPPT Data Input Display

**Output Page**

Output design is an output design in the form of a report as information resulting from data processing. Input/output design is also a reference for application makers in designing and building systems. Output design is the design of reports from the system to users taken from the database. The Report Page is the output of the system that we build and is used to analyze and serve as documents such as:

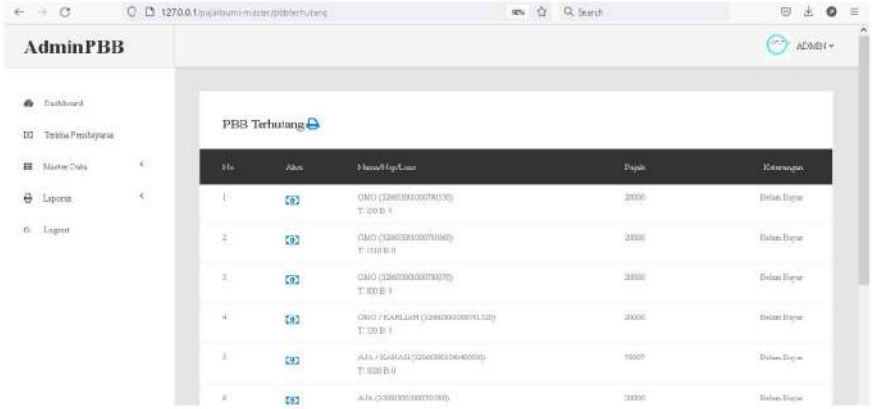


Figure 39. View of the PBB Outstanding Report

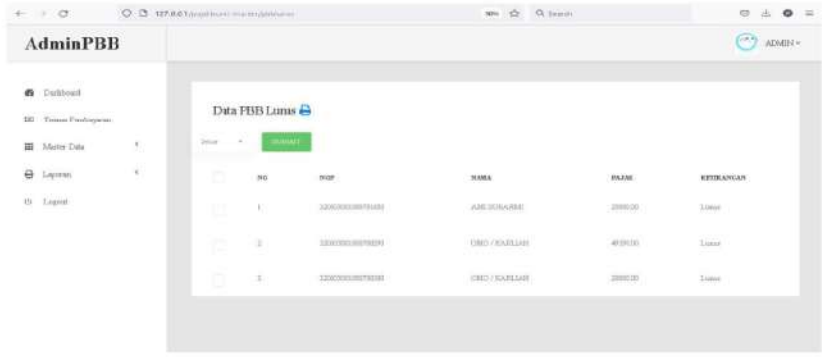


Figure 40. View of the UN Paid Off Report

PEMERINTAH KABUPATEN BANDUNG
KECAMATAN PANGALENGAN
DESA TRIBAKTIMULYA
Jl. Karyawinata No. 05 Kode Pos : 40378
Website: https://tribaktimulya.desa.id/

**Laporan PBB Terhutang**

| No | NOP | Nama | Luas | | Keterangan |
|---|---|---|---|---|---|
| | | | Tanah (T) | Bangunan (B) | |
| 1 | 32060300100070108 | Aja | 1.400 m2 | 0 | Belum bayar |
| 2 | 32060300100040122 | Amir | 1.680 m2 | 0 | Belum bayar |
| 3 | 32060300100040161 | Maman | 1.205 m2 | 0 | Belum bayar |
| 4 | 32060300100070227 | Aso S | 60 m2 | 60 m2 | Belum bayar |

Figure 41. Display of Payable PBB Print Out


PEMERINTAH KABUPATEN BANDUNG
KECAMATAN PANGALENGAN
DESA TRIBAKTIMULYA
Jl. Karyawinata No. 05 Kode Pos : 40378
Website: https://tribaktimulya.desa.id/

**Laporan PBB Lunas**

| No | NOP | Nama | Pajak | Keterangan |
|---|---|---|---|---|
| 1 | 32060300100070056 | Omo | 20000 | Lunas |
| 2 | 32060300100070246 | Idun | 20000 | Lunas |
| 3 | 32060300100070071 | Icin | 20000 | Lunas |
| 4 | 32060300100070244 | Mara | 20000 | Lunas |
| 5 | 32060300100040124 | Yayat | 20000 | Lunas |

Figure 42. View of the Lunas PBB Print Out

## Conclusion

Following an extensive examination of the prevailing system inside a village in West Java, coupled with interviews conducted with pertinent stakeholders, the present program was designed according to the specific requirements of the village. The e-PBB is a software application designed to calculate the Land and Building Tax. Implementing this system promises to enhance the organization and efficiency of land and building tax computation, management, and allocation.

The results derived from this Final Project are as follows:

The distribution of SPPT remains unchanged; however, it has now been reorganized in terms of its allocation. The systematic storage of data input and archiving of DHKP files and SPPT from every taxpayer is undertaken. Accessing information about the track record of both building and land area is a straightforward process that may be conducted during regular inspections.

Recommendations:

Regular maintenance of both software and hardware is essential to ensure seamless operation. In order to ensure accuracy and reliability, it is imperative to conduct surveys to validate data about the precise land or building area actively. This practice is essential as it prevents overreliance on preexisting information alone.

## References


Huda, B., & Priyatna, B. (2019). Penggunaan Aplikasi Content Management System (CMS) Untuk Pengembangan Bisnis Berbasis E-commerce. *Systematics*, *1*(2), 81. https://doi.org/10.35706/sys.v1i2.2076

Iqbal, M., Sutarman, S., & Irmansyah, D. (2019). Perancangan Sistem Informasi Project Management Berbasis Web Pada PT Visionet Data Internasional. *Academic Journal of Computer Science Research*, *1*(1). https://doi.org/10.38101/ajcsr.v1i1.235

Ramadhan, R. F., & Mukhaiyar, R. (2020). Penggunaan Database Mysql dengan Interface PhpMyAdmin sebagai Pengontrolan Smarthome Berbasis Raspberry Pi. *JTEIN: Jurnal Teknik Elektro Indonesia*, *1*(2), 129–134. https://doi.org/10.24036/jtein.v1i2.55

Rioni, Y. S. (2020). Analisis Peningkatan Kepatuhan Wajib Pajak Dalam Pembuatan NPWP UKM di Kebun Lada Kecamatan Hinai Kabupaten Langkat. *Jurnal Perpajakan*, *1*(2), 28–37. http://jurnal.pancabudi.ac.id/index.php/jurnalperpajakan/article/view/805

Siahaan, M. (2018). Penerapan Penghitungan Pajak Dan Pencatatan Serta Pelaporan Pajak Bumi Dan Bangunan Apartemen X. *Jurnal Riset Manajemen Dan Bisnis (JRMB) Fakultas Ekonomi UNIAT*, *3*(1), 121–134. https://doi.org/10.36226/jrmb.v3i1.94

Sudjiman, P. E. S. dan L. S. (2018). KOMPUTER DALAM PROSES PENGAMBILAN KEPUTUSAN Paul Eduard Sudjiman dan Lorina Siregar Sudjiman COMPUTER BASED MANAGEMENT INFORMATION SYSTEM. *Jurnal TeIKa*, *8*, 55–67. https://jurnal.unai.edu/index.php/teika/article/view/2327

Syarif, M., & Nugraha, W. (2020). Pemodelan Diagram UML Sistem Pembayaran Tunai Pada Transaksi E-Commerce. *Jurnal Teknik Informatika Kaputama (JTIK)*, *4*(1), 70 halaman. http://jurnal.kaputama.ac.id/index.php/JTIK/article/view/240

# Rifka-EBMTJ-TURNITIN

ORIGINALITY REPORT

| 2% | 1% | 1% | 2% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Submitted to Gujarat Technological University<br>Student Paper | 1% |
| 2 | Nela Dharmayanti. "The Effect Of Tax Office Service Quality And Taxpayer Income On Land And Building Tax Compliance", JAK (Jurnal Akuntansi) Kajian Ilmiah Akuntansi, 2023<br>Publication | 1% |
| 3 | garuda.kemdikbud.go.id<br>Internet Source | 1% |

| | | | |
|---|---|---|---|
| Exclude quotes | On | Exclude matches | < 1% |
| Exclude bibliography | On | | |