# Implementation of MVC (Model-View-Controller) Concept in Develop Management Information System for Transportation Fees (Uang Jalan Operasional) and Revenue in Trucking Companies

Rehulina Tarigan
Faculty of Computer Science
Universitas Banten Jaya
Jl Syech Nawawi Albantani Serang, Banten, Indonesia

Andi Usri
Faculty of Computer Science
Universitas Banten Jaya
Jl Syech Nawawi Albantani Serang, Banten, Indonesia

Ahmad Surahmat
Faculty of Computer Science
Universitas Banten Jaya
Jl Syech Nawawi Albantani Serang, Banten, Indonesia

## ABSTRACT
The trucking business in Cilegon and Anyer - Indonesia is growing quite well in line with the growth of new middle and upper scale industries. The research case study was conducted at the logistics company PT. Buana Centra Swakarsa, one of whose business units is trucking, where supervision and control of transportation fees (Uang Jalan Operasional - UJO) must be carried out properly so as not to interfere with the company's cash flow. Along with the development of the trucking business, the company's data processing must be done systematically, stored in a database with an integrated system so that data storage and data retrieval processes can be done more easily and accurately. In running the trucking business unit, PT. Buana Centra Swakarsa performs data processing of transportation fees (UJO) and revenue using spreadsheets Ms. Excel. Processing data like this, results in reports to manager or management that are not realtime, billing to customers is often late, recruitment of drivers and assigning jobs to drivers is not transparent. It is necessary to develop a computer-based information system to overcome these problems. Apart from the problems above, the process in this trucking system also often changes because it adapts to the contract or customer needs. In the future, changes in business processes or system development will inevitably occur. The development of this system requires a large amount of effort for the programmer to carry out program maintenance or repairs. An idea emerged to develop a system using the MVC (Model-View-Controller) concept so that program maintenance and adding modules would be easier and faster, especially if the programmer had to resign. In this research, the author will use the MVC concept with the Codeigniter framework. The methodology used to collect and analyze data is Research and Development (R & D) by collecting reports that are used by users, observation and interviews. The software development method used in a series of software development activities is the Waterfall Model and business process modeling using UML (Unified Modeling Language).

## Keywords
Trucking, Uang Jalan Operasional, Model View Controller, MVC Concept, MVC Architecture,

## 1. INTRODUCTION
Cilegon and Anyer are developing industrial areas that require trucking services to transport raw materials and industrial finished goods. As the trucking business grows, data processing must be carried out systematically, stored in a database with an integrated system so that data storage and data retrieval processes can be carried out more easily and accurately. In running the trucking business unit, PT. Buana Centra Swakarsa performs data processing of transportation fees (UJO) and revenue using spreadsheets. Broadly speaking, the business processes for managing cash out (UJO) and cash in (revenue) can be explained as follows:

a) The marketing party provides order data to the trucking administrator (supervisor) which contains information: the name of the customer, the schedule and the tonnage of goods shipped, the type of truck required, the location of loading and unloading.

b) Supervisor allocates the driver, schedule and license plate number of the vehicle that will carry out activities according to the order data from marketing and then to make a letter of assignment for the driver on duty.

c) The cashier will record the data in the assignment letter to spreadsheet, such as: vehicle number, route, and name of the customer as well as the amount of transportation fees (UJO) that will be submitted to the driver. UJO include: diesel money, food allowance, loading and unloading fees, tolls, user fees, commissions and ritase.

d) After receiving the UJO, the driver will go to the loading location and receive a Delivery Note (DN) to be taken to the unloading location.

e) After the driver returns, they will submit a travel document to the cashier as proof that the activity of transporting goods to the customer has been completed. Based on the travel document that has been signed and stamped by the customer at the unloading location, the cashier will record the amount of tonnage and revenue to be paid by the customer in the form of a commercial invoice which will be attached to the billing document.

f) On the cashier side, there are several processes they will carry out on spreadsheet media, namely: closing process per day, printing the Accountability Report (LPJ) which will be submitted to the operational manager, checking UJO documents or travel documents that the driver has not returned manually.

As explained above, that data processing of cash out (UJO) and cash in (revenue) is done on a spreadsheet, so some problems often occur as follows:

a) Controlling transportation fees (UJO) is difficult because managers take a long time to get reports. Usually reports are given by cashiers to manager in the form Acccountability Report (LPJ – Laporan Pertanggung Jawaban), while reports on vehicle utility and driver productivity are provided by the supervisor.

b) Many of the revenue or income was not recorded because the driver did not return the travel document. After the driver returns to the pool, they do not straight to the cashier to return the travel document but go home to rest. So that the UJO data and revenue were out of sync and billing to customers was late because the commercial invoice was incomplete, disrupting the company's cash flow

c) Data is not presented in real time so that management has difficulty analyzing problems and taking strategic steps to solve problems that arise

d) There is no standard report, so cashiers work hard and cause fatigue which results in invalid data recorded on spreadsheets

e) Cheating officers in the field are difficult to identify, such as: making additional UJO unnecessary, supervisor are unfair in giving jobs to drivers because they give more work to certain drivers (have kinship or friends), driver recruitment is carried out without information to Human Resource Development (HRD)

The cause and effect analysis of the above problems can be described in the fishbone analysis of cause and effect problems as shown in Fig 1. The process in this trucking system also often changes because it adapts to the contract or customer needs. In the future, changes in business processes or system development will inevitably occur. The development of this system requires a large amount of effort for the programmer to carry out program maintenance or repairs, especially if the system development is done using native PHP (without framework). It is well known that native PHP is not suitable for business processes with gradually expanding scope. Programming is complex for some programmers because they have a different logic and framework than programmers who develop systems from scratch. The authors [1] revealed that in web programming with native PHP there is no specific development pattern standard, so it is possible to build small-scale applications and do not experience frequent changes, so that one developer can develop a structure that is different from other developers. Based on this problem, an idea emerged to build or develop a system using the MVC (Model-View-Controller) concept so that programming and adding modules would be easier and faster, especially if the programmer had to resign.
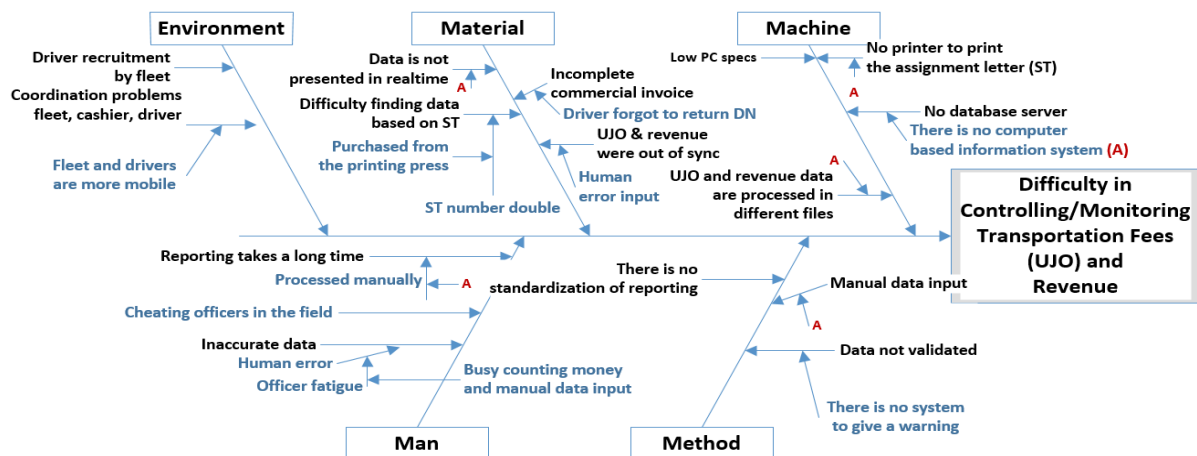


**Fig 1: Fishbone Analysis of Cause and Effect Problems**

There are several reasons why a developer would like to choose PHP with frameworks, especially because they offer developers the ability to create more complex, secure and complete web applications faster than ever before. Due to its convenient file structure, frameworks can speed up programmers coding and increase their productivity. PHP frameworks are available in all kinds of shapes and sizes and have targeted developers with different levels of experience, application needs, hosting capabilities and development time [2]. In this research, the author will use the MVC concept with the Codeigniter framework. A framework provides a basic structure for the application being developed so that programmers can immediately concentrate on creating the application features that are being developed without having to create an application from scratch [3]. Codeigniter has provided libraries that allow us to more easily and quickly develop database applications [4].

## 2. LITERATURE REVIEW
Designing good software is hard. It involves taking into consideration all aspects of the application you need to build Many experienced developers have had these problems and, over time, common patterns have emerged that assist in solving them [5]. MVC stands for Model, View, Controller, which is an architecture for creating a program. This architecture emphasizes the division of program components into three main parts, namely Model, View, and Controller [6]. The diagram in the Fig 2 provides a conceptual overview of the MVC architecture.
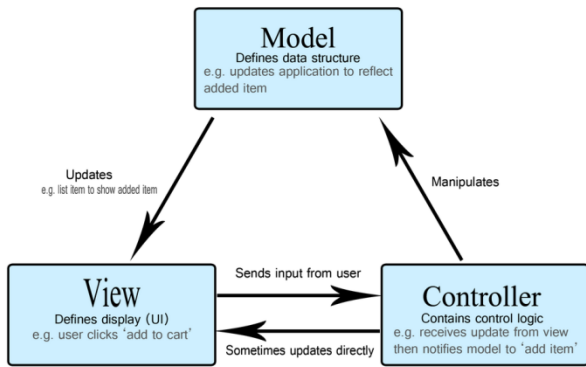
**Fig 2: Model View Controller Example**
**(Source: MVC - MDN Web Docs Glossary: Definitions of Web-related terms | MDN developer.mozilla.org)**

To clarify the concept of MVC with the three components, the author [5] explains as follows:

- The model is where all of the application's business logic is stored. If the application must access information to the database, the code to do so is stored in the model

- The view is where all the user interface elements are stored. This can include HTML, CSS, and JavaScript files. Whatever the user sees can be stored in the view.

- The controller is the component that connects the model and the view. The controller isolates the business logic present in the model from the user interface elements of a view. Controllers are the first entry point into this trio of components.

In his journal entitled Application of the Model View Controller Concept in Web-Based Software Management System Design, Suendri (2018) wrote that the MVC concept makes it easier for programmers to manage database queries, code management, data validation and security. Likewise, an international journal entitled Implementation of Model-View-Controller Architecture Pattern for Business Intelligence Architecture, discusses the application of the MVC concept in building Business Intelligent Architecture applications. This journal was written by Medha Kalelkar [9]. In this journal, the author explains the benefits of using the MVC concept in developing Business Intelligent Architecture applications, including: efficiency, complexity, scalability, heterogenity and maintainability.

In general, the framework for the PHP (Hypertext Prepocessor) programming language uses the MVC pattern concept. One of the frameworks is the CodeIgniter framework used in this research. A framework can be simply defined as a collection of functions or procedures and classes for specific purposes that are ready to be used so that it makes it easier and faster for programmers to create programs without having to create functions or classes from scratch. The framework has a good application structure such as standard coding, best practices, design patterns and common functions [10]. Another advantage of frameworks is that the programming styles can be uniform and standard. Multiple programming styles can confuse other team members, thus hindering the programming process. Programs created using a framework "force" programmers to follow the rules of their use. Thus, whoever the programmer is, as long as they master the framework, they can certainly work on programs that are made because of the uniform programming style.
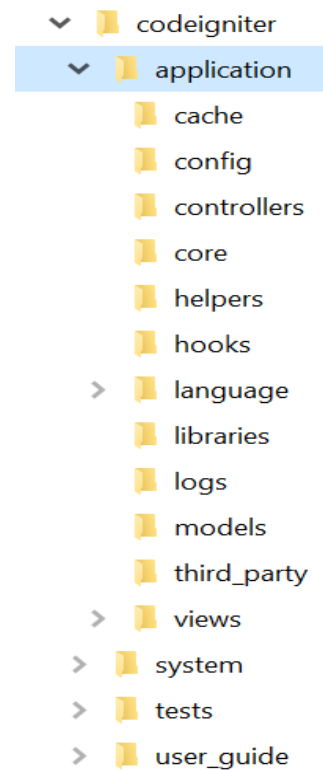
In this study, the author uses the CodeIgniter framework for certain reasons. CodeIgniter is easy to learn, simple structure, complete and clear documentation. The Fig 3 shows the directory structure of the Codeigniter framework.



**Fig 3: Codeigniter Directory Structure**

The MVC pattern concept resides in the **application folder** which consists of many folders as well. But those that are directly related to the MVC concept are the controllers, models and views folders. The author uses the HMVC (Hierarchical Model View Controller) structure in the implementation of the UJO and revenue management information system. HMVC is a hierarchy or arrangement of the MVC pattern that continues into a single application. Each MVC stack is completely independent (stand alone) and can execute in the absence of the other [11]. Fig 4 illustrates how this works. Each triad functions independently from one another. A triad can request access to another triad via their controllers. Both of these points allow the application to be distributed over multiple locations, if needed.
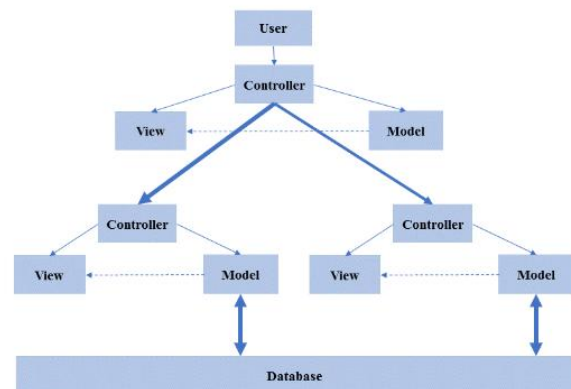


**Fig 4: Hierarchical Model View Controller Pattern**

# 3. RESEARCH METHOD

In this study, the authors and the team will carry out several activities to be able to understand the current information system and collect the necessary data and information so that they can design the database and user interface according to user needs (user requirements). The method used by the author in collecting data and information is the Research and Development (R&D) method. The activities carried out for research are as follows:

a) Interview, the author held a meeting with the user of the trucking operations section of BCS-Logistics to interview so that the ongoing business processes can be understood. The author also asks the user for some data which is usually in the form of a report. Based on reports obtained from the field, the authors and the team analyzed the data and translated it into database and user interface design

b) Observation, is carried out directly into the field to see clearly what the user is doing in carrying out the process. The activity starts with the driver receiving an assignment letter, the cashier submitting the UJO, the driver leaving for the loading location, delivering the goods to the unloading location and returning to the pool and submitting the Delivery Note (DN) to the cashier. Based on the DN, the cashier will record data such as the weight of goods shipped along with the rates to generate revenue or cash in.

c) Literature study, is carried out to obtain all the theories that support the MVC pattern concept used to develop information systems or software. The author is also looking for and studying theories related to the Codeigniter framework as one of the frameworks used in the PHP programming language to build the intended application.

The software development methodology used is the Waterfall Model which performs several steps of the activity. The Waterfall SDLC (Software Development Life Cycle) model is a sequential software development process in which progress is regarded as flowing increasingly downwards through a list of phases that must be executed in order to successfully build a computer software. The Waterfall model defines several consecutive phases that must be completed one after the other and moving to the next phase only when its preceding phase is completely done. For this reason, the Waterfall model is recursive in that each phase can be endlessly repeated until it is perfected [12]. Business process modeling uses the Unified Modeling Language (UML). The UML is the standard way to model systems, particularly software systems [13]. UML can be used to analyze problems, make business process modeling and design software to be developed.

# 4. RESULTS AND DISCUSSION

## 4.1 Comparison of Native PHP with MVC Concept

Actually, there are many things that we can discuss about the comparison between native PHP (without MVC concept) and PHP using the MVC concept, for example in terms of writing programs and the structure of each php file which is clearly very different. In this article, the author will only briefly discuss the differences between native PHP and the MVC concept in terms of modularity. In native PHP programming, all php modules or files are placed in the same folder, even though the function of each file is different (Fig 5). Examples is taken from Inventory Information Systems. The files group

(barang.php, barang_edit.php, barang_hapus.php and barang_tambah.php) has one same function, namely to input goods master data into the system, but because there is a function (CRUD - Creat, Read, Update, Delete) then we must create four different php files and mix them with other files with different functions. This makes it difficult for programmers to program maintenance because they have to find certain file among many files in one folder.

If we use the MVC concept, group file of **1a** and **1b** (Fig. 5) can be created in one folder, for example the name of the folder is barang. In that folder, there is a model, view, controller folder. Likewise, group files of **2** can be created in one folder, for example barang_masuk. In barang_masuk folder, there are models, views, controller folder. Thus easy to perform program maintenance because it has a clear file structure in the MVC concept.
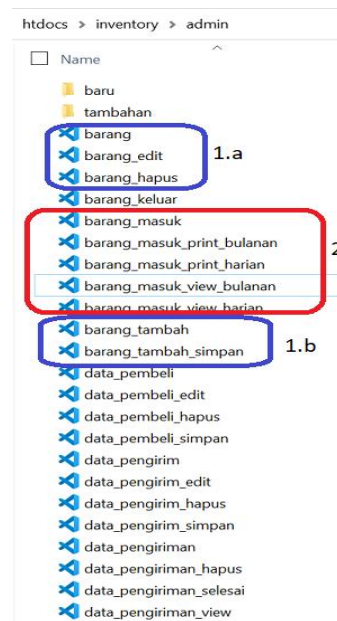


**Fig 5: The File Structure in Native PHP**

In addition to program maintenance difficulties, we will also encounter difficulties in terms of file names that will appear in the url when the file is called. The complication is, there are so many filenames that we have to call just to perform simple functions like CRUD (Cread, Read, Update, Delete) for the master data. One url represents one php file. In Fig 6 we can see that url will display a list of items: http://localhost/inventory/admin/barang.php. If we want to add item data, we can click button "TAMBAH PRODUK BARU", then http://localhost/inventory/admin/barang_tambah.php will appear in the toolbar.
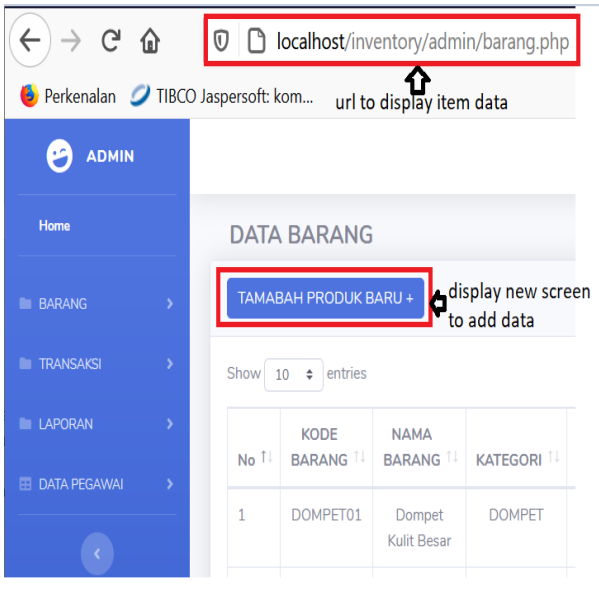
**Fig 6: One Url Represents One Filename.php (in Native PHP)**

This is different from PHP which implements the MVC concept. In the MVC or HMVC concept, each module is located in one folder consisting of three folders, namely the models, views and controller folder. It can be said that a folder represents a module to carry out a task or a function. To explain the MVC concept, the author takes the example of the UJO and Revenue management information system that has been built by the author and the team.
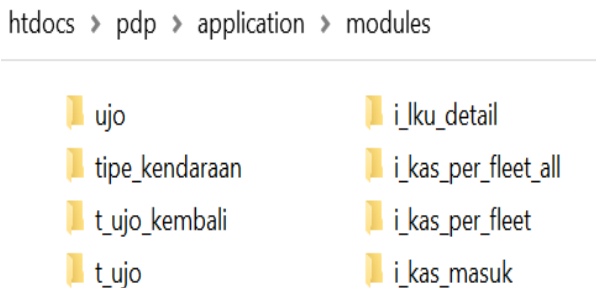


**Fig 7.a: The Modules in MVC**

Fig 7.a shows a folder representing a module that represents a task or a function. For example, the ujo folder is a module that functions to display transportation fees data. If we want to display that function, just type the folder name in the toolbar (url) like this http://localhost/pdp/ujo .The url will not change if we want to add, change or delete (CRUD) UJO data. This is because the system applies the MVC concept, in the ujo folder there are folders, models, views and controllers (Fig 7.b). If we want to do program maintenance for the ujo module, simply open the ujo folder and maintenance can be done easily because it is easy to find files and programs that are formed in a framework that has standard functions or procedures.
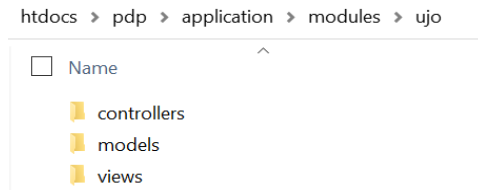


**Fig 7.b: Ujo Folder**

So that in terms of the folder and file structure, it can be said that the MVC concept creates a structure that is quite good and easy to understand. In other words, the concept of modularity is clearly seen in application programs that apply the MVC concept.

## 4.2 System Development

The system modeling developed can be described by use case diagram. The Use case diagram of the information system for the UJO and Revenue that will be developed can be described as general in Fig 8.
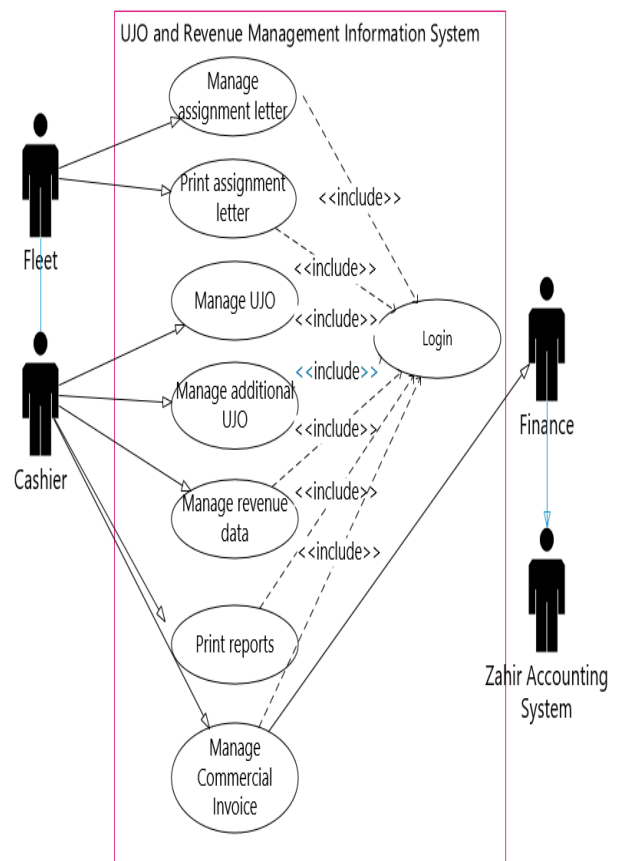


**Fig 8: Use Case UML Diagram**

### 4.2.1 Database Implementation

The following is a database structure image (Fig. 9) that has been designed and implemented to store all data that will be inputted by the user through the screen in the developed UJO and revenue management application system.
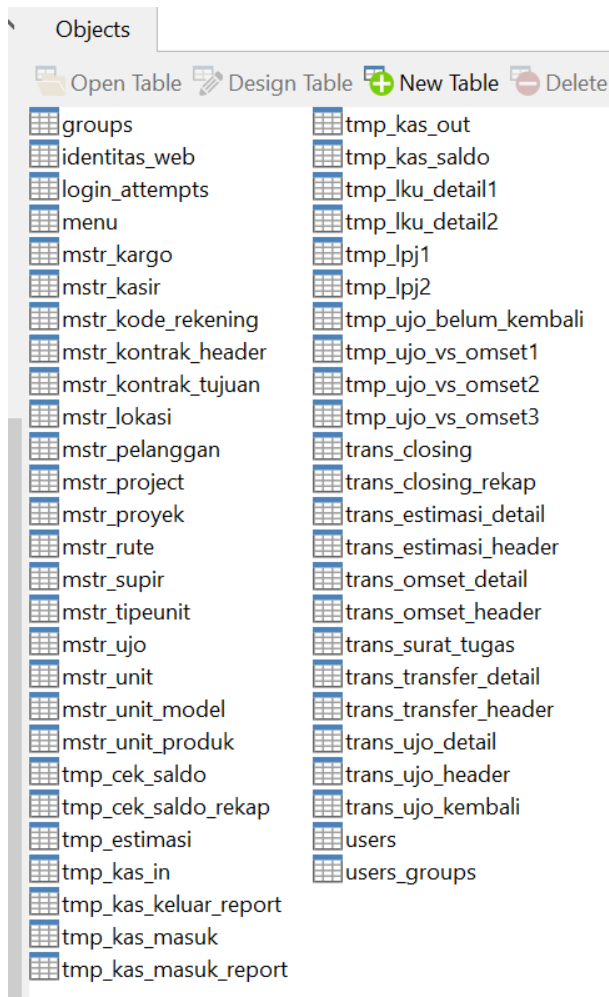
**Fig 9: Database**

### 4.2.2 MVC Concept Implementation

In this sub-chapter, the author will discuss how to implement the MVC concept in building a UJO (cash out) and revenue (cash in) management information system. The application folder will often be accessed because it is on the folder contains a number of folders with the MVC concept and a set of code written in the application folder. The contents of the application folder which consists of the model, view and controller folders.

Fig 10 is the modules folder structure which consists of 33 folder (modules) that have been created. Every folder module consists of folder model, view, controller.
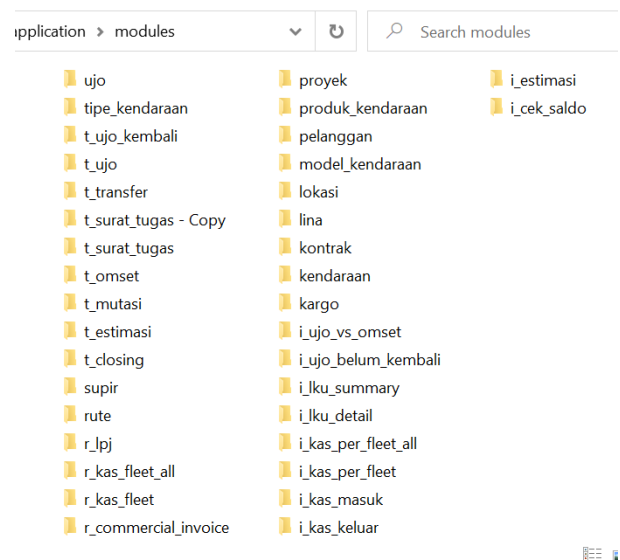


**Fig 10: Modules Folder Structure**

The kendaraan folder consists of the models, views, controller folder. Each module represents a screen that appears on the url when a menu is clicked. For example for module kendaraan, we just type kendaraan in the url toolbar, with url http://localhost/pdp/kendaraan (Fig.11)
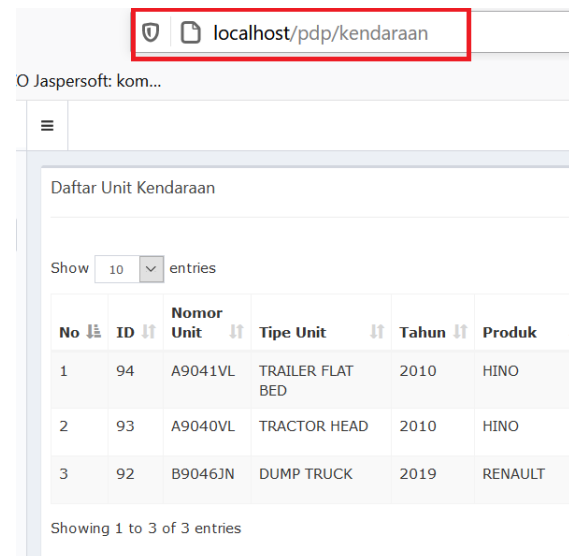


**Fig. 11: Menu Kendaraan**

### 4.2.3 Result of System Development

As a result of implementing the system, the author will display a table (Table 1) that describes how the system can be used as a tool to solve problems that have been described in the introduction.

**Table 1: Analysis Result**

| NO | PROBLEM (Before the system was developed) | RESULT (After the system was developed) | DESCRIPTION | SCREENSHOOT |
|---|---|---|---|---|
| | **ANALYSIS RESULTS** | | | |
| 1 | Cheating officers in the field are difficult to identify, such as: supervisor are unfair in giving jobs to drivers because they give more work to certain drivers (have kinship or friends) | The application system provides a report menu to display the utility of each driver | The leader (manager) can see the utility of the driver so as to prevent the unfairness of the supervisor in assigning jobs to each driver | Fig. 12: Driver Utility |
| 2 | Many of the revenue was not recorded because the driver did not return the travel document (Delivery Note) | If the driver has not returned the travel document to the cashier, then the driver may not be given an assignment letter because the application system will block it | The driver must return the previous travel document so that he can return to activities | Fig. 13: The System Blocks to Create Assignment Letters |
| 3 | It is difficult to control transportation costs (UJO) because the leader (manager) takes a long time to get reports, for example the Accountability Report (LPJ) | The application system provides a menu for managers to get realtime reports | Reports can be accessed with the path REPORT → LPJ | Fig. 14 : LPJ Report |
| 4 | Data is not presented in real time so that management has difficulty analyzing problems and taking strategic steps to solve problems that arise | In the application system that has been built, a menu is available for inquiries about the required data, for example transportation fees versus income | Leaders can easily access all the data they want. Data can be accessed via the INQUERY and REPORT menus | |
| 5 | There is no standard report, so cashiers work hard and cause fatigue which results in invalid data recorded on spreadsheets | The application system provides a menu to generate reports or perform data inquery | The types of reports are designed according to user requirements based on certain | |
| 6 | Cheating officers in the field are difficult to identify, such as: making unnecessary additional costs for transportation (UJO) | The application system will automatically display transportation costs that cannot be edited by the cashier | The leader can see the total transportation costs of each driver in real time | |

**DRIVER UTILITY**

PERIODE 2020-08-01 - 2020-08-31

| | Tanggal | No Kendaraan | No Surat Tugas | Tujuan | Driver | Jml Ritase | Pelanggan |
|---|---|---|---|---|---|---|---|
| 1 | 13/08/2020 | B9046JN | ST-20200813-00008 | Dari Cilegon ke CIKUPA | ABDUL | 2 | Indocement |
| 2 | 13/08/2020 | A9041VL | ST-20200818-00009 | Dari Cilegon ke Bekasi | Abi Madian | 1 | Cabot Indonesia, |
| 3 | 13/08/2020 | A9041VL | ST-20200813-00007 | Dari CILEGON ke Dadap | ADE EKA | 2 | Bahari Caraka |
| 4 | 05/08/2020 | A9040VL | ST-20200804-00004 | Dari CILEGON ke Bekasi | ADI | 1 | Indocement |
| 5 | 05/08/2020 | A9040VL | ST-20200805-00005 | Dari CILEGON ke Bekasi | BUDIONO | 1 | Indocement |
| 6 | 10/08/2020 | A9040VL | ST-20200805-00005 | Dari CILEGON ke Bekasi | BUDIONO | 1 | NS Bluescope |

**Fig. 12: Driver Utility**
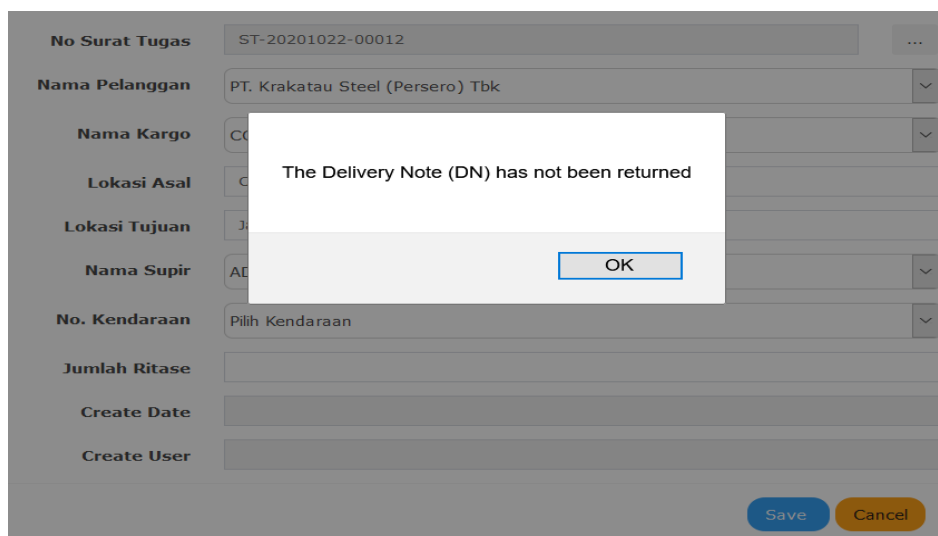


**Fig. 13: The System Block to Create New Assignment Letter**

**FORM PERTANGGUNGJAWABAN DAN PENGAJUAN UANG MUKA PROYEK**  Tanggal   Thursday 13 August 2020

**DAFTAR PROYEK**

| | | |
|---|---|---|
| Kode Proyek | : | 2-01-002 |
| Nama Proyek | : | TRANS. TRUCKING |
| Tanggal Pengajuan | : | 14/08/2020 |

**PERTANGGUNGJAWABAN UANG MUKA**

| | | |
|---|---|---|
| Tanggal | : | 13/08/2020 |
| Saldo Awal | : | 0 |
| Uang Muka Diterima | : | 20,003,500 |
| Pengembalian | : | 4,000,000 |

**Rincian Pengeluaran**

| Kode Akun | Nama Akun / Keterangan | Cost Centre Code | Project Code | Fund Code | Customer Code | Budget Code | Jumlah |
|---|---|---|---|---|---|---|---|
| CC107 | Solar | | | | | | 6,625,000 |
| CC501-2 | Tol | | | | | | 2,900,000 |
| CC502 | B/M | | | | | | 200,000 |
| CC503 | Uang Makan | | | | | | 200,000 |
| CC501-1 | Retribusi | | | | | | 200,000 |
| CC504 | Ritase | | | | | | 0 |
| CC510 | Komisi | | | | | | 0 |
| | Non UJO | | | | | | 350,000 |
| | Kas Kecil Project (KKP) | | | | | | 0 |
| | Kas Kecil Maintenance (KKM) | | | | | | 0 |
| | | | | | Total Pengeluaran | Rp. | 10,475,000 |
| | | | | | Saldo Akhir | Rp. | 2,653,500 |

**PENGAJUAN ESTIMASI UANG MUKA OPERASIONA**

| No | Rencana Kerja / Kebutuhan | Quantity | Harga Satuan | Kebutuhan Uang Muka |
|---|---|---|---|---|
| 1 | Dari BCS CIlegon ke BALAI ENDAH Bandung (FL 10) | 1 | Rp 2,750,000 | Rp 2,750,000 |
| 2 | Dari Anyer ke POSCO (CPU) | 5 | Rp 2,000,000 | Rp 10,000,000 |
| | | | | 12,750,000 |

**Fig. 14: LPJ Report**

## 5. CONCLUSION

a) Development of information systems or application systems will be easier and faster by using the MVC concept. Program maintenance will be easier by using the MVC concept.

b) UJO and Revenue management information systems can speed up the billing process to customers by displaying accurate data. Information systems can display data in real time to all users who need it, so it's easier to control and monitor UJO and revenue.

## 6. ACKNOWLEDGMENTS

## 7. REFERENCES

[1] Wildan Mualim, Gema Ulama Putra. 2017. "Implementasi Framework MVC Pada Sistem Informasi Akademik di STMIK Yadika Bangil", Jurnal SPIRIT, vol. 9, no. 2, hal. 35-39

[2] Khaoula Benmoussa, Majida Laaziri, Samira Khoulji, Mohamed Larbi Kerkeb, Abir El Yamami. 2019. "A new model for the selection of web development frameworks: application to PHP frameworks", International Journal of Electrical and Computer Engineering (IJECE), vol.9, no.1, pp. 695-703

[3] Awan Pribadi Basuki. 2014. Proyek Membangun Website Berbasis PHP dengan Codeigniter. 1st ed. Yogyakarta: Lokomedia.

[4] Betha Sidik. 2019. Mengembangkan Framework Aplikasi Database dengan Codeigniter 3. 1st ed. Bandung: Informatika.

[5] Pitt, Chris. 2012. "Pro PHP MVC". United States of America: Paul Manning.

[6] Suhanto, Agus. 2008. "Mengenal ASP.Net MVC". Indonesia. Net Developer Community.

[7] MDN Web Docs. 2019. MVC. Viewed 17 September 2020. https://developer.mozilla.org/en-US/docs/Glossary/MVC

[8] Suendri.2018."Penerapan Konsep Model View Controller Pada Perancangan Sistem Manajemen Software Berbasis Web". Jurnal JISTech, vol. 3, no.2

[9] Medha Kalelkar, Prathamesh Churi, Deepa Kalelkar. 2014. "Implementation of Model-View-Controller Architecture Pattern for Business Intelligence Architecture". International Journal of Computer Applications (0975 – 8887) Vol. 102 No.12 https://research.ijcaonline.org/volume102/number12/pxc 3898786.pdf, Viewed 17 September 2020

[10] Badiyanto, Yosef Murya. 2018. Project PHP: Membangun Sistem Informasi Akademik Dengan Framework Codeigniter". 1st ed. Yogyakarta: CV Langit Inspirasi.

[11] Randi Proska Sandra, Ahmaddul Hadi, Yasdinul Huda. 2017. "Perancangan Aplikasi Clickost Berbasis Web Menggunakan Metode HMVC (Hierarchical Model View Controller Sebagai Sistem Pencarian dan Promosi Indekost di Kota Padang". Jurnal Vokasional Teknik Elektronika dan Informatika, vol. 5, no. 2.

[12] Youssef Bassil. 2012. "A Simulation Model for the Waterfall Software Development Life Cycle". International Journal of Engineering & Technology (IJET), vol. 2, no. 5

[13] Russ Miles, Kim Hamilton. 2006. Learning UML 2.0. 1st ed, O'Reilly Media, Inc, United State of America