

System Development of Vehicle Registration and Payment of Membership Indonesian Truck Entrepreneurs Association using Hierarchical Model-View-Controller Concept

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

Dedi Juniansha
Faculty of Computer Science
University of Banten Jaya
Jl Syech Nawawi Albantani
Serang, Banten, Indonesia

Raden Kania
Faculty of Computer Science
University of Banten Jaya
Jl Syech Nawawi Albantani
Serang, Banten, Indonesia

ABSTRACT

The problems that have existed in Aprindo (Asosiasi Pengusaha Truk Indonesia) Cilegon - Indonesia are regarding the registration of trucking companies and the inaccurate payment system for each vehicle unit. Aprindo as an association of trucking companies in Indonesia, strives to obtain as many members as possible and register all vehicles owned by each trucking company. At this time, all member data in the form of trucking company data and their vehicle units are recorded manually on a spreadsheet. Aprindo also has the authority to block vehicles that have not been registered but still carry out activities at certain locations such as Pelabuhan Indonesia II (Pelindo 2), Merak Mas IKPP (Indah Kiat Pulp and Paper) and KBS (Krakatau Bandar Samudera. Inspection of vehicle registration status at the location is carried out manually by officers in the field based on a sheet of paper that is very prone to damage, torn or missing. The data printed on the paper is also not necessarily accurate because it is not printed every day or in other terms, the data on the paper is not necessarily in sync with the data in the Aprindo admin spreadsheet. This out of sync results in vehicle units that can still operate when in fact the company that owns the vehicle has not paid dues in the current month. Field officers and admins also find it difficult to check the payment of membership dues. Apart from the problems that arise above, the admin also has difficulty in providing reports in a fast time. The purpose of this research is to automate and integrate all existing business processes within the scope of Aprindo Cilegon. The development of an information system for registration and payment of vehicle unit fees will have a positive impact in providing accurate information on the payment status of each registered vehicle unit, transfer of contributions for each vehicle unit and data that can be accessed by Aprindo officers or representatives at locations where vehicle units operate. With this information system for registration and payment of vehicle units, it is hoped that all interested parties can quickly find out the data and information they need. For example, Aprindo officers can block operations if it is found that a vehicle unit has not paid dues, namely by checking the status of the vehicle through their mobile phone. The methodology used to collect and analyze data is observation, interviews with users and literature study. The System Development Life Cycle (SDLC) method will be used in developing the application system. Software development methodology for business modeling using UML (Unified Modeling Language). The software to be

built uses the concept of HMVC (Hierarchical Model-View-Controller) which can speed up application development and is easy to maintain. With this software, access to data can be done on a real-time basis and the recording of all data is integrated in a database.

Keywords

Aprindo, HMVC Concept, Modularization, Reusability, Trucking Registration.

1. INTRODUCTION

Aprindo (Asosiasi Pengusaha Truk Indonesia) has members of truck entrepreneurs throughout Indonesia. The goal is to foster and develop the ability of members to be able to compete globally. Aprindo also functions as a forum for unity and association of trucking entrepreneurs, in order to obtain information, consultation and advocacy on matters relating to economic activities nationally and globally in order to create a better business climate. The problem at Aprindo Cilegon is regarding the registration of trucking companies and the inaccurate payment system for each vehicle unit of the trucking company. Aprindo is trying to get as many members as possible and register all vehicles owned by each trucking company. At this time, all member data and their vehicle units are recorded on a spreadsheet. Aprindo has the authority to stop all activities on the vehicle unit if it is proven that it has not fulfilled its obligations. So far, at certain locations, there are Aprindo representatives who are tasked with checking the payment status of each vehicle unit. The representative will find it difficult because the checking is done manually, based on the data he obtained from the printed Aprindo admin (staff). Checking based on a piece of paper will be very vulnerable, because the paper is brought to the field which can be torn or wet. The data printed on the paper is also not necessarily accurate because it is not every day the data is printed or in other terms the data on the paper is not necessarily in sync with the data in the spreadsheet of Aprindo admin. This out of sync results in vehicle units that can still operate when in fact the company that owns the vehicle has not paid dues in the current month. In this case, Aprindo hopes that there are tools that can quickly check the registration and payment status of the vehicle.

Trucking company members are required to make payments every six months so that in a year members make payments twice. In fact, this agreement is often violated by negotiations

between Aprindo and its members so that the payment of contributions is not made every 6 months. For example, the first payment is made at once for 3 months, the second payment is for the next 4 months and the third payment is made for the next 5 months. This is of course very difficult for the Aprindo admin to check whether a certain vehicle unit has paid dues in a certain month or not. Apart from the problems that arise above, the admin also has difficulty in providing reports in a fast time. For example, presenting a report on a list of all member companies, a summary of the number of vehicle units, a report on a list of vehicle units that have not made payments, total bills to members. The spreadsheet also does not allow an image or document to be uploaded, for example proof of transfer, KIR document, STNK (vehicle registration certificate), NPWP (taxpayer registration number), company SIUP (business license) and so on. Based on the description of the phenomena that have been described previously, it can be identified several problems faced by Aprindo Cilegon with a system that is run manually, including:

- a) Trucking company registration payments cannot be controlled properly because data is entered manually on a spreadsheet. Data cannot be accessed in real time.
- b) Difficulties in monitoring and controlling the payment of dues for each vehicle owned by the trucking company.
- c) Aprindo officers in the field cannot properly control whether a vehicle is still allowed to operate because it has paid dues or cannot operate because it has not paid dues. This is due to checking based on paper that is easily damaged or wet so that the writing on the paper can be lost or unreadable.
- d) From the administrative side, reporting cannot be done quickly and is prone to data errors or data inconsistencies.

The point is, because the system is done manually, cash inflows from payments made by trucking companies cannot be monitored and controlled properly so that the income for Aprindo Cilegon is small, not as expected. Based on this background, the author intends to build an information system for registration and payment of vehicle unit fees that is able to provide accurate information about the status of member company registration payments, payment or transfer status of each vehicle unit and data that can be accessed by Aprindo officers or representatives at locations where the vehicle unit operates. In more detail, the purpose of developing this application system can be explained as follows:

- a) Create a database of company business units registered with Aprindo or others become a member of Aprindo
- b) Recording registered vehicle units in Aprindo
- c) Recording the payment of fees per vehicle unit by the trucking company every month or per certain period to Aprindo
- d) Validate where the system insists that there are no fees per vehicle which is not transferred per month. For example, if the last transfer was January 2021, then the system will fail if the next user input transfer is in the March 2021 period. That means February 2021 dues data must be inputted first.

2. LITERATURE REVIEW

The development of this system requires a large amount of effort for the programmer to carry out program maintenance.

Based on this problem, an idea emerged to build or develop a system using the HMVC (Hierarchical Model-View-Controller) concept so that programming and adding modules would be easier and faster, especially if the programmer had to resign. 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 [1]. In this research, the author will use the HMVC 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 [2]. Codeigniter has provided libraries that allow us to more easily and quickly develop database applications [3].

Hierarchical Model View Controller (HMVC) is a development version of the MVC (Model View Controller) design pattern or can be said to be a hierarchically implemented version of MVC [4]. MVC architecture emphasizes the division or separation of program components into three main parts, namely Model, View, and Controller [5]. Figure 1 provides a conceptual overview of the MVC architecture.

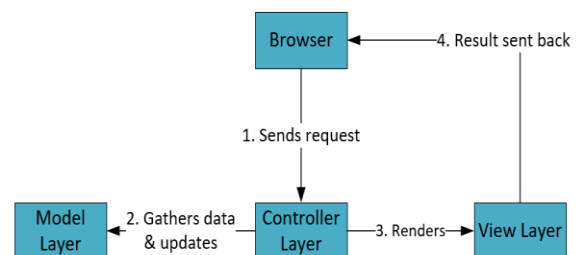


Fig 1: MVC Concept
(Source: <https://www.edureka.co/blog/mvc-architectureinjava/>)

In HMVC (Hierarchical Model View Controller), the programmer will be facilitated in managing the data section. In MVC, when the admin and team have a project, of course it will create many views, many controllers, and many models. Because it is done by many people, of course you will be confused when you put the program together. From these problems, a solution can be taken, so that one function or one module is made into one folder containing the model view controller [6]. HMVC consists of many independent MVC. All requests from each MVC must use its own view and controller and never load models outside of the MVC domain itself.

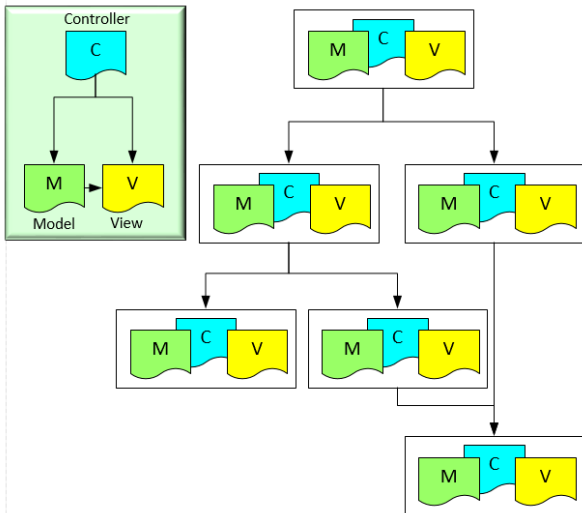


Fig 2: HMVC Concept

(Source: <https://code.tutsplus.com/id/tutorials/hmvc-an-introduction-and-application--net-11850>)

The HMVC architecture consists of several triads. Each triad functions independently of each other. A triad can request access to other triads through its own controller [7]. The author [8] often also mentions that each MVC (triad) is contained in a module or wrapped in a folder. In each of those folders there is a model-view-controller folder. The folder or module structure can be explained in the image below using the codeigniter framework. According to the authors [8], the HMVC concept has several advantages over MVC, including:

- **Modularization:** reduce dependencies between separate parts of the application.
- **Organization:** has folders (modules) of related MVC so as to make the work load or system workload lighter.
- **Reusability:** by nature by design, it is very easy to reuse any part of the code.
- **Extendibility:** making applications easier to develop without having to sacrifice maintenance or program maintenance.

In this study, the author uses the Codeigniter framework to build software applications. Someone who has mastered the MVC concept, can be said to have mastered the outline of any framework [2]. In the framework, a good application structure is available such as coding standards, best practices, design patterns and common functions so that programmers can directly focus on the system development process without having to think about application structures, coding standards and so on [9]. Codeigniter is a PHP framework that is popular and widely used around the world. Codeigniter has provided libraries that allow us to more easily and quickly develop computer-based applications (database applications) [3]. According to the results of the author's review [10] of the performance of codeigniter that the response time is quite fast. Besides that, CodeIgniter is also flexible, secure & user friendly, light weight and open source.

3. RESEARCH METHOD

3.1 Research Design

Research design is one of the stages that must be passed or made in order that research achieved its objectives. Research design is a work plan to make a construction that every

question can be answered. The stages of the process to be carried out in this research are depicted in Figures 3.a, 3.b and 3.c below.

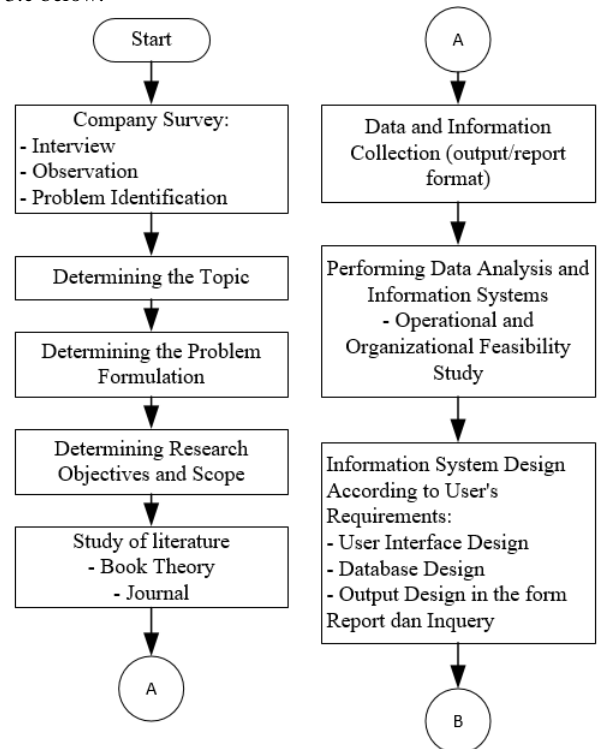


Fig 3.a: Model Research Flowchart

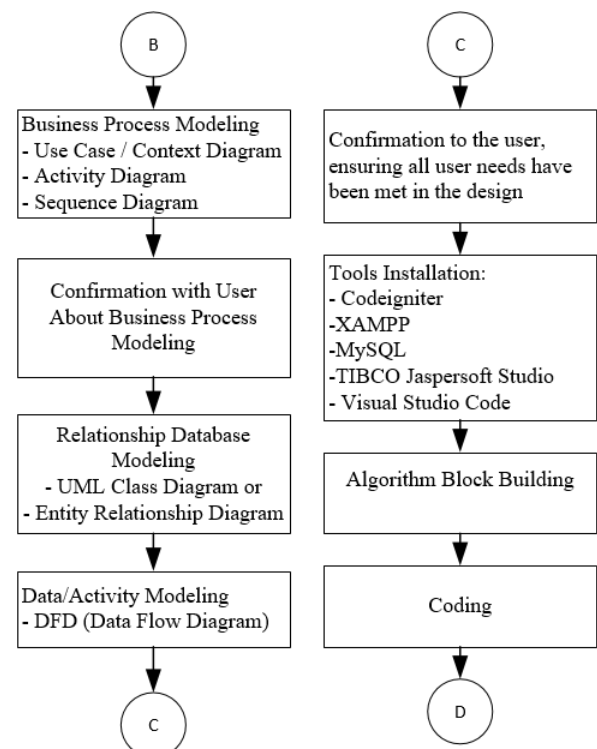


Fig 3.b: Model Research Flowchart (Continue)

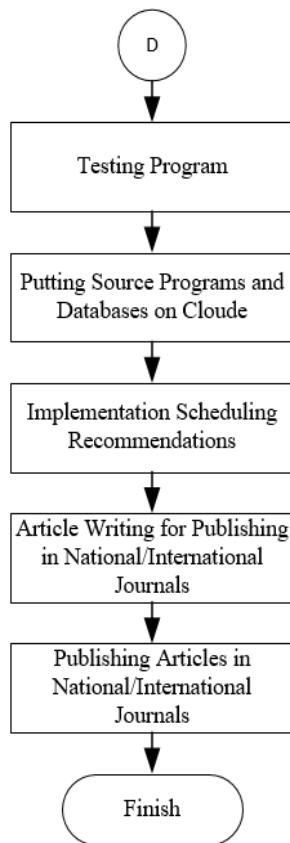


Fig 3.c: Model Research Flowchart (Continue)

3.2 Method of Collecting Data

The research methodology carried out in the early stages is as follows:

a) **Field Study (Company Survey)**

The field study is the initial research activity carried out to find out and understand the conditions in the company, the description of the system, as well as some of the problems that occur in the company. The methods used in carrying out this field study include:

- i) **Observation**, namely direct observation of the company to find out the vehicle registration process and payment of Aprindo membership dues, finding symptoms of problems identified in the company, which then these symptoms will be used as research object
- ii) **Interview**, which is a technique of collecting data and information by asking questions directly with the user or admin of Aprindo. This activity is carried out to obtain additional information that will support the identification of symptoms of problems in the company.

b) **Literature Study**

Literature study is used as a theoretical basis in solving scientific problems. After the topic is determined at this stage, a literature study is carried out that can support research work. In this stage, books that support research materials and journals and theses from previous research are used. For system developers, literature study also includes searching for references from the internet which can

only be in the form of articles on certain community websites. Usually looking for references when installing supporting software, looking for certain syntax when coding and so on. In this literature study, the author learned a lot from previous journals about the MVC concept and the HMVC concept and their differences. The author also learns how to configure to implement the HMVC concept.

Before conducting further discussion, based on the results of interviews with users, the authors can identify several problems in general, namely:

- a) It is difficult for officers in the field to find out the status of the vehicle from a member whether it has been registered and has paid dues up to a certain month. As a result of this difficulty, it is often the case that vehicles that have not paid dues freely or freely without being detected can transport their customers' cargo or goods. So that the contributions that should have gone into Aprindo's treasury did not occur.
- b) It is difficult for admins to make reports on the registration status and payment of dues for each vehicle unit in a certain period.
- c) Admins also often don't remember the ownership status of a vehicle. For example, the vehicle with the license plate number B9048BX originally belonged to PT. Buana Centra Swakarsa. But then sold to the PT. Saba Transindo and has been recorded in the Aprindo admin sheet. But it often happens when the dues collection, the B9048BX vehicle is still billed to PT. Buana Centra Swakarsa. Of course, this often causes problems in the field.
- d) It takes a long time to generate billing attachment data to members.
- e) For Aprindo management, it is not possible to quickly see the number of outstanding bills and measure which members have a good or bad level of compliance regarding their obligations in payment of contributions.

3.3 Software Development Method

The software development methodology used is the Waterfall Model which performs several steps of the activity. 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 [12]. These phases are as follows:

Analysis Phase: Often known as Software Requirements Specification (SRS) is a complete and comprehensive description of the behavior of the software to be developed.

Design Phase: It is the process of planning and problem solving for a software solution. It implicates software developers and designers to define the plan for a solution which includes algorithm design, software architecture design, database conceptual schema and logical diagram design, concept design, graphical user interface design, and data structure definition.

Implementation Phase: It refers to the realization of business requirements and design specifications into a concrete executable program, database, website, or software component through programming and deployment.

Testing Phase: It is also known as verification and validation which is a process for checking that a software solution meets the original requirements and specifications and that it accomplishes its intended purpose

Maintenance Phase: It is the process of modifying a software

solution after delivery and deployment to refine output, correct errors, and improve performance and quality.

4. RESULTS AND DISCUSSION

4.1 System Modeling

The modeling system does not use context diagrams so that it can be seen more clearly about the modules and external entities (actors) involved in the developed system. Data or activity modeling uses Data Flow Diagrams to explain the

relationship between modules or processes and the tables used.

4.1.1 UseCase Diagram

The use case diagram will give us a complete model of what the organization does, who is inside the organization, and who is outside the organization. This diagram also gives an overview of the organization so that we can focus more on a particular workflow [11].

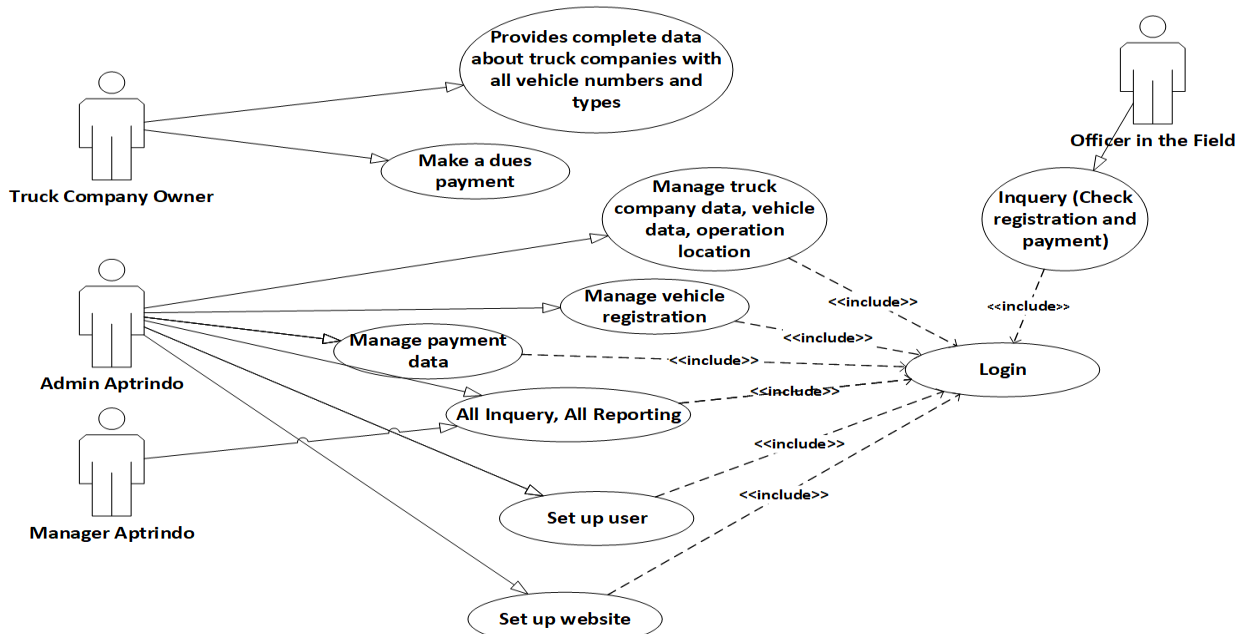


Fig 4. UseCase Diagram

4.1.2 Data Flow Diagram

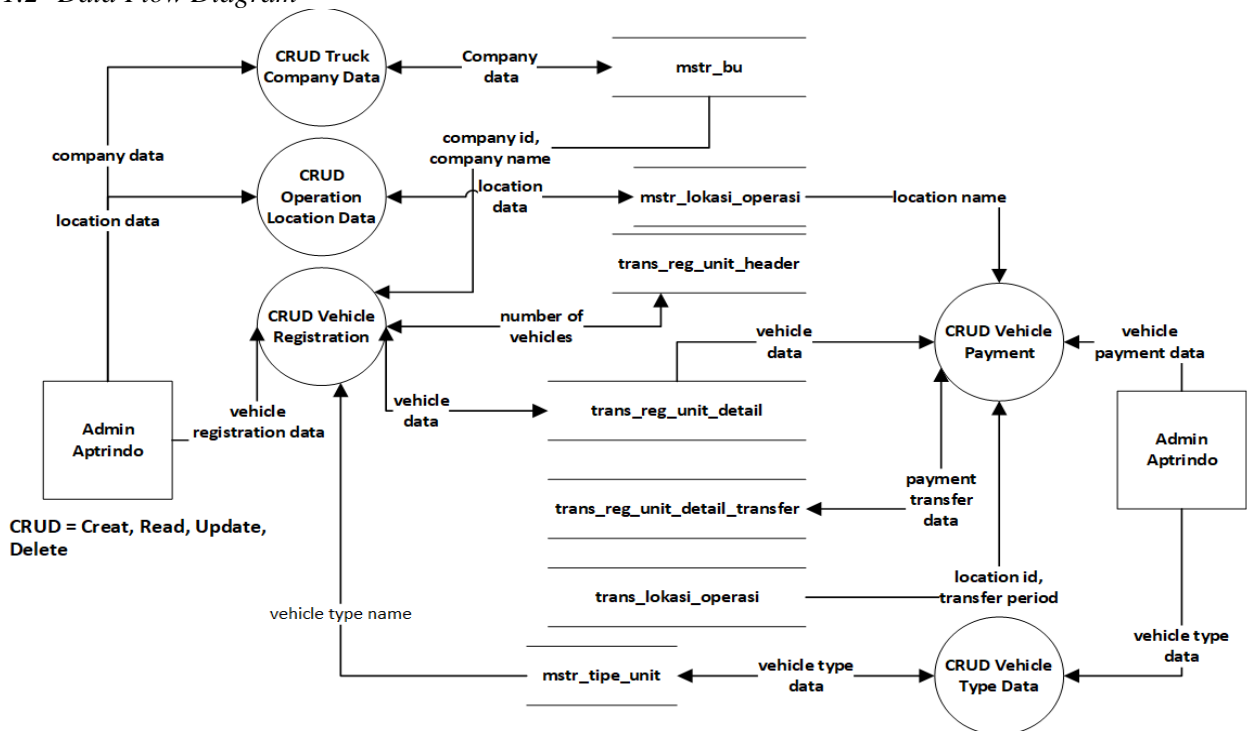


Fig 5. Data Flow Diagram (Input Data)

4.2.3 Result of System Development

Table 1. Analysis Result

NO	PROBLEM (Before the system was developed)	RESULT (After the system was developed)	DESCRIPTION	SCREEN SHOOT
1	Officers in the field find it difficult to check whether a vehicle has been registered by the owner or not	Vehicle registration status can be checked in real time	Through mobile phones, field officers can easily find out whether a vehicle has been registered by the owner or not. Menu path: Inquiry -> Cek Registrasi	Fig. 9
2	It took a long time for admin officers to find data on when the company made the last payment for all its vehicles	Admin officers can quickly retrieve data from the system to find out when was the last period a trucking company made payments for all its vehicles	Menu path: Inquiry -> Status Transfer	Fig. 10
3	Admin has difficulty in monitoring and controlling the payment of dues by the company for a certain period because there is no control system that can automatically present fast and accurate data.	Payment status in a certain period can be obtained quickly by inquiring payment status	Admin and management easily obtain payment status for each vehicle by making an inquiry	
4	Admin and management easily obtain payment status for each vehicle by making an inquiry	Admin and management easily obtain payment status for each vehicle by making an inquiry	Admin and management easily obtain payment status for each vehicle by making an inquiry	Fig. 11
5	It takes a long time to generate billing attachment data to members.	It takes a long time to generate billing attachment data to members.	It takes a long time to generate billing attachment data to members.	Fig. 11
6	Admins don't remember the ownership status of a vehicle. For example, the vehicle B9048BX originally belonged to PT. Buana Centra Swakarsa. But then sold to the PT. Saba Transindo and has been recorded in the Aprindo admin sheet. But it often happens when the dues collection, the B9048BX vehicle is still billed to PT. Buana Centra Swakarsa.	The system validates the ownership of a vehicle	The system will notify if a vehicle changes ownership	



Fig 9. Check Registration

Nama Perusahaan						
SABA TRANSINDO						
Daftar Tipe Unit						
Show	10	entries	Search: <input type="text"/>			
No	No Unit	Tipe	Tahun	Transfer Terakhir	Bukti Transfer	
1	A9041BX	Trailer	2018	2021-06-08	No. Kwitansi: C313232	
2	A9065BX	Dumptruck	2019	2021-06-21	No. Kwitansi: A101234	
Showing 0 to 0 of 0 entries						Previous Next

Fig 10. Last Payment Status

ASOSIASI PENGUSAHA TRUK INDONESIA							
DAFTAR UNIT BELUM TRANSFER SAMPAI PERIODE July-2021							
No.	No Unit	Tipe	Tahun	No KIB	Last Transfer	Bukti Transfer	Belum Transf (Bulan)
PT . SABA TRANSINDO							
1	A9041BX	Trailer	2018		08-06-2021	No. Kwitansi: C313232	1
2	A9065BX	Dumptruck	2019	-	21-06-2021	No. Kwitansi: A101234	1
Total Unit PT . SABA TRANSINDO : 2							2
PT . Ratu Melati							
3	B1059VW	Trailer	2018		29-06-2021	No. Kwitansi: B411232	1
Total Unit PT . Ratu Melati : 1							1
TOTAL BELUM DITRANSFER							3

Fig 11. Unpaid Status

5. CONCLUSION

- The use of HMVC and Framework concepts can speed up and simplify software development and maintenance.
- Vehicle Registration and Payment System speeds up the billing process to trucking companies, validates vehicles and data can be accessed in real time
- It is hoped that Aprindo Cilegon data can be integrated with the head office.

6. ACKNOWLEDGMENTS

The authors would like to thank the Ministry of Research Technology and Higher Education of the Republic of Indonesia (KEMENRISTEKDIKTI) for the financial support.

7. REFERENCES

- [1] Khaoula Benmoussa, Majida Laaziri, Samira Khouli, 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.
- [2] Awan Pribadi Basuki. 2014. "Proyek Membangun Website Berbasis PHP dengan Codeigniter. 1st ed". Yogyakarta: Lokomedia.
- [3] Betha Sidik. 2019. "Mengembangkan Framework Aplikasi Database dengan Codeigniter 3. 1st ed". Bandung: Informatika.
- [4] Muzakir, A. 2014. "Implementasi Manajemen Perpustakaan Menggunakan Framework Codeigniter (CI) dengan Teknik Hierarchical Model View Controller (HMVC)". Seminar Nasional Sains dan Teknologi Informasi 2014 (SeNASTi 2014), Hotel Sahid Jaya Makasar: 12 Mei 2014.
- [5] Suhanto, A. 2008. "Mengenal ASP.Net MVC". Indonesia. Net Developer Community.
- [6] Sandra, R., Hadi, A., Huda, Y. 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 & Informatika, vol. 5, no. 2.
- [7] Susetyo, Y., Saian, P., & Somya, R. 2018. "Pembangunan Sistem Informasi Zona Potensi Sumber Daya Kelautan Kabupaten Gunungkidul Berbasis HMVC Menggunakan Google Maps API dan JSON". Indonesian Journal of Computing and Modeling, vol. 1, no. 2, 101-107.
- [8] Akil, I. 2018. "Optimasi Modularitas dengan Hierarchy Model View Controller (HMVC) pada Framework Code Igniter". Jurnal Teknologi Informatika dan Komputer, vol. 4, no. 2
- [9] Badiyanto, Murya, Y. 2018. Project PHP Membangun Sistem Informasi Akademik dengan Framework Codeigniter. Yogyakarta: CV. Langit Inspirasi.
- [10] Jahagirdar, R., Puranik, Y. 2018. "A Review on Codeigniter". International Journal of Trend in Scientific Research and Development (IJTSRD), vol. 2, page:1124-1129.
- [11] Adi Nugroho. 2005. "Rational Rose untuk Pemodelan Berorientasi Objek". Bandung: Informatik.
- [12] Youssef Bassil. 2012. "A Simulation Model for the Waterfall SDLC". International Journal of Engineering & Technology (IJET), vol. 2, no. 5