Visit Results Reporting System from External Parties in Adira Finance Using CodeIgniter

Rezania Agramanisti Azdy¹, Muji Suwarno², Sepedin³

¹,²,³ Informatics, STMIK Palcomtech, Palembang, Indonesia
Email: rezania@palcomtech.ac.id¹, muji.s88@gmail.com², apudsepudinapud@gmail.com³

Abstract. The Palembang branch of Adira Finance has its own procedures in handling its overdue customers, namely customers who do not make installment payments, based on the time of late payment. One of the measures taken is to collaborate with third parties to make visits to overdue customers. This study aims to build a system that can be used by third parties for conveying information in the form of results of visits that have been made to overdue customers. System development is carried out by prototype method to obtain the results of a system that suits the needs of its users. The results of the study are a system that can be used by three types of users, namely admin, external party, and executor. The system can be used to divide the assignment of customer visits to the external, and each external can continue the assignment to the executor. Both external and executor can report the results of their visit to the system.

Keyword: visiting result, report, prototype, CodeIgniter

1. Introduction
PT. Adira Dinamika Multi Finance, Tbk or Adira Finance is a company engaged in automotive financing in new or used conditions. Adira Finance has approximately 201 branches spread across cities in Indonesia. This research focuses on Adira Finance located in the city of Palembang. At the Palembang branch of Adira Finance (then in this research will be called Adira Finance), there is a collection division that handles the payment of customer installments. The collection division is in charge of collecting bills for customers who are late in making payments (overdue), withdrawing units and other handling which is deemed necessary in safeguarding and saving company assets.

The collection division collaborates with third parties or commonly called external parties (which in this study will be called external) in making visits and handling overdue customers if the handling by internal parties is deemed no longer possible. Every visit result and handling of overdue customers is inputted into the system in accordance with company regulations. Recording of the results of customer visits conducted by internal employees is inputted by internal employees or the Remedial Officer itself by logging into the system by using each employee's username and password. Whereas for external input the results of customer visits are carried out by Section Head in the Account Receivable Head ¾ or ARH ¾ section because the external does not get access to the company system. ARH ¾ must contact external via telephone to get information on the results of externals’ visits. At present the process is not yet effective because there are still many externals’ visits that are not inputted into the
system. This is caused by various factors including the busyness of ARH ¾ in completing daily work and the large number of overdue customers handled by external so that ARH ¾ needs a long time to ask for information about the results of externals’ visits via telephone. If no improvements are made this can result in poor performance achievement for the collection division, it can be subject to sanctions for warning letters and deduction of incentives for ARH ¾ and can reduce the appraisal of branch offices if audit is conducted. Therefore, a service is needed that can facilitate ARH ¾ in receiving information on handling results or visits made by external parties.

Based on the problem mentioned above, a system will be made in this study that can be used by external parties to report the results of their visit to Adira Finance. Chapter 2 of this text will describe similar studies that have been conducted, Chapter 3 will explain the research methods that are carried out, Chapter 4 will discuss the results of research, and Chapter 5 contains conclusions and suggestions.

2. Related Works
The use of CodeIgniter to build systems was once carried out by Parlika et al. [1]. Parlika et al. using CodeIgniter to build a rentcar field employee information system. The system development method used is waterfall with the results of research in the form of an employee information system that can be used by the admin, field coordinator, driver, checker, and workshop mechanic.

The next research is the use of CodeIgniter to optimize the application of sales transactions conducted by Sutara [2]. The application is used to automatically record sales transactions without the need to write transactions on paper and calculate them using a calculator. System development method used was prototypes and research’s results in the form of web-based sales transaction applications.

Pratikto et al. using CodeIgniter in building geographic information systems for searching and ordering boarding houses in Yogyakarta [3]. The result of the study was in the form of a system that could provide detailed information on the availability of boarding houses, provide information on boarding houses based on criteria, and facilitate boarding houses to market their boarding houses.

CodeIgniter can also be used to build an e-commerce. Prabowo built e-commerce using CodeIgniter and applied it to miniature stores [4]. While Kosasi uses CodeIgniter to build e-commerce and apply it to the sale of typical Pontianak souvenir products [5].

3. Research Metodology
The research methodology was carried out in accordance with the chosen system development technique, namely prototype. The choice of prototype as a system development technique used is based on the development of prototype models so that users can have a prior picture of the system to be built [6], as well as the involvement of users who are expected to minimize the possibility of misunderstandings that can occur.

The research stages that will be carried out are:

1. Analysis of user needs.
   At this stage, researchers conduct discussions with users to analyze the needs of users. This stage aims to find out what functions must be in the system to suit the needs of users.

2. Make a prototype.
   At this stage researchers are developing a system prototype based on the results of the analysis that has been done in the previous stage.

3. Adjust the prototype to the user's wishes.
   At this stage researchers conduct discussions with users about the prototype that has been built. Users try to use a prototype and give feedback about the prototype.

4. Create a new system.
   At this stage the researcher makes a system based on user feedback obtained in the previous stage. Researchers can add functions and features to prototypes that have been created or even reduce them according to the user's desire.

5. Testing the system.
   At this stage, the functions and components in the system are tested using the blackbox method. After all the components and functions of the system are running properly, the system is tested for users to get feedback about the system that has been built.
6. Adjust to the wishes of the user.
   At this stage the system is adjusted to the feedback given by the user to the system that has been tried.
   Stage 5 and stage 6 can be done repeatedly until the user gives positive feedback to the system. If the construction of the system is final and no changes are made to the system, the research continues to the final stage.

7. Using the system.
   At this stage, the final system will be published on the Internet to be used by system users.

4. Discussion
The following are presented the results of the research based on the stages of research that have been conducted.

4.1. Analysis of user needs
To obtain user needs, researchers conduct discussions with ARH ¾ which was considered to represent all users who will use the system. From the results of the interview, it is known that users of the system are ARH ¾ who act as administrators, external as third parties who collaborate in overdue customer visits, and executors who are part of the external parties and tasked with visiting customers directly. The needs of each user for the system built are shown in Table 1.

| Num. | Users  | Requirements |
|------|--------|--------------|
| 1    | Administrator | 1. The application built can store administrator data, externals’ and executors’ data, branches data, power of attorneys’ (Surat Kuasa/SK) data, overdue customers’ data and visiting results’ data.  
   2. The application must have a security system before entering into the application with a username and password every time the administrator or other users will log in.  
   3. Applications built can do create, read, edit and delete.  
   4. The data displayed on the application is in accordance with the power of attorneys’ data that has been printed and given to the external.  
   5. The application that is built can display data from visits or handling carried out by an external or executor. |
| 2    | External    | 1. The application that is built can help users in conveying information about the results of visits or handling SK.  
   2. Applications can display customer data and detailed SK data.  
   3. The application built can be accessed easily by the user.  
   4. The application built can store external data, executor data and customer handling data. |
| 3    | Executor    | 1. The application that is built can help users in conveying information about the results of visits or handling SK.  
   2. Applications can display customer data and detailed SK data.  
   3. The application that is built can store data from visits and handling of customers. |

Based on the analysis of the needs obtained for each user, the researcher determines the functional needs that the system must have. The functional needs of this system are expressed in the form of use case diagrams (Figure 1).
4.2. Make a prototype
The prototype that is built has functional needs in accordance with those described in Figure 1. Each user can access the function that suits their needs. In general, the admin can process all master data on the system and can assign customer visits to external, external can add executors and continue assignment of customer visits to the executor or complete the assignment itself, so that both external and executor can add the results of the visits he has made to the system. The prototype is built in the form of a system’s user interface and an example report that the system will produce. Figure 2 (left) shows the dashboard page prototype and menu choices that can be selected by the administrator, namely the menu for managing branch data, managing user data both external and executor, managing the SK making for assignment of visits to overdue customers, viewing the list of visits, and managing overdue customer data.

Figure 2 (center) also shows the prototype of dashboard page and menu options that can be selected by external. The existing menus are managing executor data including the request function for creating accounts for executors, menu for viewing information in the form of SKs assigned to them, managing visiting results, and managing accounts. And Figure 2 (right) shows the prototype for dashboard page and menu options that can be selected by the executor, namely the menu to see the list of SKs assigned to it, manage the results of the visit, and manage the account.

4.3. Adjust the prototype to the user's wishes
The prototype results that have been made are shown to the user for evaluation and obtaining feedback from users. From the results of discussions and tests conducted by the user, the user provides input to the process of recording the results of the visit so that not only external and executor can input the results of the visit but the admin is also given the function to input the results of the visit into the application. This is because not all customers authorized to be handle by external or will be visited by the executor entirely so that it will cause a delay in the rotation process and extension of SK for
customers who have not been visited. Based on the results of the evaluation of the user, the prototype was remade in the process of recording the visiting result so that the admin can input the results of the visit.

4.4. Create a new system
At this stage the researcher encodes the system using the Model View Controller (MVC) concept. The static structure of the system is described using the class diagram that can be accessed in https://goo.gl/YUbD4j.

Application built in accordance with the prototype that has been approved by the user. Figure 3 shows the systems built for admin dashboard page (left), external dashboard page (center), and executor dashboard page (right), based on the prototype shown in Figure 2.

![Figure 3. The System Built Based on the Prototype](image)

4.5. Testing the system
Testing is performed on all component functions contained in the system using the blackbox method. A summary of the results of the tests that have been carried out is shown in Table 2.

| No. | Testing Scenario                                      | Expected Result                                                                 | Result | Concl. |
|-----|------------------------------------------------------|--------------------------------------------------------------------------------|--------|--------|
| 1.  | Admin: Click the external menu                       | The system will display a list of external data                                 | Valid  | Succeed|
| 2.  | Admin: Process added executor, edit, delete and reset password | The system will display the executor input form, edit form, delete branch data in the database and reset the executor password in the database | Valid  | Succeed|
| 3.  | Admin: Process see details and delete SK             | The system will display SK details and delete SK data in the database           | Valid  | Succeed|
| 4.  | Admin: Click the visit list menu                     | The system will display a list of visit results                                | Valid  | Succeed|
| 5.  | Admin: The process of seeing the results of the visit, delete the results of the visit and input the results of the visit | The system will display the results of the visit details, display the visit input form and delete the results of the visit in the database. | Valid  | Succeed|
| 6.  | External: Click the request executor button          | The system will display the executor request form                              | Valid  | Succeed|
| 7.  | External: Request executor process                    | The system stores the request data to the database and sends the request to the admin to be confirmed | Valid  | Succeed|
| 8.  | External: Select executor and see SK details         | The system will display a list of options for executor names and see SK details | Valid  | Succeed|
9. **External:**
   Click the input menu from the visit
   The system will display the input form resulting from the visit
   Valid | Succeed

10. **External:**
    Click the results list visit menu
    The system will display a list of visit results
    Valid | Succeed

11. **Executor:**
    Click the power of attorney list menu
    The system will display a list of power of attorney
    Valid | Succeed

12. **Executor:**
    Click the input menu from the visit
    The system will display the input form resulting from the visit
    Valid | Succeed

4.6. **Adjust to the wishes of the user**
At this stage the system evaluation is carried out together with the user. From the results of user evaluation, the application built has met the needs of users and the functional needs of the system in the process of recording the results of external visits at Adira Finance.

4.7. **Using the system**
The final system is hosted on the Internet and can be used by Adira Finance together with external parties and executors to deliver information on the results of overdue customer visits. The website can be accessed in http://sukara.me/adira/login.

5. **Conclusions and Suggestions**
Based on the results of the research that has been done, it can be concluded that:
   1. The resulting system can be used by the admin (ARH ¾), a third party who works with Adira Finance (external), and third party employees who are assigned to direct visits to customers overdue (executor).
   2. The system can be used to convey information on the results of externals’/executors’ visits to the ARH ¾.
   3. The use of the prototype method in the construction of the system produces a system that is in accordance with the wishes to meet the needs of users.

The suggestion for the development of the results of subsequent research is that the system can be integrated with the existing system in Adira Finance, so that the results of visits made by both internal and external parties can be accessed on the same system.

6. **References**
[1] R. Parlika, A. H. Mubarok, and M. S. Munir, “Rancangan Sistem Informasi Pegawai Lapangan Rentcar Menggunakan Framework CodeIgniter,” J. Inf., vol. 2, no. 2, pp. 55–60, 2017.

[2] B. Sutara, “Optimalisasi Aplikasi Transaksi Penjualan Berbasis Web Menggunakan Framework CodeIgniter (Studi Kasus: Rumah Makan Tirta Sandi Sumedang),” J. Inform. J. Pengemb. IT, vol. 2, no. 1, pp. 1–5, 2017.

[3] H. S. Pratikto, Suraya, and E. Sutanta, “SISTEM PENCARIAN DAN PEMESANAN RUMAH KOS MENGGUNAKAN SISTEM INFORMASI GEOGRAFI (SIG),” J. Scr., vol. 1, no. 2, pp. 110–119, 2014.

[4] S. Kosasi, “Perancangan Sistem E-Commerce Untuk Memperluas Pasar Produk Oleh-Oleh Khas Pontianak,” Snastia, vol. 2015, no. Oktober, pp. 110–119, 2015.

[5] D. Prabowo, “WEBSITE E-COMMERCE MENGGUNAKAN MODEL VIEW CONTROLLER ( MVC ) DENGAN FRAMEWORK CODEIGNITER Studi Kasus: Toko Miniatur,” J. Ilm. Dasi, vol. 16, no. 1, pp. 23–29, 2015.

[6] S. Mulyani, “Sistem Informasi Manajemen Rumah Sakit: Analisis dan Perancangan”, (Bandung: ABDI SISTEMATIKA), 2016.