Reverse Engineering in Student Mark Recapitulation Application

S Mauluddin¹, R Sidik²
Departemen Sistem Informasi, Universitas Komputer Indonesia, Indonesia.

E-mail: syahrul.mauluddin@email.unikom.ac.id

Abstract. This study intends to conduct reverse engineering at mark recapitulation application at the Information Systems Department, Universitas Komputer Indonesia (UNIKOM). This research is the first step of the reengineering process in the mark recapitulation application. Reengineering needed because the application of the mark recapitulation that is currently running does not meet user needs, and there is no documentation of application development. This study used the big bang approach, the reverse engineering stage follows the stages of general software reengineering model by analyzing three things, analysis of application functions, analysis of user activities with applications and database analysis. The results of this research are documentation of application function requirements, documentation of user activities with application and database documentation.

1. Introduction
The Information System Department has an application to recapitulate student mark data to manage student mark data. Currently, the application is still being used, but there are important features that cannot be running properly. Besides the use of the latest operating systems, this causes the application to not work properly. Therefore, the application does not support department needs. Another problem is that there is no application documentation.

To overcome these problems, reengineering is needed to develop applications according to new needs. In some literature, reengineering is defined as The process of adjusting a system that exists to changes in its environment or changes in technology without having to change its overall functionality. However, modification and future development of an existing system. Repairing a system through reverse engineering and restructuring followed by forwarding engineering [1]. The initial stage of the reengineering process is reverse engineering. Reverse engineering is the opposite of the software development stage. The software development process itself consists of several stages with general stages starting from analysis-coding-design-testing-implementation, with various variations depending on the software development methodology such as waterfall, iterative, Rational Unified Process (RUP), Extreme Programming (XP), etc. [2]. Based on the results of previous research, the results of reverse engineering can be adjusted to the needs of the type of documentation. There are in the form of use case diagrams, use case realization, and sequence diagrams. In 2008, Eko K. Budiardjo and Yanti Andriyani conducted research in the form of rearranging the software requirements specification (SRS) of academic information systems with a reverse engineering approach. The reverse engineering results in the form of SRS are described through use cases and use case realization [2]. According to Indra Kharisma Raharjana and Army Justitia conducted research in the form of model sequence diagrams by reverse engineering database applications on smartphones to maintain consistency in software design. The results of reverse engineering applications are in the form of sequence diagrams [3].

This study intends to conduct reverse engineering at mark recapitulation application at the Information Systems Department, Universitas Komputer Indonesia (UNIKOM). This research is the first step of the reengineering process in the application of brand recapitulation. Re-engineering is needed because the brand recapitulation application that is running does not meet user needs and there is no documentation of application development. This study uses the big bang approach, which is the...
reverse engineering stage following the steps of the general software reengineering model by analyzing three things, analyzing application functions, analyzing user activities with applications and database analysis. The reverse engineering approach must consist of 3 stages namely, (Extraction) In this phase, information from the source code, documentation, and documented system history extracted (eg, bug reports, administrative data) can be changed later (Abstraction) In this phase, the information extracted abstracted according to the purpose of reverse engineering activities. Abstraction must bring large amounts of information extracted to a manageable size. and finally (Presentation) In this phase, abstract data is transformed into a comfortable presentation for users [4].

2. Method

2.1 Approach method

There were three approaches in software engineering [5-6]:

1. Big Bang approach

The "Big Bang" approach, is an approach that replaces the entire system at once, as shown in Figure 1.

![Figure 1. Big Bang Approach](image)

This approach is often used by projects that require direct problem solving rather than migration to another system architecture.

2. Incremental approach.

The incremental approach is shown in Figure 2.

![Figure 2. Incremental or Phase-out Approach](image)

In this approach, parts of the system are being reworked and added incrementally as a new versions of the system needed to meet the new goals. This project is divided into reengineering sections based on existing system parts.

3. Evolutionary Approach

In the "Evolutionary" approach, as in the Incremental approach, parts of the original system are replaced with parts of the newly re-engineered system. However, in this approach, the parts are chosen based on their function, not on the existing system structure. The target system was built using functionally cohesive parts as needed. The Evolutionary Approach allows developers to focus on re-engineering efforts on functional objects wherever tasks are in the current system. The Evolutionary approach can be seen in Figure 3.
In this research, the old mark recapitulation application will be changed on both application platform and the database. So, that the approach method used is the big bang approach.

2.2 Reverse Engineering Stage

Reverse engineering is the process of analyzing a system with the aim to identify system components and the relationship/interaction between components and representing the system in a form that has a higher level of abstraction [7]. The usefulness of reverse engineering is to do program analysis, get architecture back, and the design of the UML model of the application, as well as visualization. In addition, reverse engineering can also be used to create artifacts such as design models to support software development [8]. Reverse engineering in software can be grouped into two, static and dynamic, where static if the model is extracted from source code and dynamic if the model is obtained from program execution [9]. Static reverse engineering is suitable for use in software development that serves to ensure the correctness of implementation based on design, whereas reverse engineering is dynamically extracted without the need to know the source code and can retrieve information about memory management, code coverage, and concurrency in the program [3].

The reverse engineering stage in this study refers to the general reengineering software model, as shown in Figure 4.
Based on Figure 4, the core of this reverse engineering activity is to re-identify the existing system to obtain an SRS (Software Requirement Specification). In this research the proposed reverse engineering stages are as follows:

1. Analysis of applications to identify application functions. The application function will be described through a use case diagram.
2. Analysis of user activities with applications that illustrated through the activity diagram
3. Data Analysis to identify database structure. The database structure illustrated through table relationships.

3. Results and Discussion
Reverse engineering is started by analyzing the application to identify application functionality in accordance with the features in the application. Based on the results of the analysis that the mark recapitulation application made using visual basic 6. The function of application shown in Figure 5.

![Diagram of Student Mark Recapitulation Application]

Figure 5. The functionality of Mark Recapitulation Application
Figure 5 shows that the application consists of nine functions. After carrying out the analysis there are two functions that cannot be used, namely import marks and print a pass certificate. The import mark function cannot be used because of changes in the excel file format while the print function of the pass certificate cannot be used because of curriculum changes. The next step is analyzing user activity on the application of each use case. As an example, see Figures 6 and 7 describe an activity of diagrams for use case import marks and mark management.

**Figure 6. Activity Diagram of Import Mark**

**Figure 7. Activity Diagram of Mark Management**
The next step is analyzing the database. The database used at mark recapitulation application is MySQL and way to integrate application with database is using ODBC (Open Database Connectivity). Illustration of mark recapitulation application is shown in Figure 8.

![Figure 8](attachment:image.png)

**Figure 8.** Database of Mark Recapitulation Application

Figure 8 shows that the database of mark recapitulation application has five tables of student, mark, subject, lecturer and system. System table has no relation with other tables. System table data is used to approve and produce document number of academic transcript and certificate of graduation.

4. **Conclusion**

After conducting reverse engineering on the application, it was concluded that the application of mark recapitulation had nine functions that had to be included: lecturer management, curriculum management, student management, mark import, mark management, system regulation, temporary transcript printing, final transcript printing, and graduation certificate printing. The database consists of five tables: table of the student, mark, subject, lecturer, and system. Description of the needs of this system is expected to be a reference in the forward engineering process.
References

[1] Herdiana, B., and Sanjaya, I. F. 2018. Implementation of telecontrol of solar home system based on Arduino via smartphone. In *IOP Conference Series: Materials Science and Engineering*, **407**(1), p. 012088. IOP Publishing.

[2] Budiardjo, E. K., and Andriyani, Y. 2008. Penataan Ulang Software Requirement Specification (SRS) Sistem Informasi Akademik dengan Pendekatan Reverse Engineering. In *Proceedings of Seminar Nasional Aplikasi Sains dan Teknologi (SNAST)*, 150-156.

[3] Raharjana, I. K., and Justitia, A. 2015. Pembuatan Model Sequence Diagram dengan Reverse Engineering Aplikasi Basis Data pada Smartphone untuk Menjaga Konsistensi Desain Perangkat Lunak. *JUTI: Jurnal Ilmiah Teknologi Informasi*, **13**(2), 133-142.

[4] Stringfellow, C., Amory, C. D., Potnuri, D., Andrews, A., and Georg, M. 2006. Comparison of software architecture reverse engineering methods. *Information and Software Technology*, **48**(7), 484-497.

[5] Wibiyanto, A., and Afrianto, I. 2018. QR code and transport layer security for licensing documents verification. In *IOP Conference Series: Materials Science and Engineering*, **407**, 012069.

[6] Majthoub, M., Qutqut, M. H., and Odeh, Y. 2018. Software Re-engineering: An Overview. In 2018 8th International Conference on Computer Science and Information Technology (CSIT) (pp. 266-270). IEEE.

[7] Utama, J., and Saputra, M. D. 2018. Design of electric wheelchair controller based on brainwaves spectrum EEG sensor. In *IOP Conference Series: Materials Science and Engineering*, **407**, 012080.

[8] CanforaHarman, G., and Di Penta, M. 2007. New frontiers of reverse engineering. In 2007 *Future of Software Engineering* (pp. 326-341). IEEE Computer Society.

[9] Di Lucca, G. A., Di Penta, M., Antoniol, G., and Casazza, G. 2001. An approach for reverse engineering of web-based applications. In *Proceedings Eighth Working Conference on Reverse Engineering* (pp. 231-240). IEEE.

[10] Hirawan, D., and Sidik, P. (2018, August). Prototype Emission Testing Tools for L3 Category Vehicle. In *IOP Conference Series: Materials Science and Engineering*, **407**(1), p. 012099. IOP Publishing.