Designing a Student Performance Monitoring Dashboard Application Using the MVC Method and the Spring Boot Framework at Media Nusantara Citra University

Adi Fitrianto[1], Donny[2]
[1,2]Universitas Media Nusantara Citra, Jakarta Barat, Indonesia
Email: adi.fitrianto@mncu.ac.id

Abstrak

The student academic performance monitoring dashboard is an application used by lecturers at the Faculty of Business and Education, Media Nusantara Citra University to digitally monitor student performance. The monitoring process currently carried out is still manual, which involves both data searching and manual file searching. Apart from that, this data can only be accessed by the academic department and lecturers must make a request first. On the other hand, the large amount of work and busyness of lecturers means that lecturers do not have much time in the process of collecting data for monitoring, while the monitoring process is very important for lecturers to know the progress and results of the learning process of the lecturers' students. Model, View, Controller (MVC) is an application development architecture that emphasizes 3 important components. These three components each have a focus of attention, responsibility and logic so as to speed up performance. For this reason, to solve the problem of monitoring student academic performance, a dashboard application was built using the MVC method. This monitoring dashboard application was built using the Java programming language with the Spring boot framework and is web-based. Spring boot is a framework for the Java programming language which is equipped with a Tomcat server, Annotation technology, and Maven as a Build Manager. With this application, it is hoped that it will make it easier for lecturers to carry out the monitoring process so that the monitoring process can be carried out anytime and anywhere. The output targeted in this research is a Student Academic Performance Monitoring Dashboard application. This research has a Technology Readiness Level (TKT) at level 2 which is expected to be useful for lecturers at Media Nusantara Citra University. This research also targets additional output, namely publication in a national journal indexed at least SINTA 4 with a target publication in November 2023.

***Keywords**: Application Dashboard, Academic Performance Monitoring, MVC, Spring boot Framework.*

– – – – – – – – – – ◆ – – – – – – – – – –

A. INTRODUCTION

Student academic performance is the result of teaching and learning activities that have been participated in by students and lecturers after carrying out lecture activities. Student academic performance is generally measured by holding ongoing exams. Monitoring student academic performance is very important because it provides useful information for students and faculty in monitoring student progress and achievements in learning (Sihombing, Aryadita & Rusdianto, 2019).

The author conducted a survey in the form of a questionnaire to lecturers to find out whether lecturers experienced problems in monitoring the academic performance of their students. The data obtained is that the majority of lecturers

experience several obstacles when carrying out the process of monitoring academic performance, namely the absence of practical tools or tools for viewing academic performance monitoring data, the absence of presenting data in the form of visuals with graphs that are easy to read for monitoring student academic performance, data What is needed to monitor academic performance is still in separate tools, not yet on one screen, so it requires efforts a lot of clicking.

The aim of this research is to design an application that has a dashboard feature to monitor student academic performance, and build an application that has a dashboard feature for web-based monitoring of student academic performance.

The student academic performance monitoring dashboard application was built using the Java programming language and Model View Controller (MVC) as the programming method. The MVC method is a programming technique that separates code into 3 parts, namely model, view, and controller (Pa & Karim, 2017). The model section handles the interaction between the application and the database, the view section handles the interface logic and the controller section manages the interaction between the view, model and logic on the backend side (Wijaya & Christian, 2019).

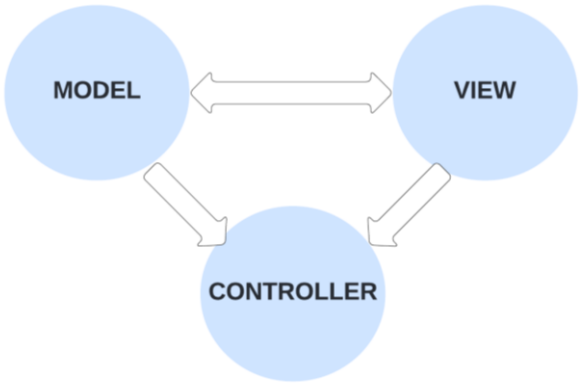

Figure 1. MVC Framework's Workflow

1. Model
   The model is the part of the program code that handles the database, the contents of the model are the parts (functions) that are directly related to the database.
2. View
   This section contains all the details of the user interface implementation. View is a part of the program code that controls the appearance of the website.
3. Controller
   The controller contains commands that are responsible for processing data and sending it to a web page. The function of the controller is to receive requests and data from users and then determine what will be processed by the application (Wijaya & Christian, 2019).

The MVC programming method is widely used to create web-based applications such as book room design applications (Ulfa, Suryayusra & Hardini, 2020), and academic performance monitoring dashboard applications (Triono,

Hakim & Amelia, 2018), to production monitoring applications. The MVC programming method has several advantages, namely the use of a controller which is the logic center so that if there is a process change it will not touch the interface, there is a separate view section making it easier to integrate with other sections, and there is a model section that can provide detailed data placement and not spread throughout the application. Apart from that, the MVC method can also help developers focus more on working on an application so that it is easier to detect errors (Triono, Hakim & Amelia, 2018). In research Wijaya & Christian (2019), the MVC programming method was used to design a monitoring dashboard application for student academic performance. Data visualization in the form of graphics is done in the view section using an external library while the numerical data is done in the controller section. This makes it easier for developers to separate which parts will be worked on first. Not only that, the MVC programming method is also used to build a book room system where part of the model is used to map data to tables in the database (Saputra, Husein & Qana'a, 2021). Apart from that, the use of the MVC method is also used to design web-based health clinic information systems (Ilhamsyah & Rahmayudha, 2017).

B. METHOD

The systematic literature review (SLR) method was used in this research in order to prove theory and determine solutions. Researchers use the SLR method with the aim and consideration that this case study must be reviewed first before further processing is carried out. On the other hand, SLR is carried out to seek and obtain knowledge from previous studies regarding benefits, concept proposals, implementation challenges, use of technology, and updates from similar research.

The following are the steps or work flow of a systematic literature review that can be seen in figure 2.

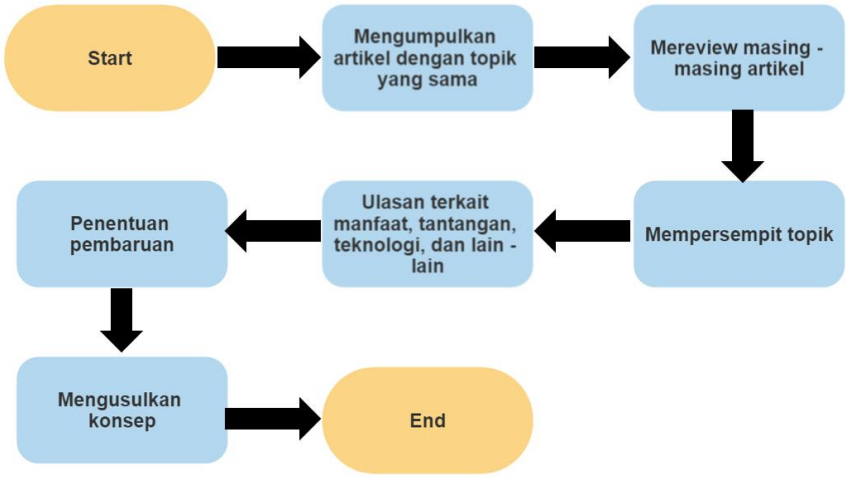

Figure 2. Sistematic Literature Review's Workflow

In the SLR method, the first step is to collect papers with dashboard application keywords. Then do a review of each article/paper that has been collected. After that, we search and collect related articles/papers again by narrowing down

the topics which include MVC programming methods and the Springboot framework. Then we did another review to find something that had not been discussed before to develop a similar study (Triono, Hakim & Amelia, 2018).

As for the stages, this research has several stages starting from conducting an interview, then conducting a literature study, then designing the application, then coding, and the final stage is testing the application. The research stages can be seen in Figure 3 below.

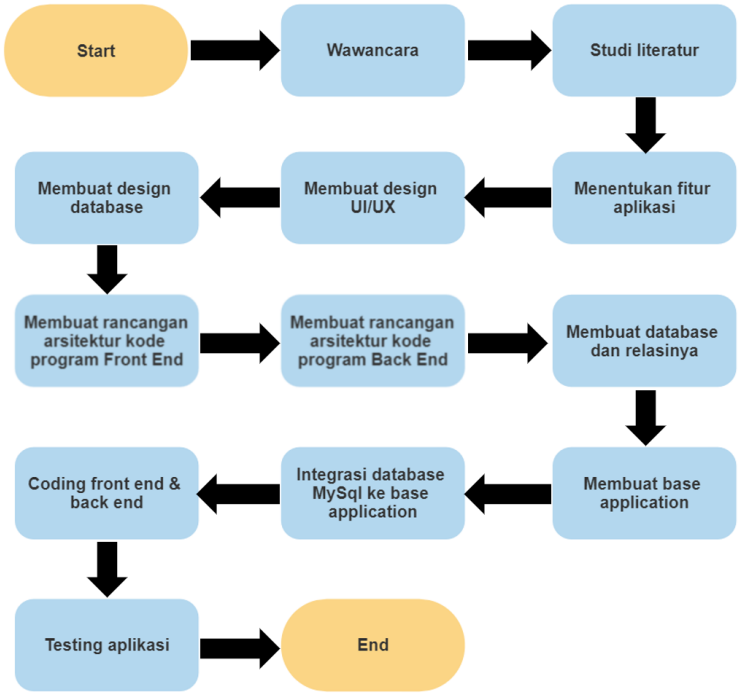

Figure 3. Research Stages

Based on the diagram above, that can be seen that this research has several stages starting from interviews, then continuing to literature study, then determining what features are in the application, after that proceeding to creating UI/UX, creating a database and architecture, creating a base application, then ends with the application testing process.

To design the appearance of the application that will be built, the following is the raw user interface design for the student academic performance monitoring dashboard application that will be built in this research. This design is a design that is still raw or draft as an illustration of the application that will result from this research proposal.

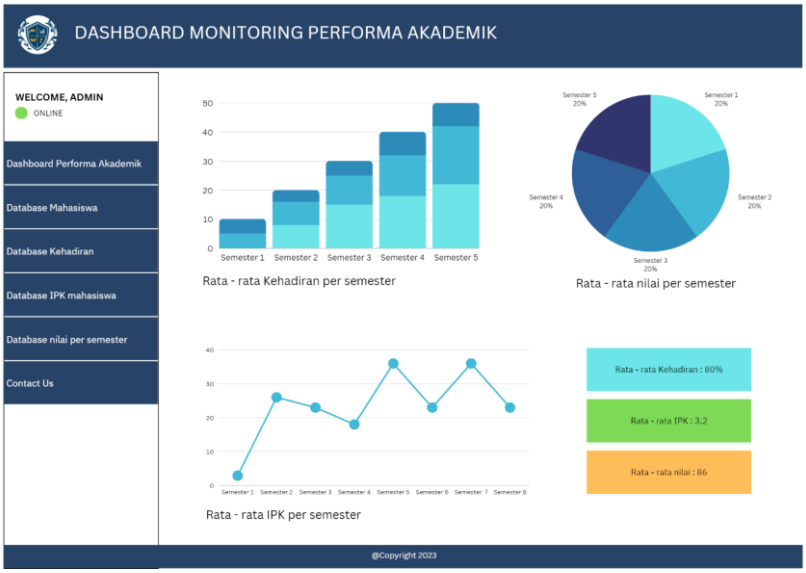

Figure 4. Raw Design of User Interface for Academic Performance Monitoring Dashboard Application

This application will have several features, namely: average GPA per semester, average student attendance per semester, average student grades per semester, student status, occupation of student parents, and education of student parents. These features will later be built according to user needs.

C. RESULT AND DISCUSSION

In designing this application, it has 2 users, namely Admin and Lecturer. Admins have access to use the dashboard viewing feature, manage student data, manage study program data, courses, student attendance, student grades, and student GPA. Lecturer users have access to use the dashboard viewing feature, view student data, courses, study programs, grades, GPA and student attendance. This application is made according to needs where lecturers can see the data needed for the study program while the admin can manage the data. For more brevity, you can see the use case diagram in Figure 5.

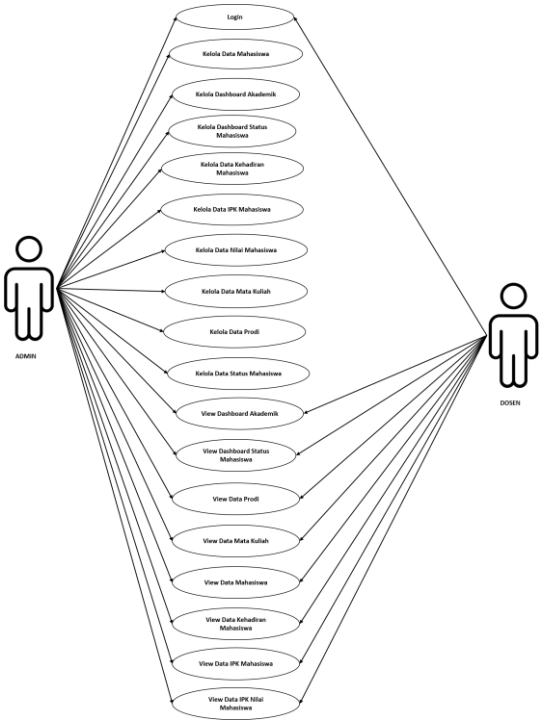

Figure 5. Use Case Diagram

The dashboard application that is built has several entities, namely students, study programs, courses, GPA, grades, and attendance. These entities are the basis for creating data structures in the application. The following is a brief explanation of the entities used in this application using the Entity Relationship diagram.

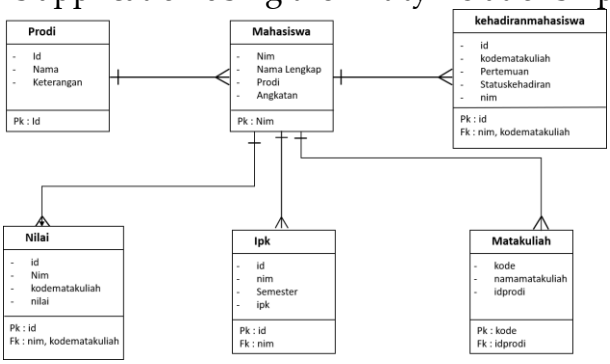

Figure 6. Entity Relationship Diagram

This research resulted in a prototype dashboard application for monitoring student academic performance. After carrying out the feature analysis process, UI/UX design, and application creation using coding techniques, an application prototype was formed with the following features.

1. Login

   The login feature is used to validate the username using password input for each user. This feature is also used for security so that not all parties have access to the application.

2. Student Academic Dashboard

   This feature is used by lecturers in each study program to monitor the academic performance of their students. This feature contains information needed by lecturers starting from student summary data, GPA data summary,

student status summary, to student grade data summary. The academic performance dashboard features can be seen in Figure 7.

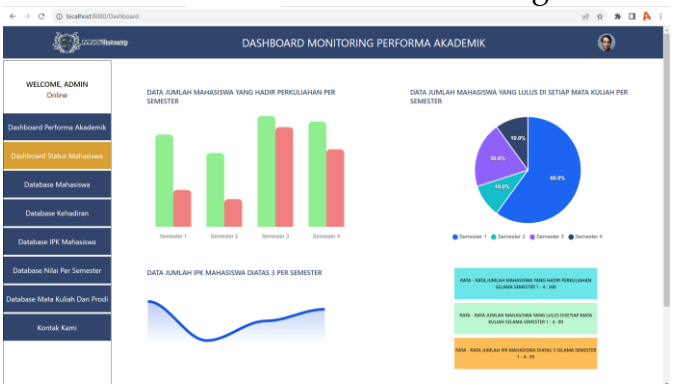

Figure 7. Academic Performance Dashboard Feature

a. Student Status Dashboard

This feature has summary information from data on the number of students who are still active and inactive, the percentage of students' parents' latest education, and the percentage of students' parents' current employment. This feature provides diagrams with different colors to make it easier for lecturers to read the available data. The student status dashboard feature can be seen in Figure 8.

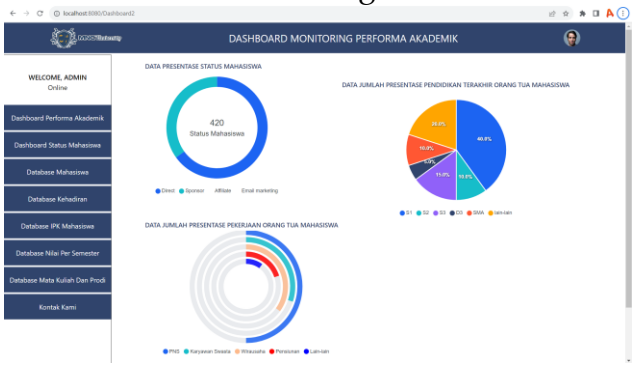

Figure 8. Student Status Dashboard Feature

b. Student Management

This feature functions to carry out the process of managing student data. The student data includes Nim, full name, gender, chosen study program, class, parents' occupation, parents' highest level of education, and student activity status. The management process includes adding data, updating or editing data, viewing data, and deleting data. This feature can be seen in Figure 9.

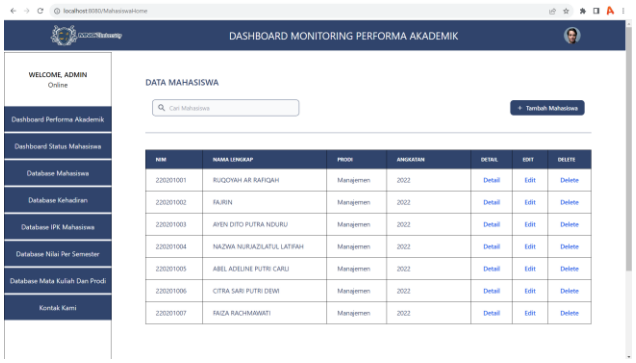

Figure 9. Student Management Feature

3. Manage Subject and Major Data

Apart from managing student data, this application also provides features for managing other data. The course and study program data management feature functions to manage data related to courses and study program data. The data managed includes course codes, course names, study program codes, and study program names. This data will later be mapped with students taking courses so that their grades can be input. So, the dashboard feature can display grades according to the courses taken by students. This feature can be seen in Figure 10.

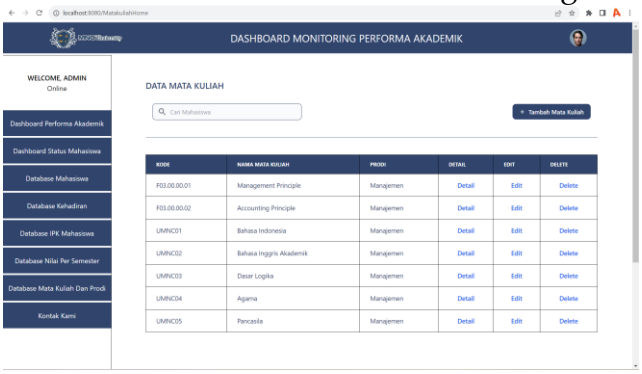

Figure 10. Features Manage Subject and Major Data

4. Manage Student GPA Data

This feature functions to carry out the process of managing student GPA data. The data managed includes assessment ID, NIM, semester, and GPA. NIM is needed so that student data can be mapped against the GPA table. Meanwhile, the semester column is needed to indicate the student's GPA in a particular semester. This feature can be seen in Figure 11.

Figure 11. Manage Student GPA Data Feature

5. Manage Student Attendance Data

This feature functions to manage student attendance data. The management process includes adding data, editing / updating data, viewing data, and deleting data. Managed attendance data includes attendance ID, course code, meeting, attendance status, and ID. The course code is needed for mapping attendance transactions and courses taken by students. Meanwhile, nip is needed for mapping attendance data and student data. This feature can be seen in Figure 12.

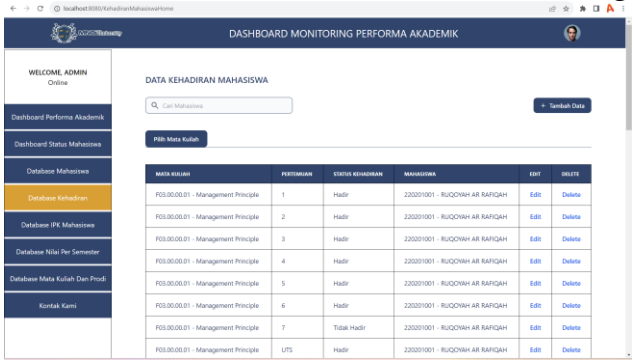

Figure 12. Manage Student Attendance Data Feature

6. Manage Student Grade Data

Apart from managing attendance and GPA data, there is also a feature that provides management for student grade data. This feature functions to carry out the CRUD (Insert, update, view and delete) process of student grade data which, if necessary, will later be displayed in the academic performance dashboard feature. The data managed in this feature includes assessment ID, nip, course code and grades. Nip and course code are needed to map grades to courses taken by students. This feature can be seen in Figure 12.

Figure 11. Student Grade Data Management Feature

D. CONCLUSION

This research produces an application with the main feature of a dashboard for monitoring academic performance activities. This application was built using the Java programming language and the SpringBoot framework and MySql for the database. Based on the discussion above, it can be concluded that this application has features that can support lecturer monitoring activities on student academic performance. This monitoring activity can be carried out using the dashboard feature.

REFERENCES

1. Afreza, I. N. (2022, November). Rancang Bangun Sistem Informasi Monitoring Dan Evaluasi Akademik Siswa Berbasis Website. In *Prosiding Seminar Nasional Teknologi Informasi dan Komunikasi (SENATIK)* (Vol. 5, No. 1, pp. 195-204).
2. Arimbawa, I. W. A., Wantari, N. K. D. F., & Husodo, A. Y. (2018). Rancang Bangun Sistem Monitoring Akademik Program Studi Teknik Informatika Universitas Mataram Menggunakan Data Warehouse. *Journal of Computer Science and Informatics Engineering (J-Cosine)*, *2*(1).
3. Carrillo, R., Renaud, C., Prié, Y., & Lavoué, E. (2017, July). Dashboard for monitoring student engagement in mind mapping activities. In *2017 IEEE 17th International Conference on Advanced Learning Technologies (ICALT)* (pp. 433-437). IEEE.
4. Hariyanti, E., & Purwanti, E. (2014). Perancangan Sistem Dashboard Untuk Monitoring Indikator Kinerja Universitas. *SESINDO 2014*, *2014*.
5. Horeb, A. (2023). Perancangan Dashboard Untuk Memantau Kinerja Dosen Fakultas Teknologi Informasi di Universitas Tarumanagara. *Jurnal Ilmu Komputer dan Sistem Informasi*, *11*(1).
6. Ilhamsyah, I., & Rahmayudha, S. (2017). Perancangan Model Dashboard Untuk Monitoring Evaluasi Mahasiswa. *Jurnal Informatika: Jurnal Pengembangan IT*, *2*(1), 13-17.
7. Kusmanto, T. H., Butar, F. T. S. B., & Irawan, A. (2017). Implemestasi Web Service Dalam Layanan Sistem Informasi Akademik Siswa Pada Perangkat Mobile Android. *URECOL*, 219-222.
8. Megawaty, D. A. (2020). Sistem Monitoring Kegiatan Akademik Siswa Menggunakan Website. *Jurnal Tekno Kompak*, *14*(2), 98-101.
9. Pa, N. C., & Karim, F. (2017, October). Dashboard system for measuring green software design. In *2017 3rd International Conference on Science in Information Technology (ICSITech)* (pp. 325-329). IEEE.
10. Saputra, F. A., Husein, I. G., & Qana'a, M. (2021). Perancangan Dashboard Untuk Monitoring Performa Mahasiswa D3 Sistem Informasi Fakultas Ilmu Terapan Berbasis Web. *eProceedings of Applied Science*, *7*(6).
11. Sihombing, W. W., Aryadita, H., & Rusdianto, D. S. (2019). Perancangan Dashboard Untuk Monitoring Dan Evaluasi (Studi Kasus: FILKOM UB). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, *3*(1), 434-441.
12. Susilo, W. (2020). Perancangan Bangun Aplikasi Dashboard Untuk Monitoring Produk Berbasis Web. *Engineering and Technology International Journal*, *2*(02), 139-149.
13. Triono, T., Hakim, Z., & Amelia, R. (2018). Perancangan Aplikasi Dashboard Pengelolaan Hasil Produksi Departemen Finishing Berbasis Web Pada PT Panarub Industry. *Jurnal Sisfotek Global*, *8*(2).
14. Ulfa, M., Suryayusra, S., & Hardini, S. (2020). Penerapan Model View Controller (MVC) Untuk Perancangan Sistem Ruang Buku Indonesia. *CESS (Journal of Computer Engineering, System and Science)*, *5*(1), 53-56.

15. Wijaya, K., & Christian, A. (2019). Implementasi Metode Model View Controller (MVC) Dalam Rancang Bangun Website SMK Yayasan Bakti Prabumulih. *Paradigma*, *21*(1), 95-102.