Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 4, **Number** 2, November 2024
https://doi.org/10.47709/brilliance.v4i2.5011

Implementation of Docker Platform for Integration Of Thesis Information System And Web-Based Thesis Title Checking Information System In Informatics Study Program

Rizky Adi Ryanto[1], M. Julkarnain[2*], Yuliadi[3], Nora Dery Sofya[4], Siska Atmawan Oktavia[5]
[1,2,3,4,5]Informatika, Sumbawa University of Technology, Sumbawa, Indonesia
[1]adiryantorizky140820@gmail.com, [2*]m.julkarnain@uts.ac.id, [3]yuliadi@uts.ac.id, [4]nora.dery.sofya@uts.ac.id, [5]siska.atmawan.oktavia@uts.ac.id

***Corresponding Author**

Article History:
Submitted: 27-11-2024
Accepted: 30-11-2024
Published: 13-12-2024

Keywords:
System Integration; Qualitative Methodology; Docker Platform; Software Development Life Cycle (SDLC).

Brilliance: Research of Artificial Intelligence is licensed under a Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0).

ABSTRACT

The use of technology-based information systems not only improves operational efficiency but also enriches the learning and teaching experience. In this context, the management of theses, as an essential part of higher education programs, is one area that receives special attention. Based on the results of the observations, the Informatics Study Program registration process and scheduling of proposal seminars and final project trials are currently carried out conventionally. Meanwhile, the Informatics Study Program already has two previous systems, namely the thesis title checking system and the registration system and final project scheduling. However, they still need to be integrated and implemented on the server so the registration process is carried out conventionally. This study uses the Docker platform to discuss integrating the thesis information system and title-checking system in the Informatics Study Program at the Sumbawa University of Technology. The main goal is to improve operational efficiency and user experience, especially for students and teaching staff. This study covers various steps through the Software Development Life Cycle (SDLC) approach and qualitative methods, from analyzing the running system and proposals to implementing Docker as a containerization platform. This integration brings significant changes, uniting the thesis title-checking and registration processes in one integrated platform. The results show increased administrative efficiency, more accessible access, and more focused use. However, this study also encourages further development for system maintenance, data security, and improvements according to user feedback to improve system functionality and resilience in the future.

INTRODUCTION

The digital era has significantly changed various sectors, including higher education. Today, universities worldwide are transforming and managing their academic and administrative processes, thanks to advances in information technology. The use of technology-based information systems not only improves operational efficiency but also enriches the learning and teaching experience. In this context, thesis management is an integral part of higher education programs and is one area that receives special attention (Danur Wijayanto et al., 2021)(Fatihah, 2019)(A. Ikegami, I.D.M.B.A. Darmawan, 2022)(Mochammad Reza Ramadhan , Lukito Edi Nugroho, 2017).

At the University of Technology Sumbawa, especially in the Informatics study program, Faculty of Systems Engineering, the thesis management process covers various aspects, from title registration and guidance to final assessment. Each of these aspects requires proper coordination and data management. On the other hand, checking the thesis title is a crucial step to ensure originality and avoid topic duplication. However, challenges arise when the thesis management and title-checking systems operate separately. This leads to problems such as data redundancy, difficulty synchronizing information, and increased workload for staff and students (Rahman & Ningsi, 2022)(R. Wijaya et al., 2021).

The author's interview with the Secretary of the Informatics Study Program shows that the registration process and scheduling of proposal seminars and final project trials are currently carried out conventionally. Meanwhile, the informatics study program already has two previous systems: the thesis title-checking system and the final assignment registration and scheduling system. However, they must be integrated and implemented on the server so the registration process is still conventionally carried out.

Docker offers an effective solution to this integration problem as an innovative containerization platform. This platform allows applications to run in isolated containers, ensuring consistency in the operational environment regardless of infrastructure differences (Parama, 2022)(Fihri et al., 2019)(Mubarok & Saptono, 2021)(Bik & Asmunin, 2017). Using Docker in an academic context, especially for information system integration, can provide several advantages. These include increased efficiency in application deployment, ease of version management, and reduced

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 4, **Number** 2, November 2024
https://doi.org/10.47709/brilliance.v4i2.5011

system conflicts due to differences in development and production environments.

Furthermore, the application of Docker in integrating the Thesis Information System and the web-based Thesis Title Checking System in the Informatics Study Program is expected to increase operational efficiency. By combining these two systems, it is hoped that there will be an increase in data accuracy, reduced time in administrative processes, and a more compelling user experience for students and teaching staff. In addition, this study also seeks to explore the potential of Docker in facilitating broader usage scenarios in the academic environment.

Therefore, this study aims to address the integration challenges between these two essential information systems and demonstrate how containerization technology can be effectively adopted in academic management. Through the implementation of Docker, this study is expected to provide valuable insights into the ease of integration, scalability, and resource management in complex computing environments such as universities.

METHOD

Research Methods

In this study, the author uses two methods, namely data collection methods and software development methods. The type of research used is qualitative research. The stages in this study can be seen in the following flowchart (Wiki Nasmansyah, 2023) (Apriansyah & Julkarnain, 2023):

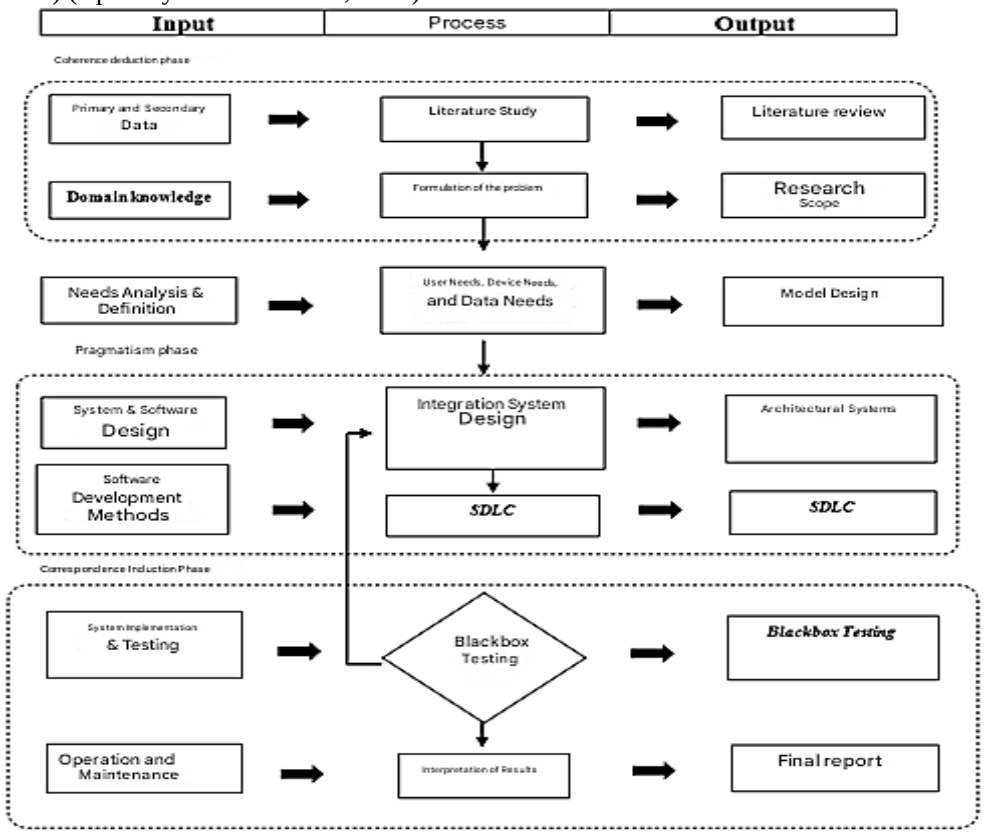

Figure 1. Research Methods

Based on Figure 1, the data collection method used in this study is qualitative and includes several approaches. First, observation is used to record the needs in implementing the Docker platform to integrate the thesis information system and the thesis title-checking system in the UTS Informatics study program. Second, interviews were conducted with the Secretary of the UTS Informatics Study Program to obtain direct data through questions and answers. In addition, documentation is used to collect information and important notes related to the thesis information system and thesis title checking in the Informatics study program. Finally, literature study is an approach taken to refer to related references from books, e-books, theses, articles, and internet sources relevant to the research topic. This approach aims to support thinking and obtain appropriate information in preparing research programs and reports (Bik & Asmunin, 2017).

Development Method

In this study, the software development method applied is the Software Development Life Cycle (SDLC) (Pressman, 2015) (A. F. Wijaya & Eliezer, 2019), which consists of several stages that can be seen in Figure 2.

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 4, **Number** 2, November 2024
https://doi.org/10.47709/brilliance.v4i2.5011

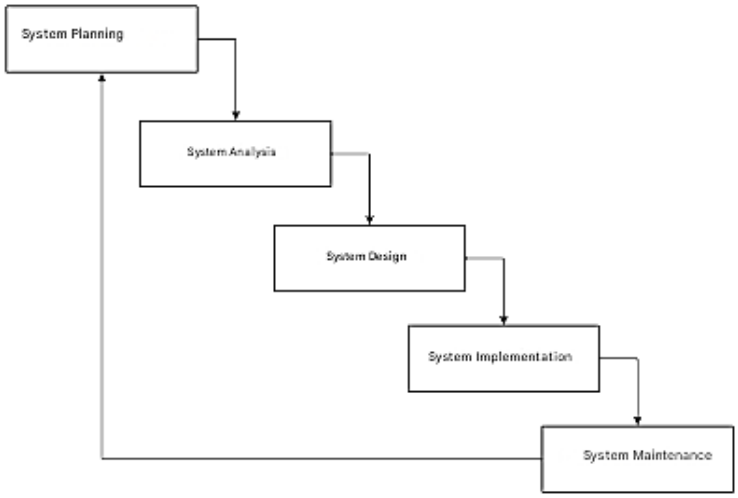

Figure 2. Software Development Life Cycle (SDLC)

Based on Figure 2, the first stage is system planning, which involves collecting data through observations in the Informatics study program, interviews with the Secretary of the Informatics Study Program, documentation related to scheduling in the study program, and literature studies from various relevant sources. The second stage is the subsequent system analysis, which reveals several things, including the non-implementation and non-integration of the two information systems, differences in the stacks used, and the presence of bugs in the two systems. The third stage is system design, which then utilizes Docker as a deployment platform and REST API as a tool for integrating the two systems (Isron & Putra, 2021). It continued with the system implementation stage, which involved coding the previously designed system using Visual Studio Code as a text editor, VPS, and Docker as a system deployment platform. The last stage, namely the system maintenance stage, is carried out to fix bugs and errors reported by users during the use of the developed software (Setiyani, 2019).

RESULT

System Planning

System design is the initial stage in software or information systems development that is the basis for the following stages. This stage includes creating concepts, structural designs, and technical specifications to ensure the system meets user needs and predetermined goals. This design activity is carried out by collecting data through observation, interviews, and literature studies to obtain the information needed for development. System design focuses on how system functions will be implemented, from architectural design and user interface to data models. The results of this system design become a guide for realizing an efficient, scalable, and easy-to-manage system.

System Analysis

a. Current System Analysis

The following is a current system analysis of the information system developed by the author:

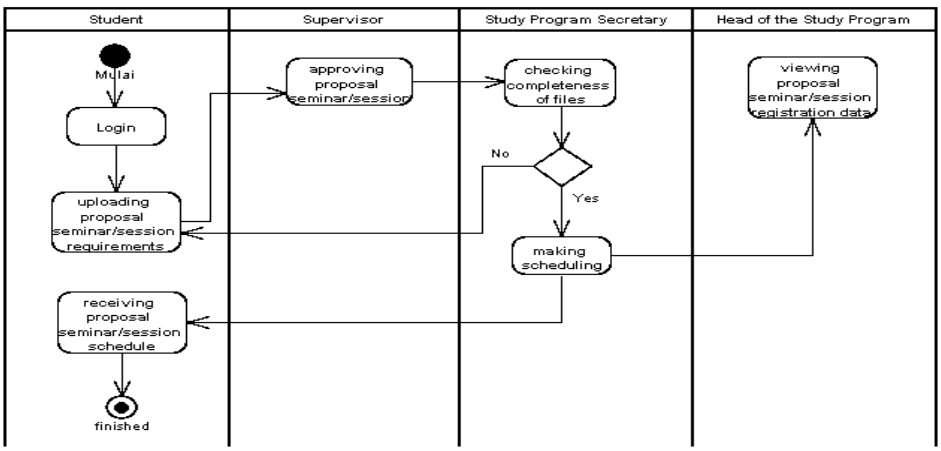

Figure 3. Current System Analysis

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 4, **Number** 2, November 2024
https://doi.org/10.47709/brilliance.v4i2.5011

Based on Figure 3 for the current system analysis, the process begins when students log in and upload the required files for the proposal seminar or trial. Furthermore, the supervisor has a role in approving the proposal or trial submitted by the student. The next step involves the role of the Study Program Secretary, who is tasked with checking the completeness of the uploaded files. If any deficiencies exist, the files are returned to the student for correction. However, if the files are complete, the process continues to the seminar or trial scheduling stage. At this stage, the Head of the Study Program can view the seminar or trial registration data entered into the system. After completing all stages, students will receive a schedule for the seminar or trial, and the process is considered complete. Each stage plays a crucial role in organizing and tracking progress from preparation to implementation of the student proposal seminar or trial.

b. Proposed System Analysis

The following is an analysis of the proposed system in the information system developed by the author, as follows:

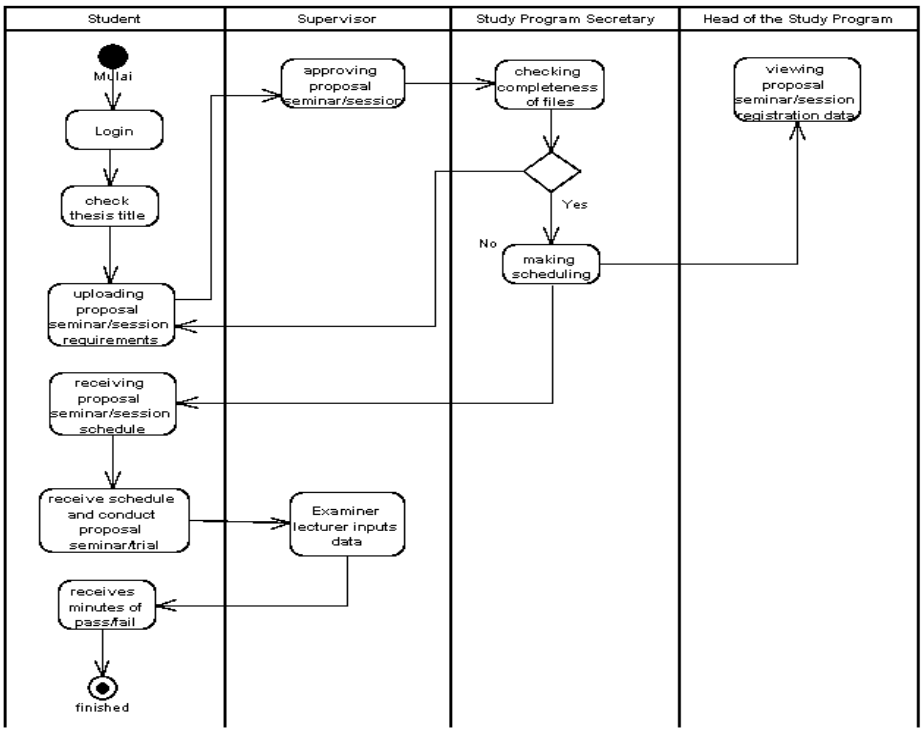

Figure 4. Proposed System Analysis

This proposed information system begins when a student logs in and checks the title of the thesis that has been submitted. Next, the student uploads the required files for the proposal seminar or trial. The following process involves the supervisor, who must approve the proposal or trial submitted by the student. After that, the study program secretary checks the completeness of the files. If the files are incomplete, the process will return to the stage of uploading the required files by the student. However, the study program secretary will continue scheduling the proposal seminar or trial if the files are complete. At this stage, the Head of the Study Program can see the registration data for the proposed seminar or trial received. Then, students will receive a schedule and conduct a proposal seminar or trial. After that, the examiner will give a score for the seminar or trial that the student has conducted. Finally, students will receive a report regarding their graduation status, and the process is considered complete. In this system, each stage is essential in regulating the process from preparation to final assessment related to the proposal seminar or student trial. This system is designed to facilitate the management and tracking of the stages required in this academic process in a more structured and organized manner.

System Design

a. Use case Diagram

The use case diagram in the information system of the system developed by the author is as follows:

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 4, **Number** 2, November 2024
https://doi.org/10.47709/brilliance.v4i2.5011

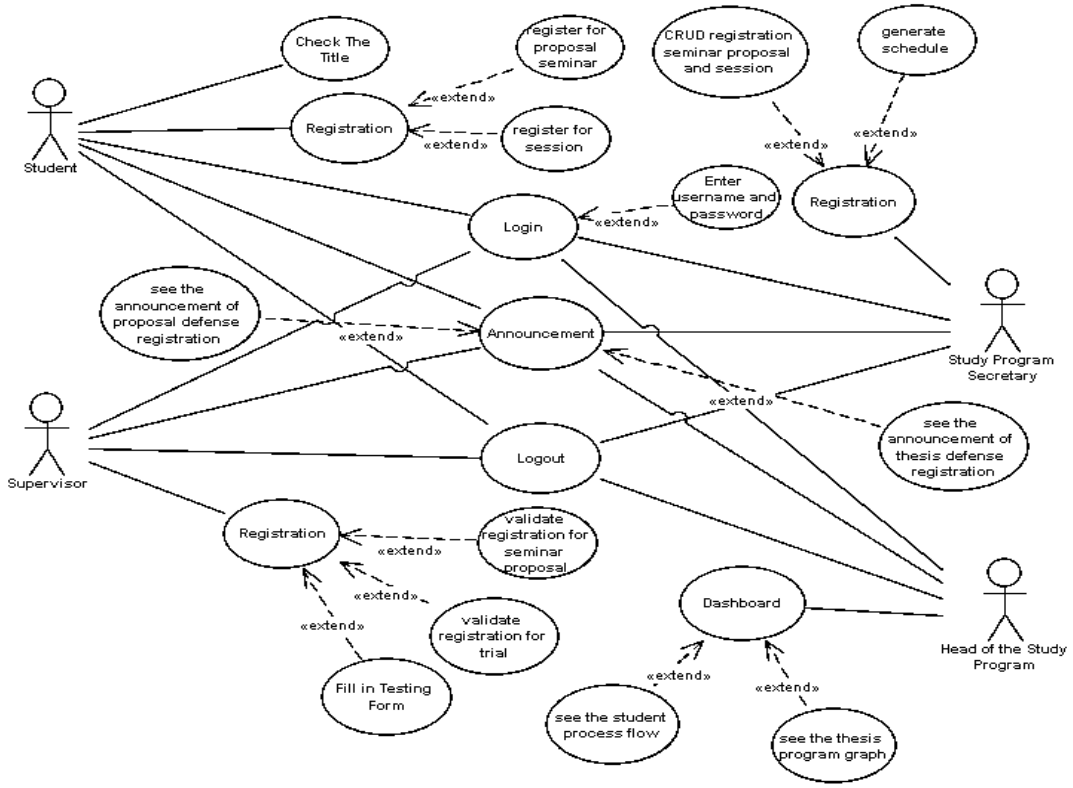

Figure 5. Use case Diagram

The use case diagram in Figure 5 shows several actors and actions taken by each actor in the system. There are three main actors: students, Supervisor/Examiner, Secretary of Study Program, and Head of Study Program. Students can perform a series of essential actions.

Students can check the thesis title, which is the first step in registering for a proposal seminar or thesis defense. Furthermore, students can register for a proposal seminar and register for the defense. This action illustrates the active involvement of students in their academic process, allowing them to manage and organize essential stages in their thesis journey. The Supervisor/Examiner also has a vital role in this system. They can validate the students' registration of proposal seminars and thesis defenses. In addition, they are responsible for filling out the test form related to the proposal seminar or thesis defense of students.

This action shows the critical role of lecturers in ensuring the quality and feasibility of proposal seminars and student thesis defenses. On the administrative side, the Study Program Secretary is responsible for managing registration. They can update the registration of proposal seminars and trials by adding, deleting, or updating relevant data. This capability allows them to ensure the integrity and completeness of the information required in the administrative process. The Head of Study Program has a role in monitoring the entire academic process. They can see a dashboard that provides an overview of the process flow undergone by students and the overall thesis progress graph. This allows them to monitor student progress and understand the dynamics in the academic environment. The combination of these functions will enable actors to manage essential stages in the journey of students' theses. With the roles and functionalities defined in this use case diagram, the system can provide comprehensive and organized support in managing student theses..

b. Class Diagram

A class diagram is a diagram that shows the relationship between classes and primary elements in the system. The class diagram for the thesis information system can be seen in the image below.

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 4, **Number** 2, November 2024
https://doi.org/10.47709/brilliance.v4i2.5011

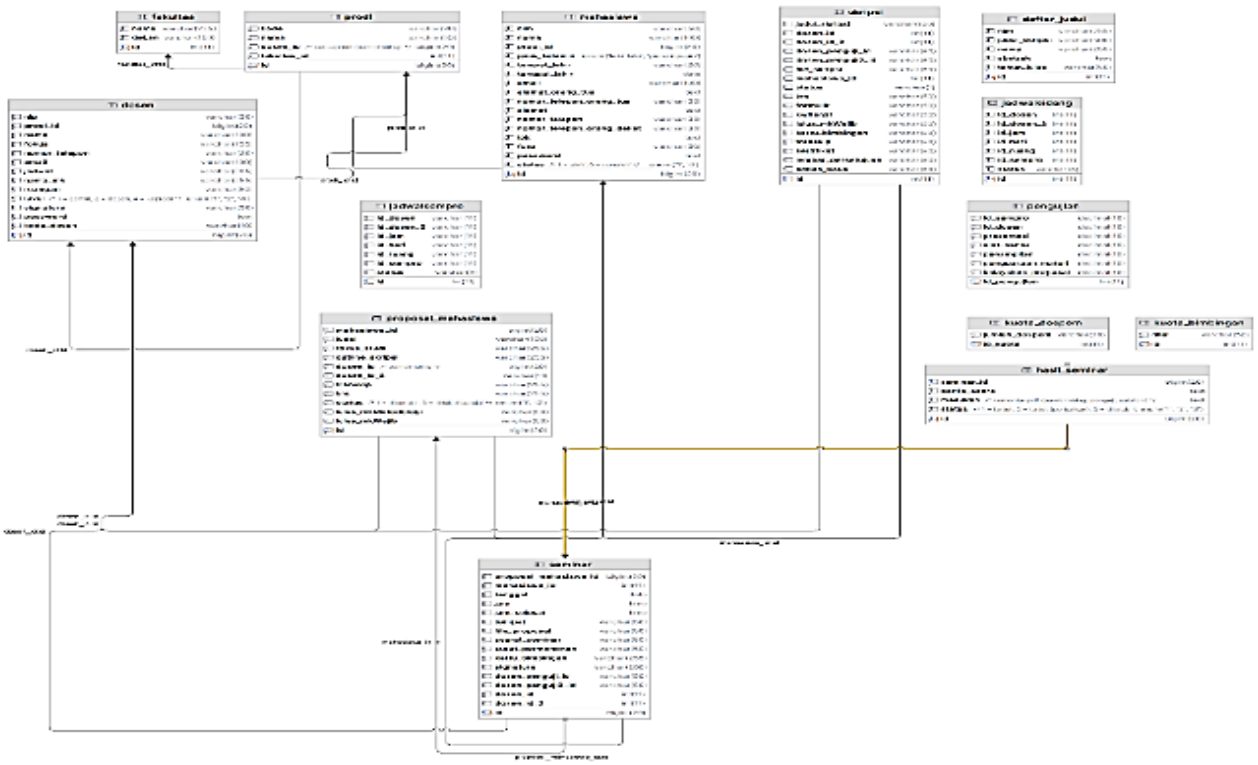

Figure 6. Class Diagram

System Implementation

The results of the implementation of the supervisor and guidance quota settings in the thesis information system are as follows:

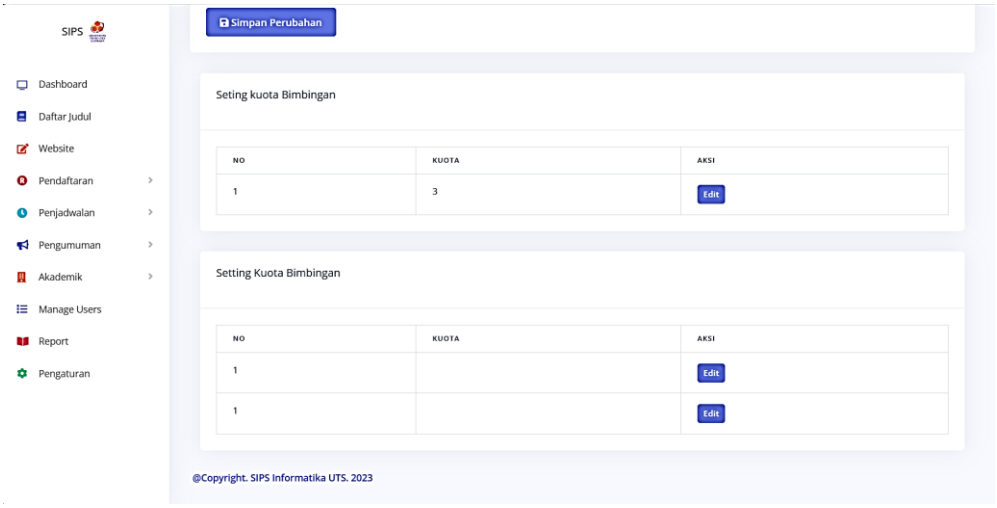

Figure 7. Supervisor and Guidance Quota Settings

Figure 7 displays the settings page for setting the supervisor and guidance quota. This page includes several fields, such as quota and action fields. The study program secretary can click the edit button in the action section to edit the supervisor quota. This will create a pop-up window to change the supervisor and guidance quotas. After that, the study program secretary can save the changes by pressing the save button. This setting enables the study program secretary to easily manage and change guidance-related quota-city. This process makes it easy to adjust the supervisor quota and guidance quota according to the needs of the study program. With this feature, quota editing can be done quickly and efficiently, ensuring the smooth running of the administrative process related to academic guidance.

The results of the implementation of Uploading lecturer signatures in the thesis information system are as follows:

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 4, **Number** 2, November 2024
https://doi.org/10.47709/brilliance.v4i2.5011

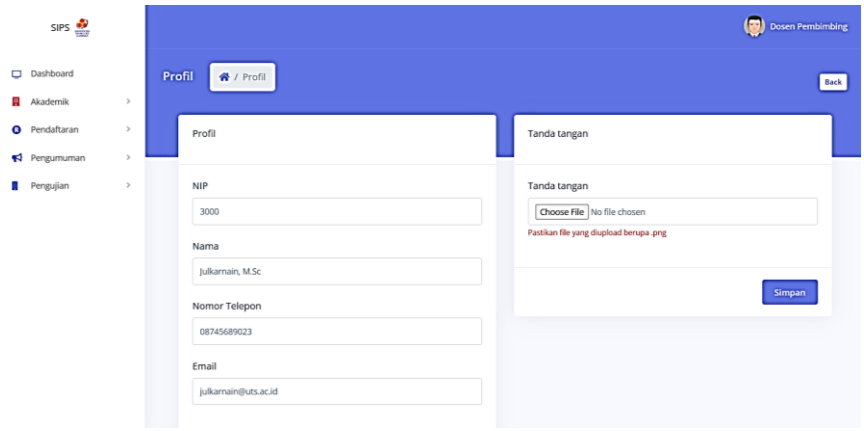

Figure 8. Uploading Lecturer Signatures

The image shown is a settings page that allows adjustments to quotas for supervisors and guidance. This page contains several columns, including quota and action columns. The study program secretary can click the edit button in the action section to change the supervisor quota. This step will create a modal window that allows editing of the supervisor and guidance quota. After the settings are complete, the secretary can save the changes by pressing the save button. This process ensures that the quota that has been set is successfully saved. This feature facilitates quota management by providing access to the study program secretary to make adjustments without other parties' intervention. Changing the supervisor quota directly through this simple action helps manage and allocate resources more efficiently in the academic environment.

The results of the implementation of Inputting the examiner's value in the thesis information system are as follows:

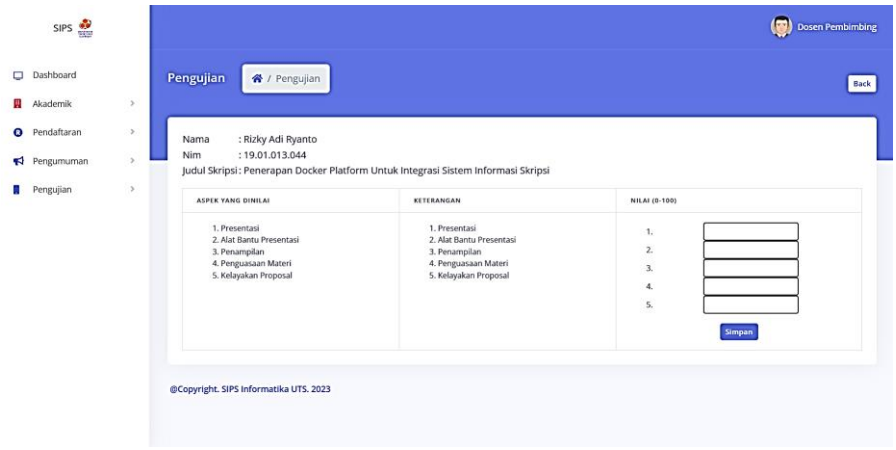

Figure 9. Inputting the Examiner's Value

The image shows a page dedicated to the role of lecturers in uploading signatures, both as supervisors and examiners. On this page, lecturers are asked to upload their signatures in PDF or image format by clicking the "Select File" button. Once the signature upload is complete, lecturers can save it by clicking the "Save" button. This feature facilitates the efficient digital signature storage process for lecturers, allowing them to provide their signatures as needed without excessive technical difficulties. The ability to upload in different formats will enable lecturers flexibility, so they can use the format that best suits their type of signature. This process helps to archive essential data while speeding up various administrative processes requiring lecturer signatures.

DISCUSSION

Docker Platform Integration

Implementasi Docker Platform yang direncanakan untuk mengintegrasikan Sistem Informasi Skripsi serta Sistem Informasi Pengecekan Judul Skripsi yang berbasis web di Program Studi Informatika. Dimana Rancangan dapat di lihat pada Gambar 13 dibawah ini.

The implementation of the Docker Platform is planned to integrate the Thesis Information System and the Thesis Title Checking Information System based on the web in the Informatics Study Program, the design of which can be seen in Figure 10 below.

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 4, **Number** 2, November 2024
https://doi.org/10.47709/brilliance.v4i2.5011

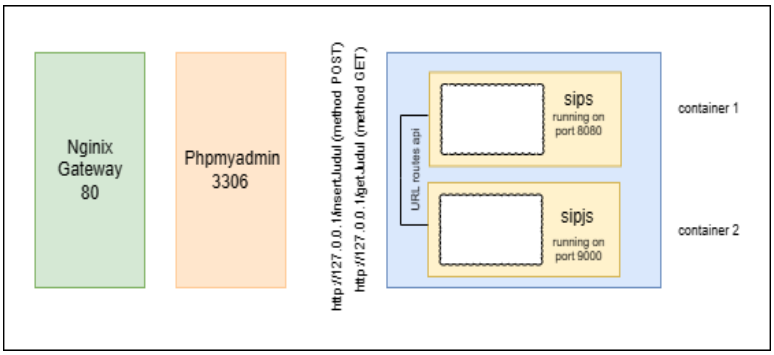

Figure 10. Docker Platform Integration Architecture

In this study, Nginix Gateway 80 acts as a web server operated by Docker as a reverse proxy. Port 80 is chosen as the operational location of Nginix in the Docker design, where this server will handle all HTTP requests that come in through that port. Furthermore, Phpmyadmin will accommodate the database stored in the Docker container. The plan also involves two containers operating on different ports. The first container aims to run the SIPS project on port 8080, while the second container will execute the SIPJS project on port 9000. Both will connect via the API route URL with the endpoint http://127.0.0.1/getJudul using the GET method. In addition, the endpoint http://127.0.0.1/insertJudul functions as an API endpoint with the POST method. If an addition such as /angkat in the endpoint, for example, http://127.0.0.1/getJudul/1, the number 1 will be a required parameter. Not only that, in both containers, there are Dockerfile and docker-compose files, which are the Docker configurations in each container. This design is essential in research because it presents a structured way to integrate various system elements. The configuration presented in each container allows for smooth and efficient operation. This is crucial in ensuring that the system developed in this study can run according to project needs and provide the expected benefits in Web-based Thesis Information Systems. The docker-compose and Dockerfile configuration files can be seen in Figure 11 below.

File konfigurasi docker-compose

```
# docker-compose.yml
version: '3'

services:
  db:
    image: mysql:5.7
    environment:
      MYSQL_ROOT_PASSWORD:
      MYSQL_DATABASE: skripsi
      MYSQL_USER: user
      MYSQL_PASSWORD:
    volumes:
      - db_data:/var/lib/mysql

  phpmyadmin:
    depends_on:
      - db
    image: phpmyadmin/phpmyadmin
    restart: always
    ports:
      - 8081:80
    environment:
      PMA_HOST: db
      MYSQL_ROOT_PASSWORD:

volumes:
  db_data:
```

File konfigurasi Dockerfile

```
# Dockerfile
FROM php:7.4-apache

# Install extensions
RUN apt-get update && apt-get install -y \
    libicu-dev \
    libfreetype6-dev \
    libjpeg62-turbo-dev \
    libpng-dev \
    libxml2-dev \
    libonig-dev \
    default-mysql-client \
    && docker-php-ext-install -j$(nproc) intl gd mysqli xml mbstring
\   && docker-php-ext-configure gd --with-freetype --with-jpeg \
    && docker-php-ext-install -j$(nproc) gd

# Enable mod_rewrite for images
RUN a2enmod rewrite

# Copy application source
COPY . /var/www/html/
```

Figure 11. Docker-Compose Configuration File and Dockerfile

Previously integrated using the docker platform, the thesis title checking system and the thesis information system required two separate steps: students accessed the thesis title checking system website to check the title and used the thesis information system for registration. Two accesses are necessary because the title-checking and registration functions are separate. Students must take the first step to check the title and then the second step to register the thesis. This ensures the separation of tasks so that the registration and title-checking processes run independently according to system needs. The appearance of the Title Checking Information System before integration can be seen in Figure 12:

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 4, **Number** 2, November 2024
https://doi.org/10.47709/brilliance.v4i2.5011

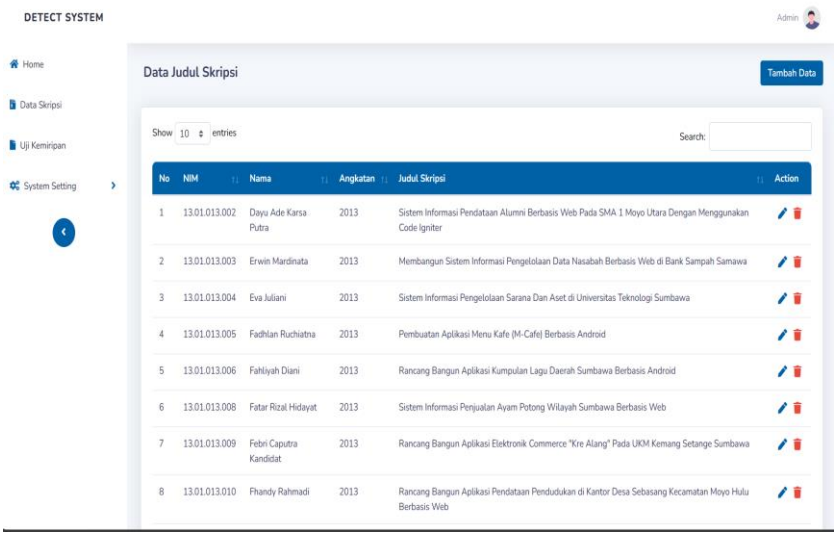

Figure 12. Title Checking Information System Before Integration

After being integrated through the Docker Platform, this system combines the title-checking and thesis registration processes into one step. Students now only need to access one integrated platform for both methods. Docker allows the consolidation of previously separate functionality into one, making it easier for students to access and use. This integration makes the process more efficient and effective. It allows users to complete the title-checking and thesis registration process with more straightforward and centralized steps in one platform. The appearance of the Thesis Information System that has been integrated can be seen in Figure 15 below.

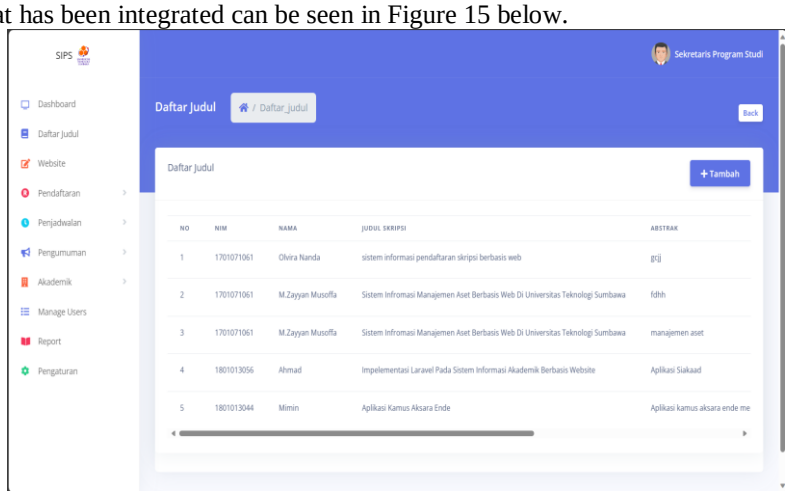

Figure 13. Title Checking Information System After Integration

Black box testing method

Black box testing is a software testing method that evaluates functionality without considering the internal structure. This test examines the input and output of the system without knowing its internal code or logic. By simulating user situations, the goal is to validate the system's functions according to specifications. This test evaluates external functionality, detects bugs, and ensures user needs match application behavior. Black box testing focuses on the interaction between the system and the user and ensures that the system functions appropriately without considering its internal parts. The results of testing with Black box testing can be seen in Table 1 below.

Table 1. Black Box Testing Results

Expected	Action	Conclusion
Opening thesis title list page	Displaying integration result page	Successful
Uploading lecturer signature	Saving signature data	Successful
Inputting Test Score	Saving test score data	Successful
Setting supervisor quota and guidance quota	Saving and limiting supervisor quota and guidance quota	Successful

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 4, **Number** 2, November 2024
https://doi.org/10.47709/brilliance.v4i2.5011

CONCLUSION

Based on the results of the implementation and testing carried out in this study, integrating the Thesis Information System and the Thesis Title Checking System using the Docker platform significantly impacts the efficiency of administrative processes in the academic environment. After being integrated through the Docker Platform, this system unites checking the title and registering the thesis into one step. Students now only need to access one integrated platform for both processes. Docker allows the consolidation of previously separate functionalities into one, facilitating student access and use. With this integration, the process becomes more efficient and effective because it will enable users to complete the process of checking the title and registering the thesis with more straightforward and more centralized steps in one platform.

This integration has succeeded in consolidating the separate functions of checking the title and registering the thesis into one integrated platform, facilitating student access and use. The implementation steps carried out are by the Software Development Life Cycle (SDLC) and involve various methods of data collection and software development. Using Docker as a containerization platform, this study provides valuable insights into the ease of integration, resource management, and efficiency in deploying academic applications. The suggestions are that further development related to system maintenance is needed, including fixing detected bugs and errors and adjusting according to user feedback. It is also necessary to improve system security to protect the sensitive data of students and other users.

REFERENCES

A. Ikegami, I.D.M.B.A. Darmawan, dan I. G. A. G. A. K. (2022). Implementasi Rest Api Pada Fitur User Dari Aplikasi Peminjaman Ruangan. *Jurnal Pengabdian Informatika*, *1*(1), 73–82. https://doi.org/10.24843/jupita.2022.v01.i01.p12

Apriansyah, S. R., & Julkarnain, M. (2023). Penerapan Algoritma Untuk Mendeteksi Kemiripan Judul Skripsi Pada Program Studi Informatika. In USC (Ed.), *UTS STUDENT CONFERENCE* (Vol. 1, Issue 2, pp. 139–150). USB. https://conference.uts.ac.id/index.php/Student/article/download/711/330/3344

Bik, F. R., & Asmunin. (2017). Implementasi Docker Untuk Pengelolaan Banyak Aplikasi Web (Studi Kasus : Jurusan Teknik Informatika UNESA). *Jurnal Manajemen Informatika*, *7*(2), 46–50. https://ejournal.unesa.ac.id/index.php/jurnal-manajemen-informatika/article/view/20789/19062

Danur Wijayanto, Arizona Firdonsyah, & Faisal Dharma Adhinata. (2021). Implementasi Continous Integration/Continous Delivery Menggunakan Process Manager 2 (Studi Kasus: SIAKAD Akademi Keperawatan Bina Insan). *Teknika*, *10*(3), 181–188. https://doi.org/10.34148/teknika.v10i3.400

Fatihah, A. (2019). Pengembangan Aplikasi Cloud Storage Dengan Pemanfaatan Storage Pada Vps Dan Androidinterface. *Journal of Informatics and Computer Science Vol.*, *5*(1), 25–30. http://jurnal.uui.ac.id/index.php/jics/article/view/454

Fihri, M., Negara, R. M., & Sanjoyo, D. D. (2019). Implementasi & Analisis Performansi Layanan Web Pada Platform Berbasis Docker. *E-Proceeding of Engineering*, *6*(2), 3996. https://jitl.web.id/index.php/engineering/article/view/10367/10222

Isron, M. H., & Putra, R. E. (2021). Implementasi Virtual Server Berbasis Container Pada Sistem Informasi Geografis Cagar Budaya Mojokerto. *(Journal of Informatics and Computer Science)*, *03*(November), 143–154. https://ejournal.unesa.ac.id/index.php/jinacs/article/view/43178/37037

Mochammad Reza Ramadhan , Lukito Edi Nugroho, S. S. (2017). Perancangan Sistem Informasi Monitoring Skripsi. *Jurnal Ilmiah Core It*, *x*, 290–295. http://www.ijcoreit.org/index.php/coreit/article/view/121

Mubarok, H., & Saptono, H. (2021). Pengujian Dan Implementasi Efektivitas Sistem Virtualisasi Berbasis Docker Platform. *Jurnal Informatika Terpadu*, *7*(1), 33–38. https://journal.nurulfikri.ac.id/index.php/JIT

Parama, R. A. (2022). *Implementasi Docker Untuk Pengelolaan Aplikasi SIAKAD PT Sentra Vidya Utama (SEVIMA)* (Issue 16). https://repository.its.ac.id/94922/1/05111840000123-Project_Report.pdf

Pressman, R. S. (2015). Software Quality Engineering: A Practitioner's Approach Fifth Edition. In *Software Engineering: A Practitioner's Approach*. McGraw-Hill Higher Education. http://gen.lib.rus.ec/book/index.php?md5=E1368B6CA046D3F456124359804C640F

Rahman, B., & Ningsi, S. A. (2022). Sistem Informasi Monitoring Skripsi Mahasiswa Berbasis Website. *Simtek : Jurnal Sistem Informasi Dan Teknik Komputer*, *7*(2), 166–170. https://doi.org/10.51876/simtek.v7i2.152

Setiyani, L. (2019). Pengujian Sistem Informasi Inventory Pada Perusahaan Distributor Farmasi Menggunakan Metode Black Box Testing. *Techno Xplore : Jurnal Ilmu Komputer Dan Teknologi Informasi*, *4*(1), 1–9. https://doi.org/10.36805/technoxplore.v4i1.539

Wijaya, A. F., & Eliezer, A. (2019). Sistem Informasi Penggajian Karyawan Ksp Talenta. *JASIEK (Jurnal Aplikasi Sains, Informasi, Elektronika Dan Komputer)*, *1*(2), 89–97. https://doi.org/10.26905/jasiek.v1i2.3153

Wijaya, R., Prasetyo, A., & Wahyuningsih, D. (2021). Implementasi Ci/Cd Untuk Buiddan Deploy Website Dengan Docker Runner Pada Organisasi Belajar Linux Id. In S. J. STI&K (Ed.), *Seminar Nasional Teknologi Informasi dan Komunikasi (SeNTIK)* (Vol. 5, Issue 1, pp. 173–178). SeNTIK. https://ejournal.jak-

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 4, **Number** 2, November 2024
https://doi.org/10.47709/brilliance.v4i2.5011

stik.ac.id/files/journals/2/articles/sentik2021/2845/submission/proof/2845-13-1873-1-10-20210901.pdf

Wiki Nasmansyah, M. J. (2023). Algoritma Genetika adalah teknik optimasi yang terinspirasi dari proses evolusi alami, menggunakan konsep seleksi,. In USC (Ed.), *UTS STUDENT CONFERENCE* (Vol. 1, Issue 2). USC. https://conference.uts.ac.id/index.php/Student/article/view/697/328

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.