

The Development Of Web-Based Application For Reporting Status Component At Aircraft Maintenance, Repair, And Overhaul (MRO) Industry

Rayhan Dhiya Rista¹, Jansen Wiratama^{2✉}

^{1,2}Information Systems, Faculty of Engineering and Informatics, Universitas Multimedia Nusantara, Indonesia

Informasi Artikel

Riwayat Artikel

Diserahkan : 29-05-2023

Direvisi : 09-06-2023

Diterima : 16-06-2023

Kata Kunci:

Aplikasi, Laporan, Perawatan, Pesawat terbang, Website.

Keywords:

Aircraft, Application, Maintenance, Reporting, Web-based.

ABSTRAK

PT GMF Aero Asia, Tbk, memiliki aplikasi yang mengintegrasikan semua kebutuhan dalam hal pemeliharaan dan pengelolaan, khususnya status komponen, dengan menggunakan tools SAP SWIFT. Namun, SAP SWIFT memiliki beberapa kelemahan, yaitu findings berulang karena keterbatasan perbaikan data komponen di SAP SWIFT, selain itu persiapan untuk setiap status dilakukan secara manual, dan ketidakmampuan untuk memenuhi kebutuhan pelanggan secara cepat atas kebutuhan audit yang meningkat. Oleh karena itu terdapat urgensi untuk mengatasi permasalahan tersebut agar proses integrasinya berjalan dengan optimal. Aplikasi berbasis website yang dapat menampilkan status secara otomatis, memiliki kemampuan dalam mengolah data, menampilkan notifikasi perbedaan data lama dan baru, serta dapat menerima data yang dikirim dari SAP SWIFT bisa menjadi solusi. Desain Front-End aplikasi ini menggunakan framework vue.js, dan desain Back-End menggunakan Nest.js yang terintegrasi dengan PostgreSQL sebagai DBMS. Pada tahap akhir, sebelum pengguna menggunakan aplikasi ini, dilakukan pengujian REST API menggunakan POSTMAN dengan hasil pengujian yang sangat baik sesuai skenario pengguna.

ABSTRACT

PT GMF Aero Asia, Tbk, has an application that integrates all needs in terms of maintenance and management, especially component status, using SAP SWIFT tools. However, SAP SWIFT has several areas for improvement, namely repeated findings due to limitations in repairing component data in SAP SWIFT; besides, preparation for each status is done manually, and the inability to meet customer needs quickly for increased audit needs. Therefore there is an urgency to overcome these problems to optimize the integration process. A website-based application that can display status automatically, process data, display notifications of differences between old and new data, and receive data sent from SAP SWIFT can be a solution. The front-end design of this application uses the vue.js framework, and the back-end design uses Nest.js, which is integrated with PostgreSQL as a DBMS. In the final stage, before users use this application, REST API testing is done using POSTMAN with very good test results according to user scenarios.

Corresponding Author:

Jansen Wiratama

Information Systems, Faculty of Engineering and Informatics, Universitas Multimedia Nusantara.

Scientia Garden Jl. Boulevard Gading Serpong, Kab. Tangerang-Banten, 15810.

Email: jansen.wiratama@umn.ac.id

INTRODUCTION

PT Garuda Maintenance Facility Aero Asia Tbk is a company engaged in aircraft maintenance services, component maintenance and calibration, engine maintenance for aircraft and industry, manufacture and maintenance of supporting advice, engineering services, and many more, especially in the field of Maintenance, Repair and Overhauling (MRO). MRO is a transportation business aiming to provide services to aviation service business actors to realize applicable flight rules (Samosir, 2020). PT Garuda Maintenance Facility (GMF) Aero Asia Tbk has an inventory or warehouse for storing thousands of components used for management or maintenance, such as expandable components with a high frequency of requests, so that the required components can be prepared effectively and efficiently (Yuwono & Saptadi, 2021). The hangar at PT GMF Aero Asia, Tbk, is one of the largest hangars in the world, which is devoted to managing and maintaining narrow body type aircraft. Hangar 4 can carry out aircraft maintenance from outside, such as KLM Airlines aircraft owned by the Netherlands. This company has an application that integrates all needs in terms of maintenance and management, especially in component status, using SAP SWIFT tools. However, SAP SWIFT has several areas for improvement, namely repeated findings from auditors and lessors due to limitations on repairing component data in SAP SWIFT, preparation for each status is still done manually, and the inability to quickly address customer needs for increased audit needs. Another limitation of the SWIFT system is that it cannot backdate incorrectly installed components. As a result of the irreversible SWIFT data, there were repeated findings by the auditors when engineers downloaded new component data from SWIFT.

After conducting several literature studies relevant to this research, it is evident that application implementation can solve problems related to business processes at PT GMF Aero Asia, Tbk in its business processes. In the first literature review, the object of this research is to apply SWIFT, a renewal of the SAP Enterprise system that can collect data on each job and even bill customers (Suryatman et al., 2019). In addition, other studies emphasize the relevance of the model to the System Development Life Cycle (SDLC) application design method, which is suitable for use in a short time and relatively good results, such as the Rapid Application Development (RAD) model, Prototyping, and the Extreme Programming (XP) Model (Chandra et al., 2023). Further literature review reveals that RAD, Prototyping, and EXP models are for achieving timely system designs with limited resources (Wiratama & Ikana Desanti, 2022). In this research, the implemented system design model is the Extreme Programming (XP) model. This model was chosen in system design because this model is superior in speed and does not require detailed user design such as RAD and prototyping (Wiratama et al., 2023). The application's design in this study used the Vue.JS front-end framework and the Nest.JS back-end integrated with the PostgreSQL Database Management System (DBMS) and tools for conducting API testing using POSTMAN. The result of designing this application is to provide reliable, accurate, and consistent TC & NON-TC component data for customers with an efficient process.

RESEARCH METHODS

Extreme Programming (XP) is an agile software development methodology with several phases. It starts with a spike phase, where estimates are made through experiments and research. Then, user stories are created to represent functional requirements, followed by refining, and prioritizing them during the requirements phase. Test scenarios are formulated based on user stories and serve as a guide for testing. Release planning involves selecting and scheduling features for each release based on estimates. The customer conducts acceptance tests to ensure the implemented features meet requirements. Finally, the software is packaged and delivered in

iterative releases. XP emphasizes collaboration, frequent communication, and adaptability to deliver high-quality software.

The Agile Methodology with Extreme Programming Model was utilized by the researchers in this study to solve the difficulties studied. XP is a model that attempts to simplify certain stages of the development process to make this method more adaptable and flexible (Gunadi, 2021).

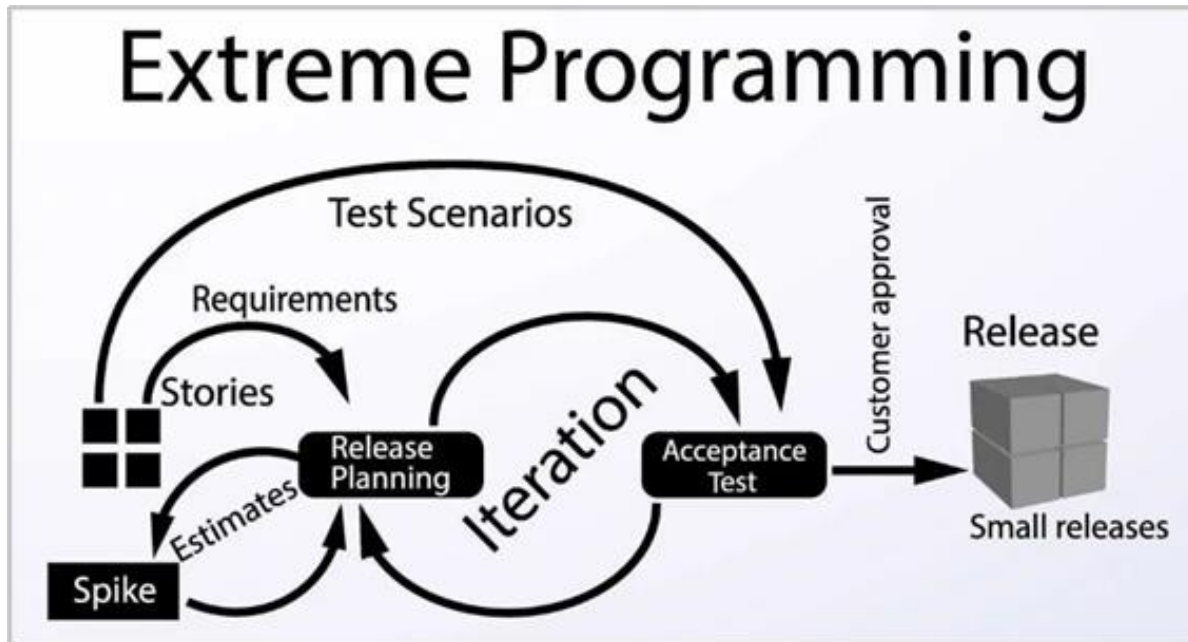


Figure 1. Extreme Programming Model

The stages of the Extreme Programming model-based system development model are depicted in Figure 1. The first step was to collect business process flow from the users. The Head of the Aircraft Data Group at PT GMF Aero Asia, Tbk, was an interview subject during the user stories phase. After user stories are used to gather system requirements, a needs analysis is conducted. The accompanying Table 1 Necessity investigation:

Table 1. Requirement Analysis

No	Problem	Needs
1	Repeated findings due to limitations on component data repair in SAP SWIFT.	Create applications that can reduce repeated findings to simplify business processes.
2	Preparation for each status done manually, and inability to quickly address customer needs for increased audit needs.	Make the process automated so that it can accelerate the needs of users.
3	Incapabilities in data processing, displays notifications of differences between previous and new data.	It displays its status automatically and provides notifications of changes to old vs. new data.

After obtaining the requirements analysis, this study enters the spike stage, which is the stage for estimating or estimating the application model or possible outcomes in the design process. There are two sorts of estimations that might occur: uncertain estimates and confident estimates. For uncertain estimations, the predictions are still skewed with an undetermined scope, specifically forecasts of the scheduling application design completion time and the application functionality based on user wants. The forecasts must then be focused and specific for confidence estimations so that they may move to the Planning step.

The Planning step comes after receiving the Requirements Analysis and Estimation from the Spike stage. Planning is carried out in building a web-based application during the planning stage. The following table 2 is the work plan:

Table 2. Extreme Programming Work Plan

No	Requirements	Planning
1	Interviews the users	Observe and Analysis the interview results
2	Front-End web application development	Develop the database using Vue.Js
3	Back-End web application development	Develop the database using Nest.Js
4	Database Development	Develop the database using PostgreSQL DBMS
5	Application Development	Implement REST API on the Application
6	Application Testing	Testing the REST API using POSTMAN

The front-end design uses Hyper Text Markup Language (HTML) and Cascading Style Sheets (CSS) to manage web-based application pages (Saputra & Astuti, 2018). Vue.js is a JavaScript framework used to create applications, mostly for website development (front-end). Vue.js is responsible for showing the user information based on the controller's commands. The controller is a function that processes the framework to be shown based on the user's preferences. Vue.js often works as layers in the form of presentations that can be integrated into projects (Sha et al., 2020). The Vue.Js framework uses the Typescript programming language . The component system in Vue.js is reactive, and Vue.js knows how to communicate or exchange information in the form of data through asynchronous events. Vue.js usually works in the form of layers in the form of presentations that can be used to integrate into projects (Ramadhanu & Priandika, 2021). Nest.Js is a framework that facilitates website development, especially on the back end. Nest has many libraries, including Express/Fastify, JWT, Redis, Mongoose, and TypeORM. By using Nest, we can efficiently reduce the time it takes to build application projects that already have standard writing or library usage. Nest.Js uses a modular system of every feature developed with the Typescript programming language, which allows developers to build highly scalable and testable applications (Wijaya, 2023).

Pg Admin 4 is a management tool for the PostgreSQL database that supports Windows, Linux, and Mac OS operating systems. PgAdmin4 is written using the Python programming language and Javascript (jQuery). This research uses PgAdmin to connect and integrate website databases with PostgreSQL. One of the advantages of pgAdmin is that it can connect to other, broader databases (Ramadhanu & Priandika, 2021). POSTMAN is an application that functions as a REST Client for testing REST APIs. POSTMAN performs API testing by sending a request to the web server and receiving a response. The features contained in POSTMAN are as follows: Collection, Environment, Response, Mockup Server, Script Test, and Automated Test (Anugrah & Fakhruddin, 2020).

RESULTS AND DISCUSSION

Following the stages in the Extreme Programming Model that have been carried out, from the initial stage to the iteration stage, this research will be continued based on the user story, spikes, and iteration results. Following are some of the user interfaces resulting from the website-based application front-end design in 4 scenarios, namely login, dashboard, items search, non-time controlled database search document, and comparison of new data vs. old data:



Figure 2. Login page

Figure 2 above displays the Login page for the admin before the admin enters the main page. The registration feature must be added because the client asks for user and admin data that only enter based on the existing database. Only those who already have an account with the company are given access rights in this application.

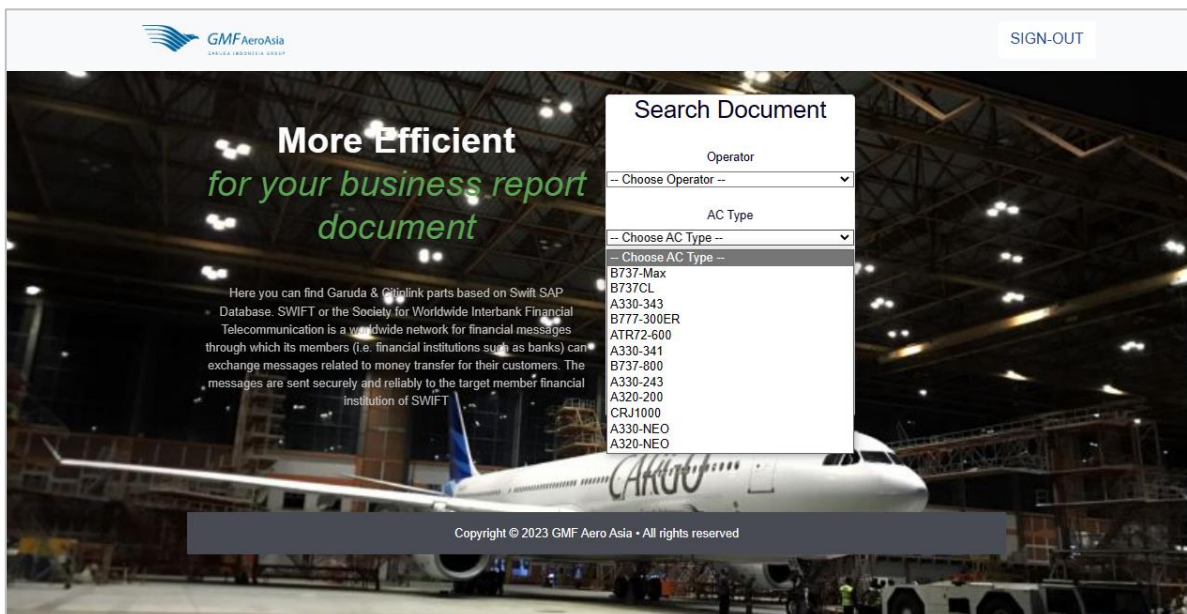


Figure 3. Home and Search page

Figure 3 is the page used by the admin to search for component status documents based on the operator, aircraft type, registration, and component status. Afterward, the admin can search by pressing the search button before entering the status component data page. Each feature on the front end is connected to the back-end database API. When using the search function, a search is performed based on the data of the desired operator, aircraft type, registration, and component status. However, the four data are connected from the Registration table as the Primary Key because they are used as attributes in calling data from SAP Swift.

Time Controlled / Non-Time Controlled

Non-Time Controlled Database

Non-Time Controlled

CONFIRM

PDF EXCEL EDIT COMPARE

☐	No	Functional Location	Part Number	Serial Number	Description	MP Number	Task	Install Date	TSN (HRS)	TSN (CYC)	Interval (HRS)	Interval (CYC)	Interval (DAYS)	TSLA (HRS)	TSLA (CYC)	TSLA(DAYS)	Life Rem
☐	1	PK-GIK.21.31.03.OV.AF	182820-3.70210	24997	HEAT EXCHANGER	2151010100	RST	2/18/2018	11697	7750		2200			2153		47
☐	2	PK-GNR.21.51.03.HE.AB.LH.FW	1152682-2.07217	24997	RESCU 406 AFN EMERGENCY LOCATOR TRANSMTR	2151010100	FCN	2/18/2018	11697	7750		2210		3456	2153	9078	47
☐	3	PK-GNR.23.26.01.LU	182820-	24997	LF-ULB	2151010100	RST	2/18/2018	11697	7750		3450		11234		678	54

Figure 4. Non-Time Controlled Database Search Document

A non-time-controlled database lacks built-in capabilities to handle time-dependent data or queries based on specific time ranges. While time-controlled databases have functionalities to manage temporal aspects like storing timestamps or valid time ranges, non-time-controlled databases focus on storing and retrieving non-temporal or static data. They are commonly used when temporal aspects are not crucial or when managing time-related information is unnecessary. The distinction between the two types of databases is about how temporal data is handled, but it doesn't prevent non-time-controlled databases from storing timestamps if needed. Figure 4 displays the data table when the user performs a search function based on documents on the main page (home). Users can only print out reports based on pdf or .xlsx formats, which are different from the functions that the admin can perform. The print-out report is done based on the selected lines.

Time Controlled / Non-Time Controlled

Time Controlled Database

RECENT DATA

What you looking for?

☐	No	Functional Location	Part Number	Serial Number	Description	MP Number	Task	Install Date	TSN (HRS)	TSN (CYC)	Interval (HRS)	Interval (CYC)	Interval (DAYS)	TSLA (HRS)	TSLA (CYC)	TSLA(DAYS)	Life Rem
☐	1	PK-GIK.21.31.03.OV.AF	182820-3.70210	24997	HEAT EXCHANGER	2151010100	RST	2/18/2018	11697	7750		2200					2153
☐	2	PK-GIK.21.31.03.OV.AW	182820-3.70210	24997	RESCU 406 AFN	2151010100	RST	2/18/2018	11697	7750		2200					2153
☐	3	PK-GIK.21.31.03.OV.FN	182820-3.70210	24997	LF-ULB	2151010100	RST	2/18/2018	11697	7750		2200					2153
☐	4	PK-GNR.23.71.00.VR	182820-3.70210	24997	UNDERWATER LOCATOR BEACON 90-DAYS	2151010100	RST	2/18/2018	11697	7750		2200					2153
☐	5	PK-	182820-	24997	HEAT	2151010100	RST	2/18/2018	11697	7750		2200					2153

DATA LAST UPDATE

No	Functional Location	Part Number	Serial Number	Description
MP Number	Task	Install Date	Time Since New (HRS)	Time Since New (CYC)

Figure 5. Table of Comparison of Old Data vs. New Data

Figure 5 displays table comparisons for admins to compare data from the SAP application with data updated on the website database. The data used for comparison is based on the table fields of MP Numbers. The NON-TC table does not have an MP Numbers field, so it is only compared to the Functional Location field.

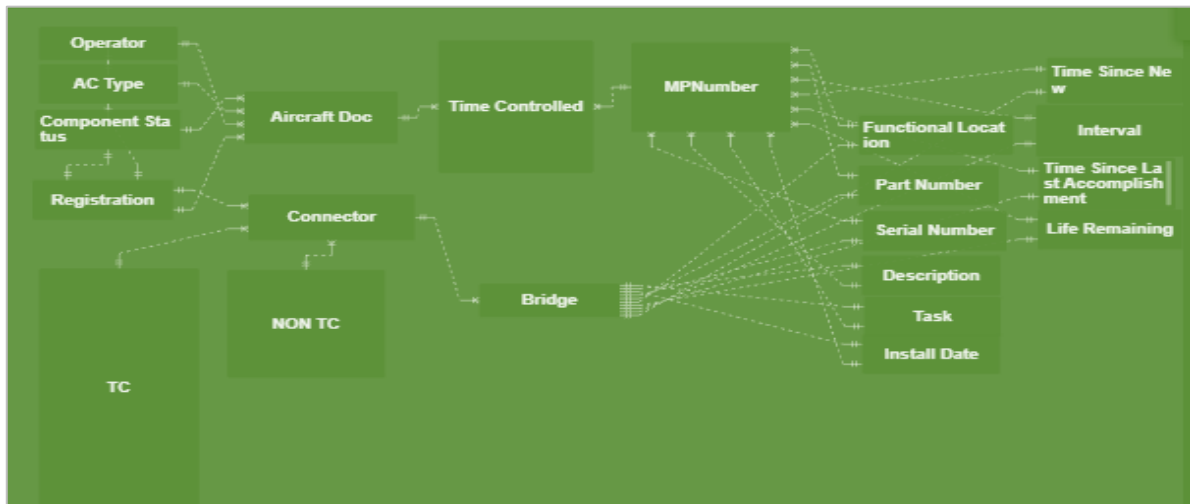


Figure 6. ERD from Database of Application

Figure 6 displays the ERD from the database used in this website application in Visual Studio Code. In the database above, 22 tables are used as back-end components so that they can be integrated into the front-end. Each table has a relationship to each table based on the Primary Key and Foreign Key. There is a user table for storing user data in the previous SAP database. Then, there are Operator, Aircraft Type, Registration, and Component Status tables which are part of the document that will be searched before entering the component status data page. By using the SAP database converted into a JSON file, it is necessary to create a table called a connector, a link between the TC and NON-TC tables, which are data from JSON and website data. Before doing a search that will enter the component status data table, the Registration table on the website has a Primary Key as an id which will be linked to the connector table as a Foreign Key. The TC table has a Primary Key warp (maintenance plan) and NON-TC Equnr (Equipment Number) connected to the cable connector as a Foreign Key. When all Foreign Keys based on registration fields are linked, the connector will generate an ID that will be used as a foreign key to display the component status table.

The last stage in this research is API Back-End application testing using POSTMAN tools. At this stage, REST API testing was carried out, as shown in the following figure:

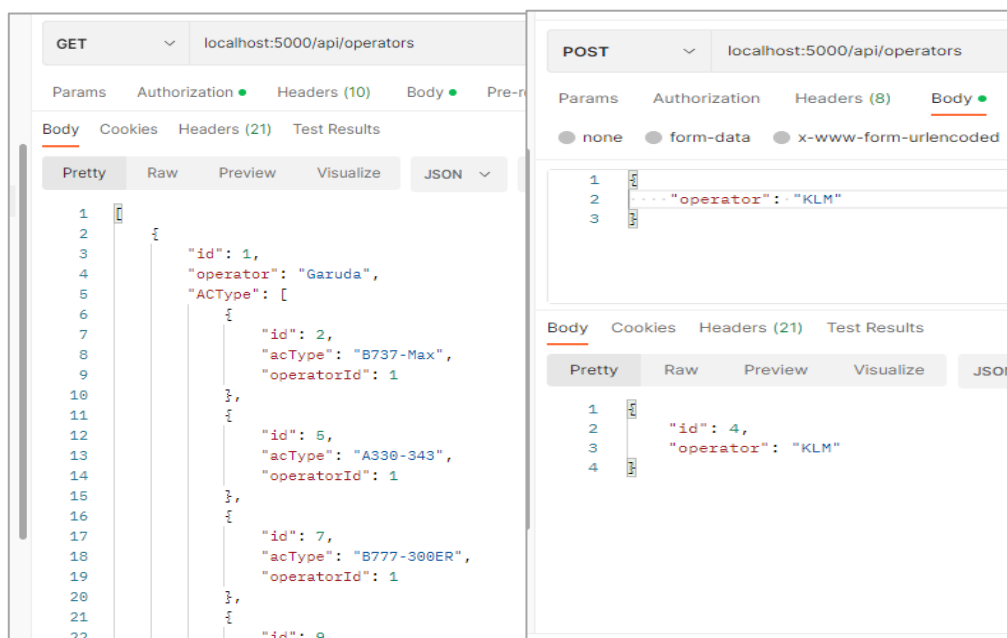


Figure 7. Testing API GET & POST Operator

The testing stage starts with operator testing. Figure 7 shown the GET function gets data from the operator table. The operator table has a relationship with the Aircraft Type table. Testing Operator with the POST function successfully adds data to the operator table and adds a new ID to enter into the table.

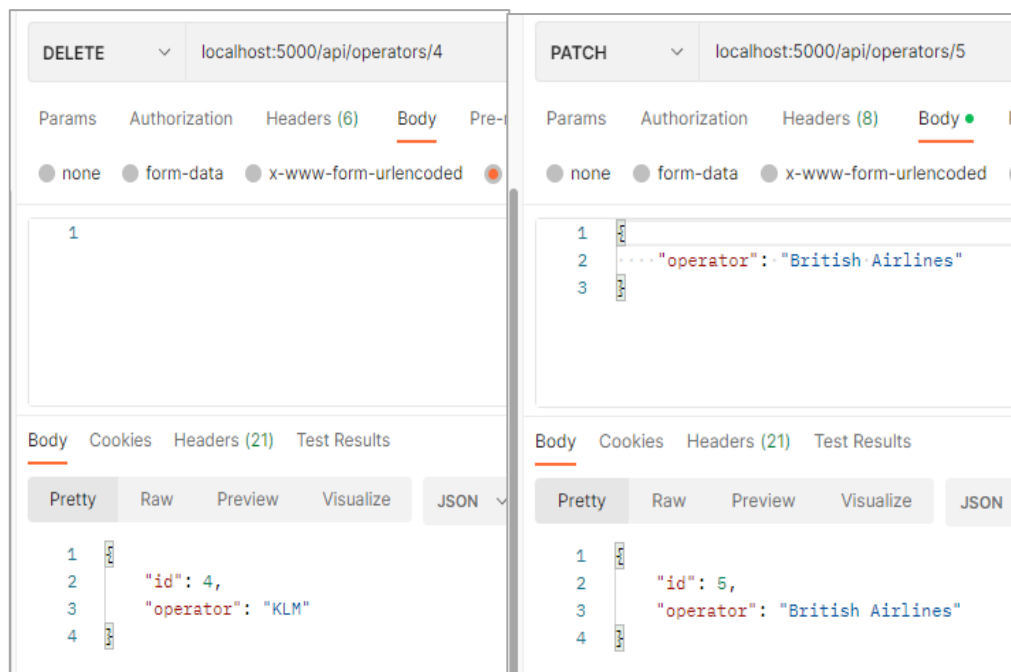


Figure 8. Testing API DELETE & PATCH Operator

Figure 8 shows that the operator testing with the DELETE function succeeded in deleting data previously added to the operator table based on the ID used. Testing the Operator with the PATCH function updated the data according to the operator field previously in the operator table. Each data deletion and update is performed based on the created operator ID.

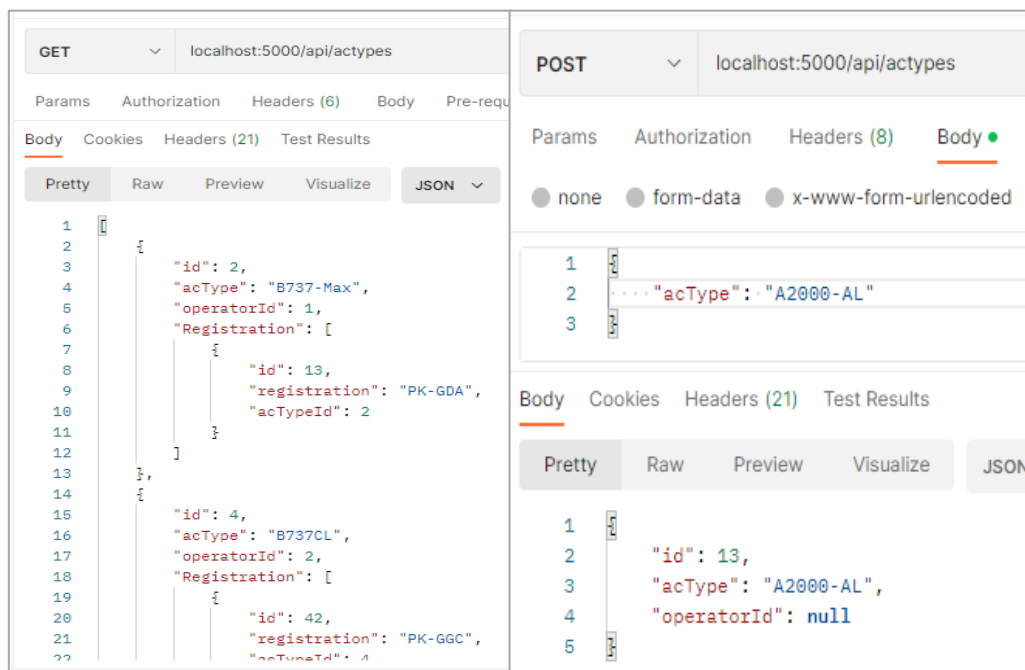


Figure 9. Testing API GET & POST Aircraft Types

Figure 9 shows that the Aircraft Type test with the GET function got data in the Aircraft Type table. The Aircraft Type table has a relationship with the Registration table. Aircraft type testing using the POST function succeeded in adding data to the Aircraft type table.

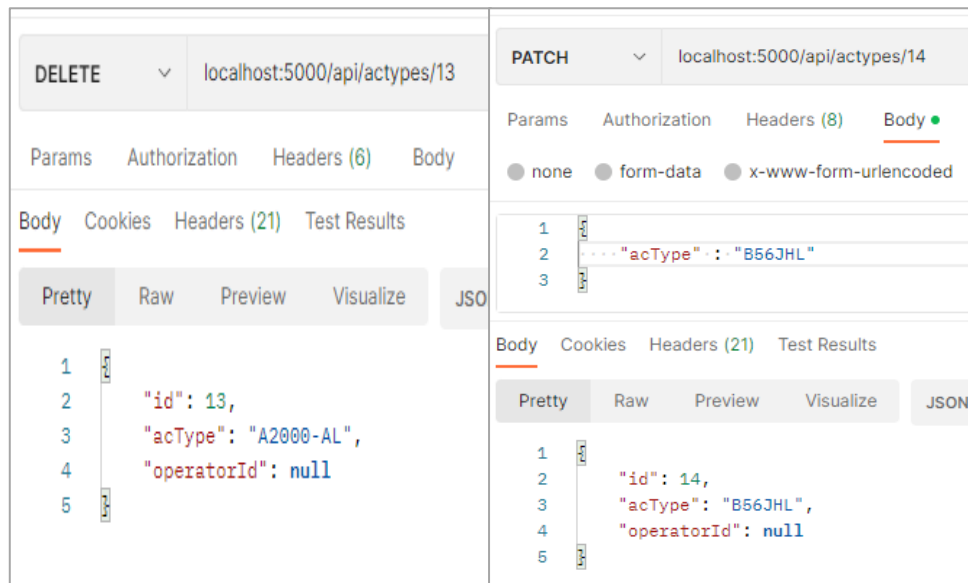


Figure 10. Testing API DELETE & PATCH Aircraft Types

Figure 10 tests Aircraft Type using the DELETE function to delete data in the Aircraft Type table based on ID. Testing Aircraft Type using the PATCH function succeeded in making changes to data in the Aircraft Type table based on ID. The Aircraft Type table also has a relationship with the Operator table.

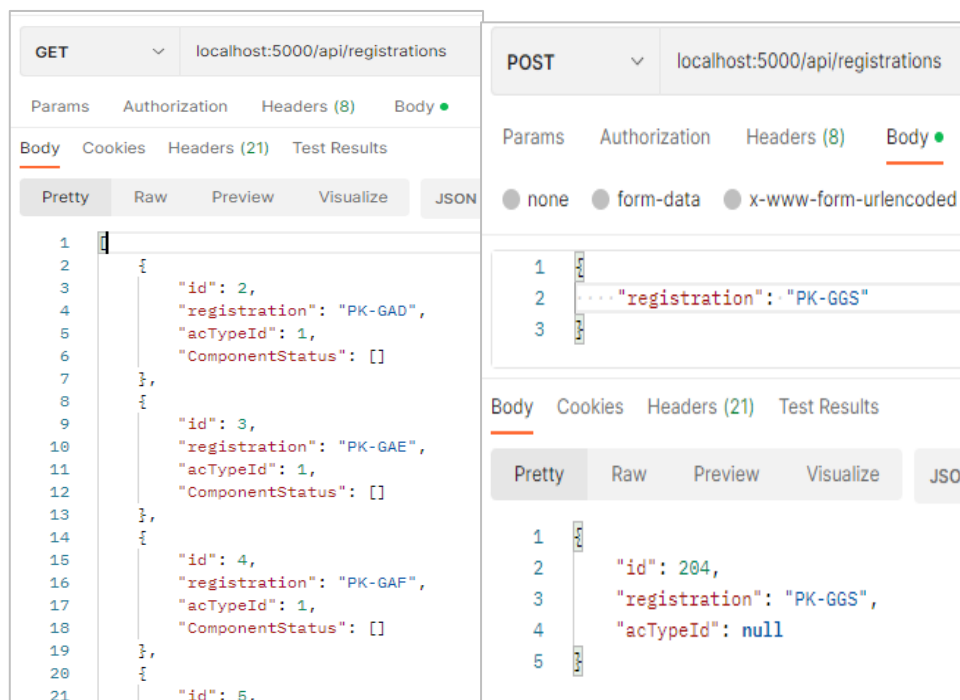


Figure 11. Testing API GET & POST Registrations

Figure 11 shows the test results of Registrations using the GET function to get data from the Registrations table. This table has a relationship with the Aircraft Type table with Component Status. Testing Registrations using the POST function successfully adds data to the Registrations table.

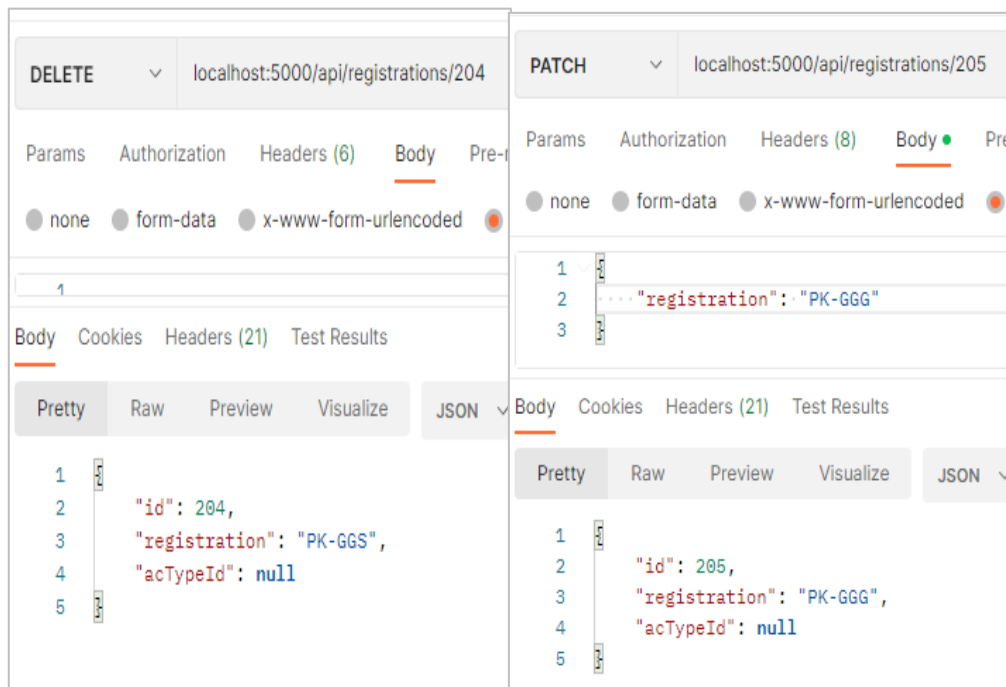


Figure 12. Testing API DELETE & PATCH Registrations

Figure 12 testing Registrations using the DELETE function. The DELETE function succeeded in deleting data in the Registrations table based on ID. Testing Registration using the PATCH function successfully changes the data in the Registrations table based on the ID. The Registrations table is also integrated into the Aircraft Type table.

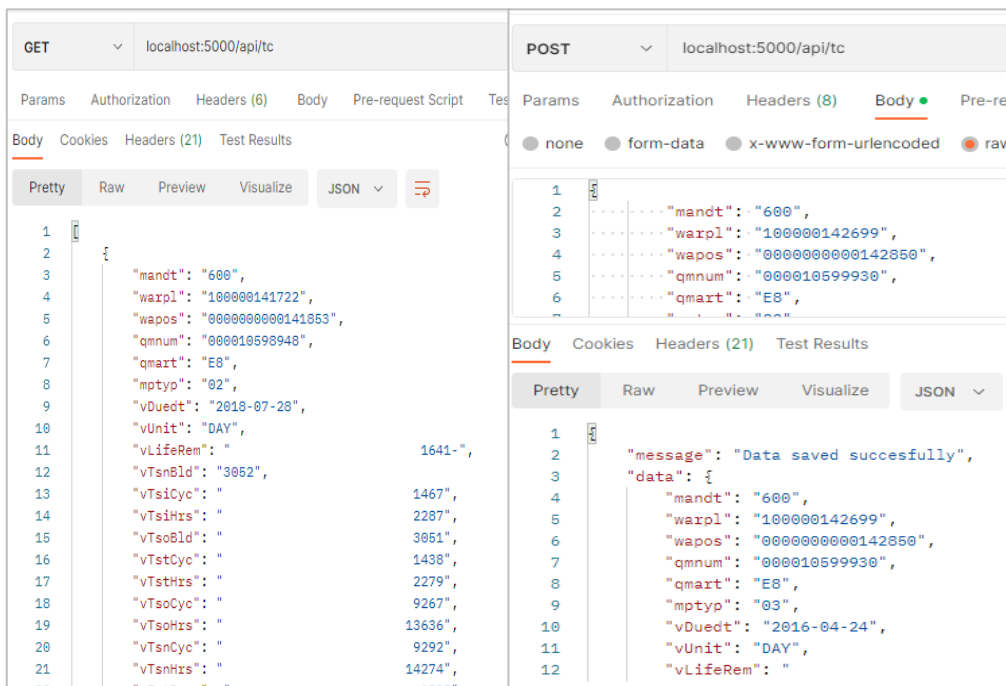


Figure 13. Testing API GET & POST Time-Controlled

Figure 13 shows Time-Controlled (TC) testing using the GET function to retrieve data from the SAP database JSON file successfully. This table has a relationship with the connector table. Testing TC using the POST function has successfully stored data based on data retrieved from the SAP JSON file.

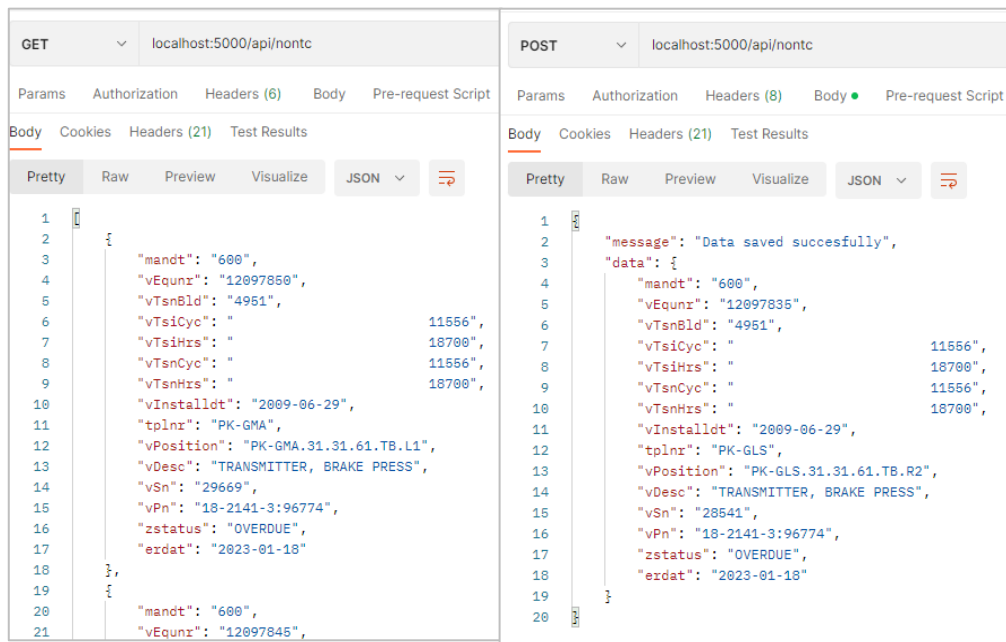


Figure 14. Testing API GET & POST Non Time-Controlled

Figure 14 shows that Non Time-Controlled (Non-TC) testing using the GET function successfully retrieved data from the SAP database JSON file. This table has a relationship with the connector table. The Maintenance Plan field is used as a Primary Key which will become a Foreign Key when pulled into the Connector table. NON-TC testing using the POST function has succeeded in storing data on the website database based on data retrieved from the SAP JSON file.

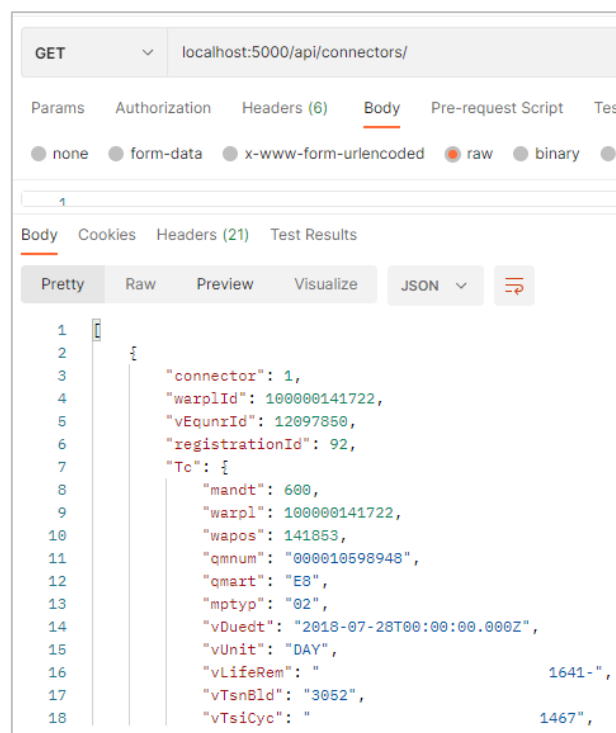


Figure 15. Testing API GET Connectors

Figure 15 shows the results of testing the Connector using the GET function to get data from the Connector table. This table contains data derived from the TC, NON-TC, and Registrations table relations. Each table has the same field, namely Registration. Therefore, Registration is used as the main field in this application.

Table 3. REST API Testing with POSTMAN

No	Test case	Results
1	Testing API GET & POST Operator	Successful add Data (Fig 7)
2	Testing API DELETE & PATCH Operator	Successful update the Data (Fig 8)
3	Testing API GET & POST Aircraft Types	Successful add new data on Aircraft Types (Fig 9)
4	Testing API DELETE & PATCH Aircraft Types	Successful update the Aircraft Types data (Fig 10)
5	Testing API GET & POST Registrations	Successful add new data on Component Status (Fig 11)
6	Testing API DELETE & PATCH Registrations	Successful update the Component Status data (Fig 12)
7	Testing API GET & POST Time-Controlled	Successful retrieve the data from JSON file database SAP (Fig 13)
8	Testing API GET & POST Non Time-Controlled	Successful retrieve the data from JSON file database SAP (Fig 14)
9	Testing API GET Connectors	Successful get the data from Connector Table (Fig 15)

Table 3 shows 9 test cases on the REST API using POSTMAN. All of these test cases successfully run according to user expectations. The testing phase in this study was limited to testing the functionality of applications developed using POSTMAN.

CONCLUSION

This research produces a website-based application used at PT GMF Aero Asia, Tbk. Before the deployment of this application, this application had gone through REST API testing using POSTMAN to validate the process of displaying status automatically, processing Aircraft Type and Components data, displaying notifications of differences between old and new data, and being able to receive data sent from SAP SWIFT.

Due to limitations of time and resources in this study, the suggestions for further research development are: 1) Evaluating the application with User Acceptance Test (UAT), 2) Evaluating the user interface using the User Experience Questionnaire, 3) Developing the Front-end and Back-end user interfaces.

ACKNOWLEDGEMENT

We want to thank PT Garuda Maintenance Facility Aero Asia, Tbk, for the opportunity and support given in this research. Their contribution played an essential role in the successful completion of this research. We also express our sincere gratitude to our colleagues from the Multimedia Nusantara University Information Systems Department, whose input and expertise greatly assisted us during the research process. Their valuable contributions helped shape the direction of this study and improve its overall quality.

REFERENCE

- Anugrah, I. G., & Fakhruddin, M. A. R. I. (2020). Development Authentication and Authorization Systems of Multi Information Systems Based REst API and Auth Token. *Innovation Research Journal*, 1(2), 127. <https://doi.org/10.30587/innovation.v1i2.1927>
- Chandra, S., Sutomo, R., & Wiratama, J. (2023). Design And Development of Car Sparepart Sales Information System For Web-Based Using RAD Method On UMKM Sinar Seroja. *G-Tech: Jurnal Teknologi Terapan*, 7(2), 494–503. <https://doi.org/10.33379/gtech.v7i2.2019>
- Gunadi, G. (2021). Rancang Bangun Sistem Peminjaman Laptop dengan Metode Extreme Programming Menggunakan Framework Bootstrap. *Ultima InfoSys: Jurnal Ilmu Sistem Informasi*, 12(1), 74–86. <https://doi.org/10.31937/si.v12i1.2087>
- Ramadhanu, P. B., & Priandika, A. T. (2021). Rancang Bangun Web Service Api Aplikasi Sentralisasi Produk Umkm Pada Uptd Plut Kumkm Provinsi Lampung. *Jurnal Teknologi*

- Dan Sistem Informasi (JTSI)*, 2(1), 59–64. <http://jim.teknokrat.ac.id/index.php/JTSI>
- Samosir, J. (2020). Formulation of Business Strategy Based on SWOT Analysis on PT. Garuda Maintenance Facility Aeoroasia . Tbk. *Palarch's Journal Of Archaeology Of Egypt/Egyptology*, 17(2020), 1193. Formulation strategy, Business, SWOT, PT GMF AeroAsia Tbk
- Saputra, A., & Astuti, D. Y. (2018). Analisis Pengaruh Struktur Html Terhadap Rangkaian Search Engine Result Page. *Jurnal Mantik Penusa*, 2(2), 34–67. <https://google.co.id>
- Sha, P., Chen, S., Zheng, L., Liu, X., Tang, H., & Li, Y. (2020). Design and Implement of Microservice System for Edge Computing. *IFAC-PapersOnLine*, 53(5), 507–511. <https://doi.org/10.1016/j.ifacol.2021.04.137>
- Suryatman, T. H., Hakim, M. Y., & Hartono, H. (2019). Perbandingan Penerapan Metode Waktu Baku Dengan Pendekatan Time Stop Pada Sistem Swift Sap Enterprises Pada Workshop Perawatan Passenger Seat Boeing 747-400 Di Pt. Gmf Aeroasia Tbk. *Jurnal Teknik*, 8(2). <https://doi.org/10.31000/jt.v8i2.2227>
- Wijaya, F. F. (2023). *Pengembangan Aplikasi Backend MVC Menggunakan Clean Architecture Layer Framework NestJs* [Universitas Muhammadiyah Malang]. <https://etd.umm.ac.id/id/eprint/403>
- Wiratama, J., & Ikana Desanti, R. (2022). The Development of Web-based Sales Reporting Information Systems using Rapid Application Development Method. *Ultima Infosys : Jurnal Ilmu Sistem Informasi*, 13(2), 110–116.
- Wiratama, J., Santoso, H., & Clairence. (2023). Developing a Class Scheduling Mobile Application for Private Campus in Tangerang with the Extreme Programming (XP) Model. *G-Tech : Jurnal Teknologi Terapan*, 7(2), 484–493. <https://doi.org/https://doi.org/10.33379/gtech.v7i2.2288>
- Yuwono, M. R. A., & Saptadi, S. (2021). Min-Max Dalam Pengendalian Persediaan Komponen Pesawat Terbang Boeing 737ng (Studi Kasus : Pt Garuda Maintenance Facility Aeroasia Tbk). *Jurnal Online Teknik Industri*.