

PENERAPAN ARSITEKTUR MICROSERVICE PADA SISTEM TATA KELOLA MATAKULIAH PROYEK POLITEKNIK POS INDONESIA

Mohammad Harry Khomas Saputra¹, Luthfi Muhammad Nabil.²

^{1,2}Program Studi D4 Teknik Informatika, Politeknik Pos Indonesia,

Jl. Sari Asih No.54, Sarijadi, Kec. Sukasari, Kota Bandung, Jawa Barat 40151, Indonesia Tlp +6222

2009570, Fax. +6222 200 9568

harrysaputra@poltekpos.ac.id

ABSTRAK

Sistem Tata Kelola Proyek pada Politeknik Pos Indonesia merupakan sistem informasi yang mengatur tata kelola matakuliah proyek yang berjalan pada kampus Politeknik Pos Indonesia prodi D4 Teknik Informatika. Sistem tersebut sudah diuji coba dan sampai akhir tidak memiliki begitu banyak masalah pada operasional baik dari awal dimulainya kegiatan proyek sampai diakhirinya kegiatan proyek. Namun, terdapat sebuah persoalan mengenai lamanya proses pengembangan fitur dan penanganan kerusakan dari sistem informasi karena banyaknya hal yang harus disesuaikan baik dalam bagian depan maupun bagian belakang. Hal ini dapat menjadi masalah saat pengembang baru dibutuhkan untuk melakukan pengembangan fitur dalam sistem informasi dikarenakan banyak komponen yang harus disesuaikan seiring berkembangnya aplikasi. Untuk mengatasi hal tersebut, akan diterapkan arsitektur *microservice* pada sistem informasi agar seorang pengembang lebih mudah untuk mempelajari bahasa yang digunakan pada sistem informasi. Pada konsep ini, akan digunakan bahasa pemrograman Javascript

dengan framework *ExpressJS* untuk pengembangan pada bagian belakang.

Kata kunci: *ExpressJS, Microservice, API.*

I. PENDAHULUAN

Sistem Informasi Tata Kelola Proyek Politeknik Pos Indonesia merupakan sebuah sistem informasi yang membantu institusi dalam pengelolaan kegiatan proyek. Sistem ini memiliki fungsi umum dalam melakukan pengelolaan data kegiatan maupun monitoring aktivitas yang dilakukan pengguna dalam melakukan sebuah kegiatan.

Sistem Informasi ini sudah berjalan dengan menggunakan arsitektur monolithic yang menyatukan seluruh bagian ke dalam satu sistem. Arsitektur tersebut digunakan agar dapat dengan mudah memajemen *codebase* karena tidak perlu mengerjakan banyak aplikasi sekaligus. Namun seiring dengan permintaan penambahan fitur untuk sistem informasi, terdapat persoalan mengenai lambatnya proses

pengembangan sistem dan penanganan kerusakan dari sistem informasi karena banyaknya hal yang harus disesuaikan baik dalam bagian depan maupun bagian belakang.

Jika hal tersebut dibiarkan, akan mengakibatkan sebuah sistem menjadi lebih sulit dikelola seiring dengan banyaknya pengembangan - pengembangan yang dilakukan dari waktu ke waktu dan menyebabkan menumpuknya dokumentasi sistem informasi. Tidak hanya itu, pengembang baru sistem informasi akan kesulitan saat melakukan pengembangan maupun penanganan kesalahan sistem karena banyaknya hal yang harus dipelajari dan wajib mengetahui seluruh komponen dari sistem informasi untuk menyelesaikan masalah dengan tepat.

Untuk menghindari masalah tersebut, diperlukan sebuah konsep yang dapat membuat sebuah sistem informasi lebih mudah untuk dipecah menjadi beberapa bagian dan membuat pengembang lebih mudah melakukan pengembangan sistem informasi seiringan dengan terpisahnya komponen - komponen dari aplikasi. Di saat yang bersamaan, penanganan kesalahan dari aplikasi akan lebih mudah dan tidak mengganggu komponen yang lain dikarenakan terpisahnya satu komponen dengan komponen - komponen lain.

Atas dasar hal tersebut, saya akan melakukan pengembangan sistem dengan tema "Penerapan Arsitektur *Microservice* Pada Sistem Tata Kelola Proyek Politeknik Pos Indonesia" yang dapat memberikan solusi terhadap masalah yang dihadapi.

II. LANDASAN TEORI

2.1. API

API merupakan sebuah merupakan sistem penyedia layanan yang berjalan pada sisi server. Namun berbeda dengan sistem yang memanfaatkan arsitektur monolitik, API berjalan terpisah dengan sistem bagian sehingga tidak hanya satu sistem saja yang dapat berkomunikasi dengan API.

API biasa digunakan untuk menyediakan layanan kecil atau membuat sebuah layanan dapat dikembangkan lebih jauh sesuai dengan apa yang dibutuhkan sehingga menjadi salah satu andalan sistem berskala besar yang membutuhkan pengembangan mudah [12][13]. Beberapa API menggunakan token untuk membatasi pengguna sehingga tidak sembarang orang dapat mengakses isi dari layanan pada API.

2.2. HTTP

HTTP merupakan protokol untuk pertukaran data antara client dan server. HTTP merupakan protokol yang paling umum digunakan oleh bahasa pemrograman karena kemudahan yang diberikan saat melakukan komunikasi menggunakan protokol tersebut.

Penerapan HTTP pada setiap bahasa pemrograman baik untuk website maupun mikrokontroler sekalipun sudah dipermudah

berkat standarisasi dan library yang dibangun pada setiap bahasa yang membutuhkannya.

2.3. JSON

JSON merupakan sebuah struktur data yang dipakai untuk melakukan pertukaran data. JSON dibangun dengan menggunakan basis bahasa pemrograman Javascript dan memiliki tujuan untuk mempermudah saat dibaca oleh manusia atau sebuah sistem.

Bahasa pemrograman pada umumnya sudah menerapkan struktur data ini untuk saling berkomunikasi dengan sistem yang lain karena ringannya struktur data ini dikarenakan penulisannya yang tidak berlebihan berbeda dengan sesamanya yaitu Java XML dan kebutuhan parameter yang hanya memerlukan kunci yang dibutuhkan saja tidak perlu diurutkan.

2.4. Javascript

Javascript merupakan sebuah bahasa pemrograman yang berjalan dalam mesin yang disediakan pada browser. Javascript biasa digunakan untuk melakukan aksi - aksi seperti mengisi data pada konten website, melakukan animasi, dan lain sebagainya pada browser yang membuat bahasa ini menjadi cukup diminati bagi pengembang website.

2.5. NodeJS

NodeJS merupakan software yang membuat bahasa pemrograman Javascript dapat berjalan pada sisi server. Berkat hal tersebut, NodeJS dapat melakukan pekerjaan - pekerjaan pada server seperti halnya bahasa pemrograman PHP, Ruby, Perl, dan lain sebagainya. NodeJS menggunakan mesin Engine V8 yang dibangun Google untuk melakukan kompilasi pada Javascript sehingga dapat dijalankan oleh server. Dengan begitu, NodeJS dapat menjadi satu layanan tersendiri.

Software ini menjadi populer karena memudahkan programmer dalam membangun aplikasi dengan alasan basis depan dan belakang dapat disamakan hanya dengan satu bahasa saja bahkan struktur datanya sendiri yang menggunakan JSON. Selain itu, NodeJS memiliki Package Manager untuk mengambil library - library yang ada bernama NPM. Library

yang disediakan oleh NPM memiliki banyak pengguna karena mudahnya mengintegrasikan library yang diambil pada NPM.

2.6. ExpressJS

ExpressJS merupakan sebuah framework yang dibangun dengan basis Javascript dan dikompilasi pada software NodeJS. ExpressJS biasanya digunakan untuk membangun API dan menampung layanan-layanan yang ada. Framework ini banyak digunakan untuk kalangan pengembang pemula karena mudahnya dalam membangun layanan-layanan dan mengintegrasikannya pada satu sistem yang dibangun pada ExpressJS.

2.7. Asynchronous Programming

Asynchronous Programming merupakan alur pemrograman yang mengeksekusi sintaks-sintaks tanpa perlu menunggu sintaks lainnya. Asynchronous Programming menjadi terobosan baru untuk membuat sebuah perintah dapat berjalan tanpa mengganggu perintah lainnya sehingga sistem dapat berjalan dengan lebih cepat.

2.8. JSON Web Token

JSON Web Token merupakan sebuah token yang biasa digunakan sebagai media autentikasi antara sebuah aplikasi depan dengan penyedia layanan. Token ini menggunakan tipe data string dan didalamnya mengandung data-data penting yang digunakan oleh aplikasi depan untuk berkomunikasi dengan penyedia layanan.

Isi dari JSON Web Token dinamakan payload dan dapat dilihat dengan mudah namun tidak dapat diubah karena token ini menggunakan sebuah kunci yang berfungsi untuk mengidentifikasi apakah sebuah token yang dikirim adalah token yang sama sehingga membatasi pengiriman dan perubahan informasi yang dimiliki oleh token.

Biasanya penggunaan JSON Web token untuk autentikasi pada sistem yang menggunakan basis REST sebagai konsep struktur komunikasi antar sistem.

III. ANALISIS DAN PERANCANGAN SISTEM

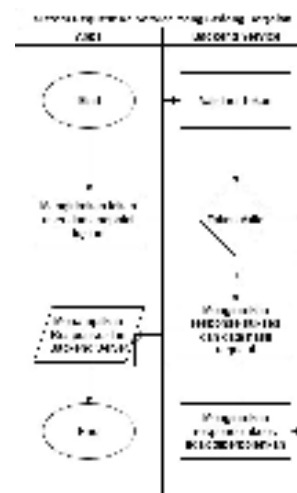
Analisis yang dilakukan pada penelitian ini merupakan analisis yang berhubungan dengan prosedur yang dilakukan untuk

mengembangkan aplikasi yang bermula dari arsitektur *monolithic* ke arsitektur *microservice*, yang terdiri dari proses sistem yang sedang berjalan dan proses yang akan dibangun.

3.1 Analisis Prosedur yang berjalan (Flowmap)

Analisis yang sedang berjalan merupakan tahap awal dari perancangan sistem. Pada tahap analisis sistem yang berjalan ini akan dibahas mengenai prosedur yang berjalan saat ini.

Analisis Sistem Yang Berjalan Pada Proses Request ke Service

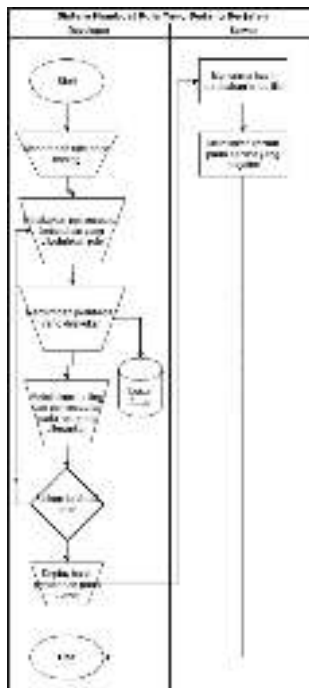


Gambar 1. Flowmap Analisis Sistem Yang Sedang Berjalan Pada Request ke Service

Pada analisis yang sedang berjalan pada proses request ke service ini dilakukan oleh *apps* ke backend service. Proses yang terjadi dalam *flowmap* dapat dijelaskan sebagai berikut

1. *Apps* mengirimkan token user dan token tujuan ke backend service.
2. Backend service memvalidasi token. Jika token valid maka akan mengirimkan respon sukses dan data hasil request. Jika tidak maka akan mengirimkan respon akses tidak diperbolehkan
3. *Apps* menampilkan respon dari backend server yang didapat.

3.1.2 Analisis Sistem Yang Berjalan Pada Proses Menambah Role



Gambar 2. Flowmap Analisis Sistem Yang Sedang Berjalan Pada Proses Menambah Role

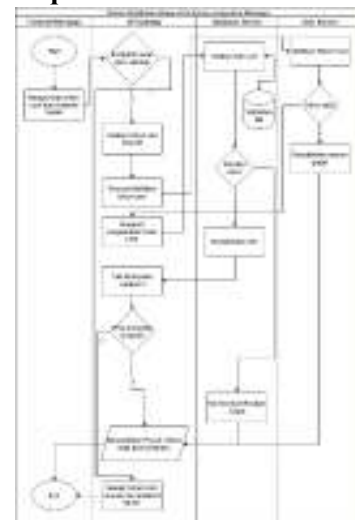
Pada analisis yang sedang berjalan pada proses request ke service ini dilakukan oleh *apps* ke backend service. Proses yang terjadi dalam *flowmap* dapat dijelaskan sebagai berikut:

1. *Apps* mengirimkan token user dan token tujuan ke backend service.
2. Backend service memvalidasi token. Jika token valid maka akan mengirimkan respon sukses dan data hasil request. Jika tidak maka akan mengirimkan respon akses tidak diperbolehkan
3. *Apps* menampilkan respon dari backend server yang didapat.

3.2 Analisis Sistem yang akan Dibangun (Flowmap)

Analisis prosedur pada kegiatan proyek akan dibangun pada Politeknik Pos Indonesia dapat digambarkan dalam bentuk *flowmap*

3.2.1 Analisis Sistem Yang Akan Dibangun Pada Proses Request ke Service



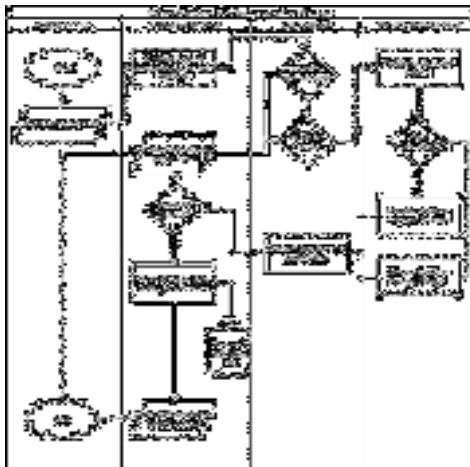
Gambar 3. Flowmap Analisis Sistem Yang Akan Dibangun Pada Proses Request ke Service

Pada analisis sistem yang akan dibangun pada melakukan request ke service terdapat actor *Frontend web apps*, *API Gateway*, *Validation Service*, dan *User Service*. Proses yang terjadi dalam *flowchart* dapat dijelaskan sebagai berikut:

1. *Frontend web apps* mengirimkan request beserta token ke *API Gateway*.
2. *API Gateway* melakukan validasi dari endpoint, jika endpoint ada maka langsung memvalidasi token yang direquest ke *User Service*.
3. *User Service* melakukan validasi token.
4. Jika token valid maka *User Service* akan mengirimkan respon sukses dan jika tidak maka akan dikirimkan respon gagal.
5. *API Gateway* melakukan request pengambilan role dari user.
6. *Validation Service* melakukan validasi role pada beberapa service. Jika role didapatkan maka akan dikembalikan role yang didapat.

API Gateway melakukan pengecekan ulang pada role yang didapat ke endpoint tujuan. Jika role terdaftar pada endpoint maka *API Gateway* akan memberikan hasil request dari service endpoint yang dituju.

3.2.2 Analisis Sistem Yang Akan Dibangun Pada Proses Membuat Role Baru

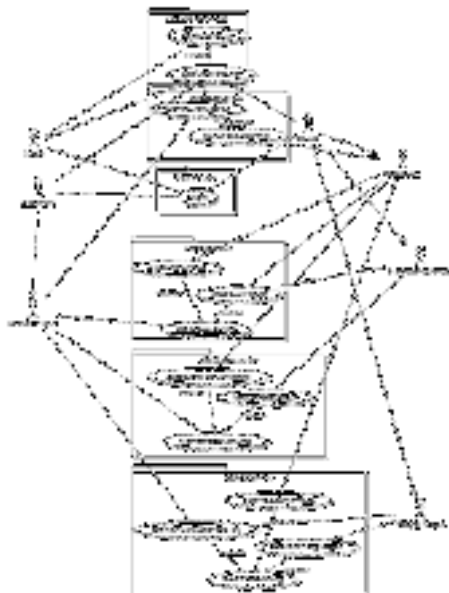


Gambar 4. Flowmap Analisis Sistem Yang Akan Dibangun Pada Proses Membuat Role Baru

Pada analisis sistem yang akan dibangun pada prosedur membuat role baru. Proses yang terjadi dalam *flowchart* dapat dijelaskan sebagai berikut :

3.2.3 Use Case Diagram

Use case diagram adalah sebuah diagram yang menunjukkan suatu kelompok use case dan actors beserta dengan relationships-nya atau hubungannya secara kompleks.



Gambar 5. Use Case Diagram

1. *Admin* menginput data role baru.
2. *Validation Service* akan memverifikasi tujuan service atau sumber data dari role ke *API Gateway*.
3. Pada *API Gateway*, jika service ditemukan maka akan langsung melakukan pengecekan ke endpoint. Jika endpoint cocok maka langsung dilanjutkan ke verifikasi Path di target Service.
4. Jika path tersedia, maka target service akan mengembalikan respon sukses.
5. *API Gateway* mengembalikan respon yang didapat.

Pada *Validation Service*, jika respon yang didapat yaitu sukses maka akan langsung dimasukkan data yang diinputkan sebagai role baru

Kesimpulan

Setelah melakukan pengujian arsitektur *microservice* maka dapat disimpulkan sebagai berikut :

1. Berdasarkan hasil pengujian arsitektur *microservice*. Telah berhasil menjawab masalah penelian ini yaitu bagaimana menerapkan arsitektur *microservice* pada sistem.
2. Efek yang dihasilkan saat mengembangkan sistem saat arsitektur *microservice* lebih tidak mengganggu sistem lain saat melakukan pengembangan sehingga dapat memperkecil ruang lingkup dari pengembangan

DAFTAR PUSTAKA

Jurnal Teknologi Volume 9 No. 1 Januari 2017
ISSN : 2085 – 1669 Website :
jurnal.umj.ac.id/index.php/jurtek e-ISSN :
2460 – 0288

Taibi, Davide, et al. "From monolithic systems to microservices: an assessment framework." arXiv preprint arXiv:1909.08933 (2019).

De Laurentis, Lorenzo. "From Monolithic Architecture to Microservices Architecture." 2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW). IEEE, 2019.

Xu, Ronghua, et al. "Blendmas: A blockchain-enabled decentralized microservices architecture for smart public safety." 2019 IEEE International Conference on Blockchain (Blockchain). IEEE, 2019.

Yan, Lei, et al. "SatEC: A 5G satellite edge computing framework based on microservice architecture." *Sensors* 19.4 (2019): 831.

Brandón, Álvaro, et al. "Graph-based root cause analysis for service-oriented and microservice architectures." *Journal of Systems and Software* 159 (2020): 110432.

Zhang, He, et al. "Microservice architecture in reality: An industrial inquiry." 2019 IEEE International Conference on Software Architecture (ICSA). IEEE, 2019.

Kurniawan, Dedy, et al. "Implementasi Arsitektur MICROSERVICE: STUDI KASUS PADA PENGEMBANGAN Surat Keterangan Pendamping Ijazah di Lingkungan Fakultas Unsri." *Annual Research Seminar (ARS)*. Vol. 5. No. 1. 2020.

El Kholy, Mohamed, and Ahmed El Fatatry. "Framework for interaction between databases and microservice architecture." *IT Professional* 21.5 (2019): 57-63.

Dewanta, Favian, and Reza Afshari. "Migrasi Mikroservis Pada Fog Computing Untuk Mendukung Kinerja Komputasi Robot dengan Cakupan Pergerakan yang Luas." *MULTINETICS* 6.1 (2020): 78-85.

Li, Jingming, et al. "Integration of Building Information Modeling and Web Service Application Programming Interface for assessing building surroundings in early design stages." *Building and Environment* 153 (2019): 91-100.

Xu, Hao, Hong Wang, and Suo Zhang. "Application programming interface (api) service apparatus and application programming interface (api) service system." U.S. Patent Application No. 16/181,927.

Rulloh, Amin, Dewi Erla Mahmudah, and Herman Kabetta. "Implementasi REST API pada Aplikasi Panduan Kepaskibraan Berbasis Android." *Teknikom: Teknologi Informasi, Ilmu Komputer dan Manajemen* 1.2 (2017): 8589.

Choirudin, Romi, and Ahmat Adil. "Implementasi Rest Api Web Service dalam Membangun Aplikasi Multiplatform untuk Usaha Jasa." *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer* 18.2 (2019): 284-293.

Rombe, Ady Nopaldi, LM Fid Aksara, and La Surimi. "ANALISIS PERBANDINGAN REAL TIME STREAMING PROTOCOL (RTSP) DAN HYPERTEXT TRANSFER PROTOCOL (HTTP) PADA LAYANAN LIVE VIDEO STREAMING." *semanTIK* 5.1 (2019): 149-156.

Kurniawan, Nico, Natanael Benino, and Enda Wista Sinuraya. "Membaca Dan Mengirim Data Melalui Protokol HTTP Menggunakan Library WebClient Arduino." *Transmisi* 17.2 (2015): 59-62.

Pratama, Ocki Bagus, Adhitya Bhawiyuga, and Kasyful Amron. "Pengembangan Perangkat Lunak IoT Cloud Platform Berbasis Protokol Komunikasi HTTP." *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* eISSN 2548 (2018): 964X.

Arifin, Muhamad Nur, Mochammad Hannats Hanafi Ichsan, and Sabriansyah Rizqika Akbar. "Monitoring kadar gas berbahaya pada kandang

- Husnibes Muchtar, Asep Hidayat : implementasi wavecom dalam monitoring beban listrik berbasis mikrokontroler Jurnal Teknologi 9 (1) pp 1- 5 © 2017 ayam dengan menggunakan protokol HTTP dan ESP8266." Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN 2548 (2018): 964X.
- Friesen, Jeff. "Introducing JSON." Java XML and JSON. Apress, Berkeley, CA, 2019. 187203.
- Haeck, Wim. A simple json format for storing and exchanging code calculation and experimental results. No. LA-UR-19-32580. Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2019.
- Malik, Rabee Sohail, Jibesh Patra, and Michael Pradel. "NI2type: inferring javascript function types from natural language information." 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE). IEEE, 2019.
- Nielsen, Benjamin Barslev, Behnaz Hassanshahi, and François Gauthier. "Nodest: feedback-driven static analysis of Node. js applications." Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 2019.
- Chopra, Umesh Kumar, Ayush Kumar Rathore, and Rajiv Pandey. "Rendering Blockchain Immutability in Chatserver: A Node. js Approach." Decision Analytics Applications in Industry. Springer, Singapore, 2020. 147-155.
- Wilson, Eddy. MERN Quick Start Guide: Build Web Applications with MongoDB, Express. js, React, and Node. Packt Publishing Ltd, 2018.
- Mardan, Azat. "Using Express. js to create Node. js web apps." Practical Node. js. Apress, Berkeley, CA, 2018. 51-87.
- Luong, Tran Thanh, and Le My Canh. "JAVASCRIPT ASYNCHRONOUS PROGRAMMING." Hue University Journal of Science: Techniques and Technology 128.2B (2019): 5-16.
- Gunawan, Rohmat, and Alam Rahmatulloh. "JSON Web Token (JWT) untuk Authentication pada Interoperabilitas Arsitektur berbasis RESTful Web Service." JEPIN (Jurnal Edukasi dan Penelitian Informatika) 5.1 (2019): 74-79.
- B. Soediono, A. Mustofa, T. Informatika, and U. D. Nuswantoro, "Analisis Dan Perancangan Sistem Informasi E-Education Berbasis Web Di Sma Pembangunan Mranggen," J. Chem. Inf. Model., vol. 53, no. 2004, pp. 1–6, 2014, doi: 10.1017/CBO9781107415324.004.
- A. Febriansyah, "Tinjauan Atas Proses Penyusunan Laporan Keuangan Pada Young Entrepreneur Academy Indonesia Bandung," J. Ris. Akunt., vol. 8, no. 2, 2017, doi: 10.34010/jra.v8i2.525.
- M. Abdurahman, "Sistem Informasi Data Pegawai Berbasis Web Pada Kementerian Kelautan Dan Perikanan Kota Ternate," J. Ilm. Ilk. - Ilmu Komput. Inform., vol. 1, no. 2, pp. 70–78, 2018, doi: 10.47324/ilkominfo.v1i2.10.
- K. Indriani and I. Marliana, "Simple Addtive Weighting (SAW) Untuk Pemilihan Penyewaan Wheel Loader," JIMP - J. Inform. Merdeka Pasuruan, vol. 5, no. 1, pp. 15–20, 2020, doi:10.37438/jimp.v5i1.244.