
**PENERAPAN SISTEM TERINTEGRASI MENGGUNAKAN RESTFUL
API PADA DEALER MANAGEMENT SYSTEM PANCA NIAGA
SEI PIRING**

Muhammad Iqbal¹, Nurwati²

STMIK Royal, Kisaran

e-mail: ¹codeegoc@gmail.com, ²nurwati763@gmail.com

Abstract: *Integrated systems are processes related to combining different subsystems or components as one big system. This ensures that each integrated subsystem functions as required. Currently, the RESTful API has become the standard API that is widely used when we create systems that need to provide APIs for other parties. This research was conducted at a PNM Dealer (Panca Niaga Mandiri) which is an Authorized Honda Dealer with the aim of facilitating data communication between various divisions so that the data is consistent and has data integrity, as well as preventing waste data (repetitive data) in making reports. The Panca Niaga Mandiri Sei Plate dealer experienced several obstacles in integrated data management. Transfer of data between divisions is limited in terms of information technology, system activities that are running have not yet optimized a centralized database system.*

Keywords: *Restful API, Microservice, Dealer, Web Service, Integrated System*

Abstrak: Sistem terintegrasi merupakan proses yang berkaitan dengan menggabungkan subsistem atau komponen yang berbeda sebagai satu sistem besar. Ini memastikan bahwa setiap subsistem terintegrasi berfungsi sesuai kebutuhan. Saat ini RESTful API sudah menjadi standard API yang banyak digunakan ketika kita membuat sistem yang butuh menyediakan API untuk pihak lain. Penelitian ini dilakukan pada Dealer PNM (Panca Niaga Mandiri) merupakan Dealer Resmi Honda bertujuan untuk memudahkan komunikasi data antar berbagai divisi sehingga data tersebut konsisten dan memiliki keutuhan data, serta pencegahan data sampah (data berulang) dalam pembuatan laporan. Dealer Panca Niaga Mandiri Sei Piring mengalami beberapa hambatan dalam pengelolaan data yang terintegrasi. Perpindahan data antar divisi terbatas dalam hal teknologi informasi, kegiatan sistem yang berjalan belum mengoptimalkan sistem basis data terpusat.

Kata kunci: Restful API, Microservice, Dealer, Web Service, Sistem Terintegrasi

PENDAHULUAN

Sistem terintegrasi digunakan untuk menambah nilai sistem melalui fungsionalitas baru yang disediakan dengan menghubungkan fungsi sistem yang berbeda. Saat ini, penerapan sistem terintegrasi bukanlah yang asing. Dengan kepopuleran dunia internet dan disertai perkembangan pemrograman saat ini, menjadikan integrasi sistem semakin banyak diterapkan khususnya para perusahaan besar yang membutuhkan kolaborasi antar bisnis. Sebagai contoh,

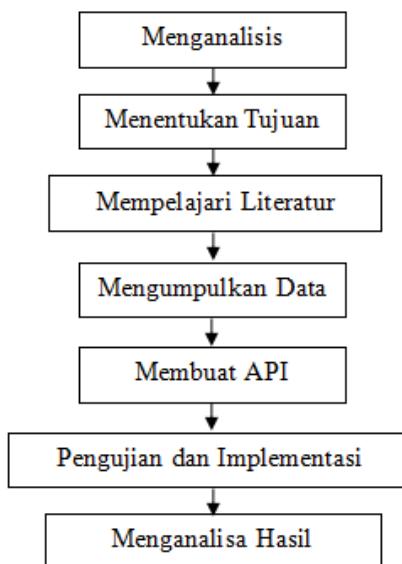
Penerapan sistem terintegrasi seperti aplikasi e-commerce (belanja online) yang di dalamnya terdapat beberapa layanan pembayaran dan logistic, dimana Pembayaran dan logistic terhubung langsung dengan aplikasi e-commerce. Sistem pembayaran menggunakan pihak ketiga seperti perbankan dan system logistic untuk mengelola pengiriman barang. Penyedia Aplikasi e-commerce tidak perlu membuat secara mandiri untuk payment gateway dan pengelolaan logistic, cukup dengan memanfaatkan.

eknologi dari pihak bank dan logistic (Iqbal et al, 2022).

Seiring dengan berkembangnya teknologi informasi, banyak teknologi baru yang muncul, salah satunya teknologi mobile berbasis Android yang memudahkan pengguna dalam mengakses informasi yang dibutuhkan (Arimbawa et al., 2019). Teknologi dapat diterapkan untuk mendapatkan informasi dari berbagai sumber data. Salah satu bentuk penerapan teknologi berada pada sistem informasi (Utomo et al., 2020). Dealer Panca Niaga Mandiri Sei Piring mengalami beberapa hambatan dalam pengelolaan data yang terintegrasi. Perpindahan data antar divisi terbatas dalam hal teknologi informasi, kegiatan sistem yang berjalan belum mengoptimalkan sistem basis data terpusat. Tujuan penelitian ini membuat sistem terintegrasi agar dapat tiap divisi berbagi data dengan menjaga konsistensi data yang telah tersimpan pada basis data. Penggunaan Konsep API web service ini dapat menguntungkan bagi pengguna untuk dapat lebih mudah dalam pengaksesan sebuah data, misalnya dengan adanya web service pada sebuah sistem website terdapat perubahan data, maka dengan tersistem data yang ada di aplikasi Android secara otomatis juga dapat berubah, dengan adanya API yang dibuat pengguna lain tidak dapat secara langsung mengakses database kita malainkan mereka harus request melalui internet lalu mengakses API kita dan kemudian direspon oleh API (Baharuddin et al., 2022)

METODE

Metodologi penelitian ini dimulai dari Menganalisis Masalah, Menentukan Tujuan, Mempelajari Literatur, Mengumpulkan Data, Membuat API, Pengujian dan Implementasi Sistem, dan Menganalisa hasil Penelitian



Gambar. Metode Penelitian

Menganalisis masalah, kemampuan untuk mengenal elemen-elemen situasi dalam permasalahan dan memahami komponen mana saja yang kritis; kemampuan untuk mengenal aktivitas kritis yang dilakukan agar dapat mengurutkan (*breakdown*) proses proses aktivitas tersebut dalam beberapa komponen aktivitas. Peneliti menggunakan metode Microservices sebagai arsitektur pembangunan sistem, microservice merupakan arsitektur aplikasi yang memfokuskan pada pembagian tugas aplikasi ke dalam beberapa bagian kecil yang terpisah dan independen, yang masing-masing dapat dikembangkan, di-deploy, dan di-maintenance secara terpisah (Newman, 2015). Arsitektur Microservice adalah sebuah arsitektur dimana mengembangkan aplikasi dilakukan dalam bentuk web service yang lebih kecil, yang saling berkomunikasi satu sama lain. Arsitektur microservice merupakan alternatif arsitektur yang lebih terukur dan fleksibel (Dharma Handayani & Uminingsih, 2020).

Tujuan utama dari microservices adalah untuk mempermudah pengembangan aplikasi dengan

mengeliminasi ketergantungan antar bagian, sehingga setiap bagian dapat dikembangkan dan di-update secara terpisah tanpa mempengaruhi bagian lainnya (Eberhard et al., 2018).

Microservices juga memungkinkan pengembangan aplikasi dengan menggunakan tim yang lebih kecil, karena setiap bagian dapat dikelola oleh tim yang terpisah (Newman, 2015). Selain itu, microservices juga memungkinkan pengembangan aplikasi dengan menggunakan teknologi yang berbeda untuk setiap bagian, sehingga dapat memanfaatkan keunggulan dari setiap teknologi (Eberhard et al., 2018). Arsitektur microservice merupakan alternatif arsitektur yang lebih terukur dan lebih fleksibel. Pada arsitektur microservice, sistem informasi dirancang untuk terdistribusi dan menyediakan layanan secara lebih fokus dan spesifik (Iqbal et al., 2022).

Meskipun memiliki banyak keuntungan, ada juga beberapa tantangan dalam implementasi microservices, seperti:

Kompleksitas sistem

Microservices membutuhkan integrasi yang lebih kompleks dibandingkan dengan aplikasi monolitik, karena setiap bagian harus terintegrasi dengan baik agar aplikasi dapat bekerja secara efektif (Eberhard et al., 2018).

Testing

Pembagian aplikasi ke dalam beberapa bagian kecil membutuhkan proses testing yang lebih kompleks dan memakan waktu lebih lama (Newman, 2015).

Monitoring

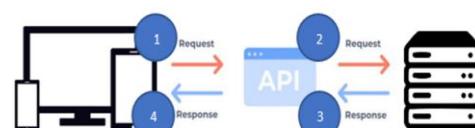
Microservices membutuhkan sistem monitoring yang lebih kompleks untuk memantau kinerja setiap bagian dan menangani masalah yang mungkin terjadi (Eberhard et al., 2018).

Meskipun demikian, banyak perusahaan yang telah berhasil mengatasi tantangan tersebut dan berhasil

mengimplementasikan microservices dengan sukses, seperti Netflix, Amazon, dan Spotify (Newman, 2015). Dengan demikian, dapat disimpulkan bahwa microservices merupakan sebuah arsitektur aplikasi yang memungkinkan pengembangan aplikasi yang lebih fleksibel dan efisien, tetapi juga membutuhkan integrasi yang lebih kompleks dan sistem monitoring yang lebih canggih (Eberhard et al., 2018).

HASIL DAN PEMBAHASAN

Beberapa cara agar sistem dapat terintegrasi, seperti File Sharing, Database Sharing, RPI (Remote Procedure Invocation) atau yang lebih dikenal dengan API, dan Messaging. Teknologi yang digunakan pada penelitian ini yaitu API dengan metode Rest API. REST merupakan singkatan dari Representational State Transfer, yang dikenalkan oleh Roy Fielding pada tahun 2000 melalui disertasinya. RESTful API menggunakan HTTP untuk dapat berkomunikasi. Penggunaan HTTP saat ini semakin intent disebabkan kemudahan internet menjadikan RESTfull API semakin populer. Saat ini RESTful API sudah menjadi standard API yang banyak digunakan ketika kita membuat sistem yang butuh menyediakan API untuk pihak lain. Beberapa kaidah yang ditetapkan oleh Roy Fielding agar web services benar-benar dikatakan sesuai dengan RESTful API, seperti Berbasis Client-Server, Stateless, Cacheable, Uniform interface, Layered System dan Code on Demand. Mekanisme dari RESTful API dapat dilihat melalui diagram berikut:



Gambar. Diagram Alur Restful API

Pada Gambar Menunjukkan bahwa Client mengakses API. Kemudian API melakukan Request ke Server. Server

Memberikan Response ke API dan API menyampaikan Response ke aplikasi. Dalam pembuatan RESTful API sangat populer itu menggunakan JSON. JSON itu sendiri singkatan dari JavaScript Object Notation, adalah suatu format ringkas pertukaran data komputer. Formatnya berbasis teks dan terbaca manusia serta digunakan untuk merepresentasikan struktur data sederhana dan larik asosiatif. JSON sangat populer digunakan dalam pembuatan RESTful API saat ini. API memiliki bermacam jenis seperti, Public API atau dikenal dengan Open API, Private API, Partner API dimana penggunanya hanya sebatas pihak yang sudah memiliki izin penggunaanya. dan Composite API. Beberapa aplikasi pendukung telah diinstal pada penelitian ini seperti Text Editor (VS. Code), aplikasi XAMPP, Postman dan Browser. Selanjutnya dibangun aplikasi berbasis webservice membuat dengan menggunakan DBMS MySQL. Pada kasus ini, table yang dibaca dari database sistem informasi dealer panca niaga mandiri yaitu table penerimaan kas. Selanjutnya proses pembuatan API menggunakan kaidah Restful API.

File **koneksi.php** ini bertujuan untuk menghubungkan antara database dengan aplikasi yang akan dibuat. Sedangkan fungsi sanitasi mencegah sql injection bertujuan untuk memberikan Batasan karakter yang valid baik melalui metode POST ataupun GET.

API untuk retrieve data mesin

Setelah membuat file **koneksi.php**, selanjutnya pada bagian pembahasan API (*Application Programming Interface*) khususnya untuk mencari data mesin. Pengujian API ini dilakukan menggunakan Aplikasi POSTMAN dapat dilihat pada Gambar 3. Hasil pengujian melalui POSTMAN menampilkan data dalam bentuk JSON

```
{
  "kode": 1,
  "pesan": "Data Tersedia",
  "data": [
```

```
{
  "id_unit": "2",
  "smh_unit": "10",
  "kode_smh": "HJ9 (SUPRA GTR
150 Sporty)",
  "nama_smh": "G2E02R21M1 M/
T"
}
]
```

Gambar. Pengujian API Mesin menggunakan POSTMAN

Melalui data JSON yang dihasilkan akan dijadikan sebagai sumber agar dapat diakses melalui aplikasi lain seperti perangkat mobile. Dengan menerapkan file JSON tersebut divisi lain dapat memanggil data tersebut sehingga permintaan data secara file sharing dapat digantikan melalui API. Pembuatan API tersebut merupakan susunan dari source code berikut:

```
<?php
require_once('../Connections/koneksi.php');
```

```
if (isset($_POST['key']) &&
($_POST['key'] == "secretkey")) { // 2

$viewSQL = sprintf("SELECT id_unit,
smh_unit, kode_smh, nama_smh, cc_smh,
warna_unit, mesin_unit, rangka_unit,
tahun_unit, statusmilik_unit, status_unit
FROM tb_unit
INNER JOIN tb_smh ON id_smh =
smh_unit
WHERE active_unit = 'Y' and
statusmilik_unit = 'S'
AND mesin_unit = %s OR kode_smh
LIKE %s ORDER BY id_unit DESC",
```

```
GetSQLValueString($koneksi,
$_POST['mesin_unit'], "text"),
GetSQLValueString($koneksi,
$_POST['kode_smh'], "text"));
$rs_view = mysqli_query($koneksi,
$viewSQL) or
die(mysqli_error($koneksi));
$total_rs_view =
mysqli_num_rows($rs_view);

if ($total_rs_view > 0) {
    $response['kode'] = 1;
    $response['pesan'] = "Data
Tersedia";
    $response['data'] = array();
    foreach ($rs_view as $data) {
        $Data['id_unit'] = $data['id_unit'];
        $Data['smh_unit'] =
$data['smh_unit'];
        $Data['kode_smh'] =
$data['kode_smh'];
        $Data['nama_smh'] =
$data['nama_smh'];
        array_push($response['data'],
$Data);
    }
} else {
    $response['kode'] = 0;
    $response['pesan'] = "Data Tidak
Tersedia";
}

echo json_encode($response);
mysqli_close($koneksi);

}
```

SIMPULAN

RESTful API menggunakan protokol HTTP sebagai basis komunikasinya dan mengikuti prinsip-prinsip dari arsitektur REST (Representational State Transfer). Penggunaan RESTful API memiliki beberapa keuntungan, yaitu dari Kemudahan integrasi, dimana RESTful API memungkinkan aplikasi untuk terintegrasi dengan sistem lain dengan mudah, karena menggunakan protokol HTTP yang umum digunakan. Sedangkan

keuntungan lainnya dari sisi Skalabilitas. RESTful API memungkinkan peningkatan kapasitas sistem dengan menambah server baru tanpa mengganggu operasional sistem yang sedang berjalan. Selain itu, RESTful API menyediakan dokumentasi yang baik dan terbuka, sehingga memudahkan pengembang lain untuk memahami dan menggunakannya. Namun, ada juga beberapa kekurangan dari penggunaan RESTful API yang tidak menyediakan keamanan yang kuat, sehingga perlu dilakukan tambahan pengamanan seperti enkripsi atau autentikasi dan dapat mengalami penurunan kinerja jika terjadi peningkatan permintaan yang tinggi. Kesimpulannya, RESTful API merupakan salah satu solusi yang efektif untuk mengembangkan aplikasi web yang membutuhkan integrasi dengan sistem lain. Namun, perlu diperhatikan kekurangan yang mungkin terjadi seperti keamanan dan kinerja yang rendah dalam kondisi permintaan yang tinggi.

DAFTAR PUSTAKA

- Newman, S. (2015). *Building Microservices: Designing Fine-Grained Systems*. United States: O'Reilly Media.
- Iqbal, M., Nurwati., Syahputra, A.K., Akmal, N., & Yori, A (2022). Sistem Terintegrasi Menggunakan API: Implementasi Website dan Android. (n.p.): Nas Media Pustaka.
- Arimbawa, I. W. A., Afwani, R., Studi, P., Informatika, T., Teknik, F., Mataram, U., & Majapahit, J. (2019). 88-1-398-1-10-20190720. 2(1), 38–45.
- Baharuddin, B., Wakkang, H., & Irianto, B. (2022). Implementasi Web Service Dengan Metode Rest Api Untuk Integrasi Data Covid 19 Di Sulawesi Selatan. *Jurnal Sintaks Logika*, 2(1), 236–241.
- Dharma Handayani, S., & Uminingsih, U. (2020). Pengorganisasian Kerja Sistem Parkir Menggunakan

-
- Arsitektur Microservice. *Jurnal Teknologi*, 13(1), 27–35. <https://ejournal.akprind.ac.id/index.php/jurtek/article/view/2891>
- Utomo, S. P., Alfiyah, N. H., Sani, Z. A., Hanafi, M., & Primadewi, A. (2020). Informasi Terintegrasi Menggunakan FrameWork CodeIgniter. *Seminar Nasional Dinamika Informatika*, 124–128.
- Arimbawa, I. W. A., Afwani, R., Studi, P., Informatika, T., Teknik, F., Mataram, U., & Majapahit, J. (2019). 88-1-398-1-10-20190720. 2(1), 38–45.
- Baharuddin, B., Wakkang, H., & Irianto, B. (2022). Implementasi Web Service Dengan Metode Rest Api Untuk Integrasi Data Covid 19 Di Sulawesi Selatan. *Jurnal Sintaks Logika*, 2(1), 236–241.
- Dharma Handayani, S., & Uminingsih, U. (2020). Pengorganisasian Kerja Sistem Parkir Menggunakan Arsitektur Microservice. *Jurnal Teknologi*, 13(1), 27–35. <https://ejournal.akprind.ac.id/index.php/jurtek/article/view/2891>
- Iqbal, M., Syahputra, A. K., & Handoko, W. (2022). *PENERAPAN MONOLITIC ARSITECTURE PADA APLIKASI UJIAN*, 213–218.
- Utomo, S. P., Alfiyah, N. H., Sani, Z. A., Hanafi, M., & Primadewi, A. (2020). Informasi Terintegrasi Menggunakan FrameWork CodeIgniter. *Seminar Nasional Dinamika Informatika*, 124–128.