

BAB I. PENDAHULUAN

1.1 Latar Belakang

PT Enigma Cipta Humanika adalah perusahaan *bootcamp IT* dan manajemen talenta yang didirikan pada tahun 2017. Awalnya perusahaan ini merupakan anak perusahaan dari PT. Square Techno Indonesia (STI) didirikan pada tahun 2019 dan berkembang menjadi perusahaan baru bernama PT. Enigma Cipta Humanika. PT Enigma Cipta Humanika juga fokus pada bisnis *IT Bootcamp* dan *IT Talent Management*. PT Enigma Cipta Humanika memiliki visi untuk memberikan dampak positif pada dunia pendidikan khususnya *IT* melalui tiga program utama: *Training as a Service, Skilled Recruitment dan Experience Development Software* yang mendukung bisnis lain melalui Camp Enigma. Enigma Camp juga memiliki visi untuk menjembatani kesenjangan antara lulusan lembaga pendidikan dan kebutuhan industri. (Purwandari, 2022).

PT Enigma Cipta Humanika juga terlibat dalam industri *smartsourcing*, sebuah industri yang tumbuh dari *outsourcing*. Industri ini memiliki organisasi-organisasi yang tersebar di seluruh perusahaan yang memberikan pelatihan kepada calon-calon potensial dan menjadikan mereka karyawan dalam jangka waktu tertentu. PT Enigma Cipta Humanika menempatkan karyawan di perusahaan teknologi terkemuka. (Yudhistira, 2022).

PT. Enigma Cipta Humanika belum menggunakan sistem terutama pada sistem penilaian untuk menilai *trainee bootcamp* tersebut untuk memperoleh hasil nilai keseluruhan kriteria *trainee*. Untuk saat ini *trainer* mencatat nilai hasil dari para *trainee bootcamp* di *spreadsheet*. Selanjutnya, catatan hasil nilai tersebut diserahkan kepada *admin training* dan dibuatkan summary untuk di masukkan ke rapor. Terakhir di validasi oleh spesialis admin training untuk mendapatkan sertifikat. PT. Enigma Cipta Humanika belum memiliki aplikasi dan *web service* untuk keperluan untuk penilaian tersebut sehingga sering terjadi data nilai tidak sengaja terhapus oleh *trainer* di dalam data *spreadsheet* nya dan juga nilai para *trainee bootcamp*. Karena proses tersebut masih berbasis tenaga kerja. *frontend* dari aplikasi ini belum memiliki *API* khusus untuk mengelola data-data tersebut.

Berdasarkan masalah tersebut, di perlukan Rancangan *web service API* data Penilaian *Trainee* sistem penilaian akademik di PT. Enigma Cipta Humanika untuk mempermudah *frontend* dalam membangun aplikasi yang memiliki tampilan *web* dan memiliki fitur lengkap untuk keperluan penilaian pada peserta pelatihan. Soulisinya adalah dengan membuat “Rancangan *Web Service Backend* Untuk Penilaian *Trainee* Pada *Training Assessment Application* Berbasis *Website* di PT. Enigma Cipta Humanika”.

Backend development adalah proses desain sistem yang terjadi di belakang situs *web* atau aplikasi. *TypeScript, Javascript, PHP, Ruby, Python*, dll adalah bahasa pemrograman untuk pengembangan backend. Semua sistem ini bekerja di belakang layar dan tentu saja tidak terlihat langsung oleh pengguna, namun ketika merancang sebuah web atau aplikasi, posisi pengembangan backend mendukung tugas-tugas yang terjadi, terutama ketika mengelola data akan ditampilkan . (Hasanuddin et al., 2022). Desain pengembangan backend berfokus pada *database, skrip*, dan struktur situs *web* atau aplikasi. *Backend* sendiri dibuat menggunakan kode pemrograman yang berguna untuk mengelola data dan informasi yang akan disediakan untuk pengembangan frontend.

1.2 Tujuan

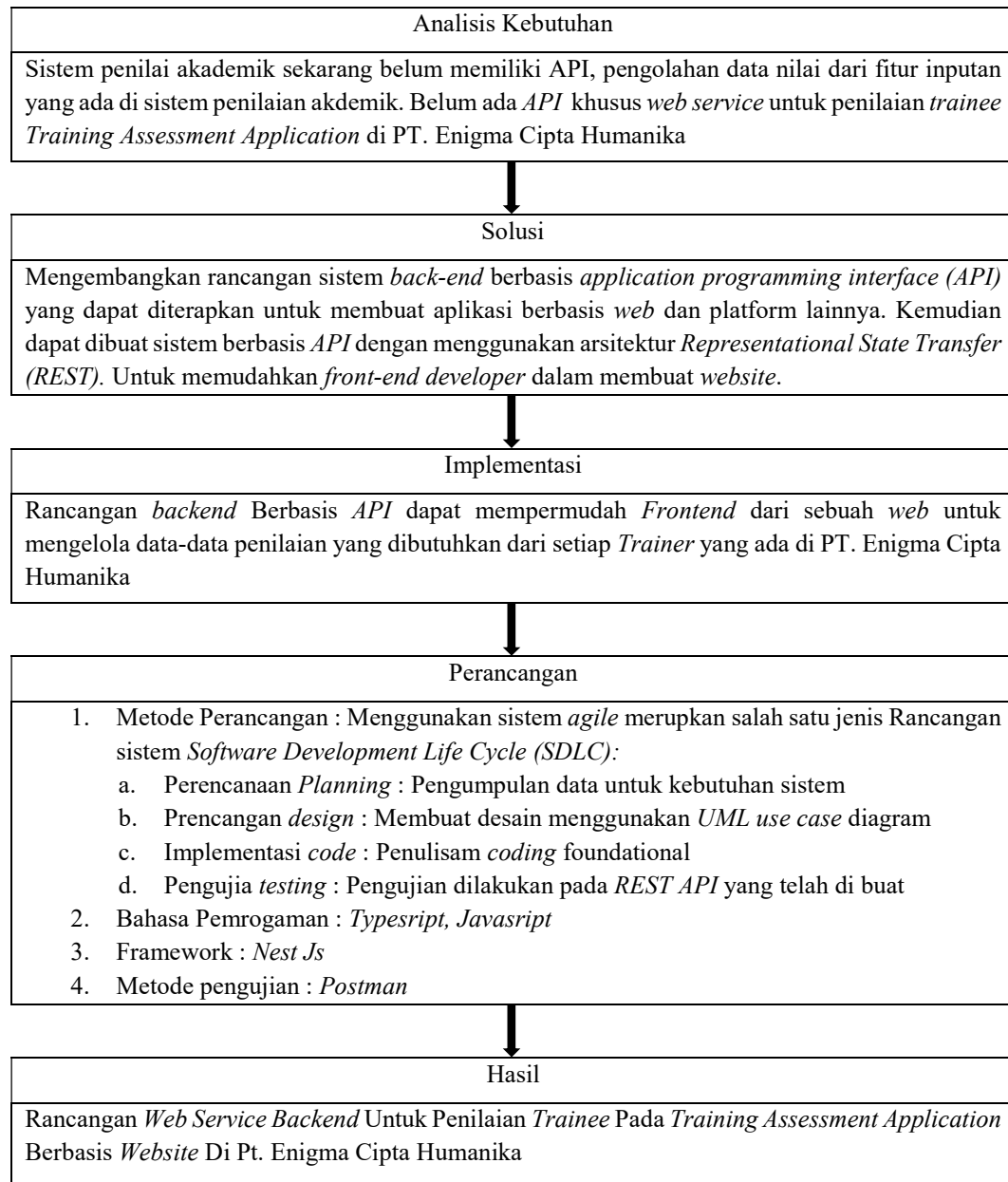
Tujuan dari pembuatan tugas akhir ini adalah mempermudah *frontend* mengola data-data penilaian *trainee* dan merancang *Web API* Untuk Penilaian *Trainee* Pada *Training Assessment Application* Berbasis *Website* di Pt. Enigma Cipta Humanika

1.3 Kerangka Pemikiran

Kegiatan penilaian yang di lakukan oleh PT. Enigma Cipta Humanika yaitu para *trainer* mencatatat nilai hasil dari para *trainee bootcamp* di *spreadsheet*. Selanjtutnya, catatan hasil nilai tersebut diserahkan kepada *admin* training dan dibuatkan *sumary* untuk di masukkan ke rapor. Terakhir di valiadasi oleh sepsialis admin training untuk mendapatkan sertifikat. Oleh karena itu PT. Enigma Cipta Humanika belum memiliki aplikasi dan *web service* untuk keperluan untuk penilain tersebut sehingga sering terjadi data nilai tidak sengaja terhapus oleh *trainer* di dalam data *spreadsheet* nya dan juga nilai para *trainee bootcamp*.

Berdasarkan masalah tersebut, di perlukan Rancangan *Web Service API* Untuk Penilaian *Trainee* untuk mempermudah *frontend* dalam membangun aplikasi yang memiliki tampilan *web* dan memiliki fitur lengkap untuk keperluan penilaian pada *trainee bootcamp*.

Bagan kerangka pemikiran disajikan pada Gambar 1.



Gambar 1. Kerangka Pemikiran

1.4 Kontribusi

Dengan dibuatnya Rancangan *Web Service Backend Training Assessment Application* Berbasis *Website* Pada PT. Enigma Cipta Humanika. Diharapkan dapat berkontribusi sebagai berikut :

1. PT. Enigma Camp Cipta Humanika
 - a. Mempermudah mengontrol kegiatan setiap pelatihan Yang diadakan di PT. Enigma Camp Cipta Humanika
 - b. Memberikan aplikasi sistem penilaian kepada PT. Enigma Camp Cipta Humanika
2. Admin dan *Trainer* PT. Enigma Cipta Humanika
 - a. Mempermudah mengelola data penilaian pada *Training Assessment Application*
 - b. Mempermudah mengelola aplikasi penilaian akademik
 - c. Mempermudah memberikan informasi nilai kepada peserta pelatihan bootcamp
3. Peserta *Talent*
 - a. Mempermudah menerima informasi hasil penilaian dan kriteria penilaian
4. *Frontend Developer*
 - a. Membantu Frontend dalam pengisian nilai pada trainee pada Aplikasi Sistem Penilaian Akademik
 - b. Mempermudah dalam mengelola data nilai pada aplikasi yang di bangun frontend

BAB II. TINJAUAN PUSTAKA

2.1 *Back end*

Back end atau sering disebut sisi *server* pada dasarnya adalah sebuah tempat dimana proses dari sebuah aplikasi atau sistem berjalan di ujung belakang ini data diproses ditambahkan, dimodifikasi atau dihapus dan semua itu biasanya tidak terlihat atau berinteraksi dengan langsung ke pengguna, seperti database dan server (Arhandi, 2018). *Back end developer* adalah *programmer* atau pengembang yang pekerjaannya berfokus pada keamanan, desain sistem, dan manajemen data pada sistem. Pengembang *back end* sistem aplikasi dinamis sangat dibutuhkan dalam pengembangan sistem yang memiliki data yang selalu berubah.

2.2 *Perancangan*

Perancangan bertujuan untuk merancang suatu sistem baru yang dapat menyelesaikan permasalahan yang dihadapi suatu perusahaan, dan merupakan kegiatan yang dilakukan dengan memilih alternatif sistem yang terbaik. (Setiawan, 2018).

2.3 *Perancangan Sistem*

Perancangan sistem merupakan serangkaian kegiatan yang menggambarkan secara detail cara kerja sistem. Hal ini bertujuan untuk menghasilkan produk perangkat lunak yang sesuai dengan kebutuhan pengguna. (Fahlendra, 2022).

2.4 *Database*

Basis data atau *database* merupakan kumpulan informasi yang disimpan secara sistematis pada suatu komputer sehingga informasi dalam basis data tersebut dapat diperoleh dengan menggunakan program komputer. (Andaru, 2018.).

Tujuan utama dari sistem basis data adalah untuk memungkinkan pengguna membangun tampilan abstrak dari data bertujuan untuk menyederhanakan interaksi antara pengguna dan sistem, dan basis data dapat memberikan pandangan yang berbeda kepada pengguna, pemrogram, dan administrator.

2.5 *Application Programming Interface (API)*

API adalah antarmuka yang digunakan program untuk mengakses aplikasi atau layanan. *API* memungkinkan pengembang untuk menggunakan fungsionalitas yang ada dari aplikasi lain sehingga mereka tidak harus memulai dari awal.. Pada

konteks *website*, *API* merupakan pemanggilan fungsi melalui *Hyper Text Transfer Protocol (HTTP)* dan dapat menerima respons yang dapat diperluas. *XML (Markup Language)* atau objek *JavaScript (JSON)* (Hasanuddin et al., 2022).

API secara fungsional dapat memfasilitasi pengembangan *web* atau aplikasi. *API* juga dapat dengan mudah memperluas pengembangan menggunakan *API* dan data dari *server*. Oleh karena itu, dikembangkan untuk tujuan akselerasi transfer data dan buat situs *web* atau aplikasi yang lebih fungsional.

2.6 RESTful API

REST (REpresentational State Transfer) adalah arsitektur komunikasi berbasis standar *web* yang sering diterapkan pada pengembangan layanan berbasis *web*. *HTTP (Hypertext Transfer Protocol)* umumnya digunakan sebagai protokol komunikasi data. *REST* pertama kali diperkenalkan oleh Roy Fielding pada tahun 2000. Server *REST* dalam arsitektur *REST* menyediakan sumber daya (*resources/data*) dan klien *REST* mengakses dan menampilkan sumber daya untuk digunakan. Setiap sumber daya diidentifikasi oleh Universal Resource Identifier (*URI*) atau *ID global*. Sumber daya ini tersedia dalam format teks (*JSON* atau *XML*). Format umum adalah *JSON* dan *XML*. (Syarifudin et al., 2018).

REST API memungkinkan mengirim dan menerima data dari berbagai sistem dengan cara yang sangat sederhana. Setiap panggilan *REST API* memiliki hubungan antara kata kerja *HTTP* dan *URL*. Dapat menggunakan titik akhir *API REST* untuk memetakan sumber daya database aplikasi (Choirudin & Adil, 2019).

2.7 Framework NestJS

NestJS adalah *Node* progresif. *Javascript framework* yang mendukung pembuatan aplikasi sisi *server*. *Nest* memperluas *Node.js Framework* seperti *Express* atau *Fastify* menambahkan organisasi modular dan berbagai perpustakaan lain untuk menangani tugas yang berulang. *Software* ini *opensource*, menggunakan *TypeScript*, dan sangat serbaguna dalam pengembangan aplikasi (Muhamad Saepuloh & Ginting, 2022).

2.8 NodeJs

NodeJS adalah perangkat lunak yang dirancang untuk pengembangan aplikasi berbasis *web* dan ditulis dalam sintaks bahasa pemrograman *Javascript*. Meskipun *Javascript* adalah bahasa pemrograman yang hanya berjalan di sisi

klien/browser, *NodeJS* melengkapi peran *JavaScript* dan juga dapat Ia bekerja dengan bahasa pemrograman yang berjalan di sisi server, seperti *PHP*, *Ruby*, *Perl*, dll. *NodeJS* dapat berjalan di sistem operasi *Windows*, *Mac OS*, dan *Linux* tanpa perubahan kode apa pun. *NodeJS* memiliki perpustakaan *server HTTP* sendiri, memungkinkan Anda menjalankan server *web* tanpa menggunakan program *server web* seperti *Nginx* dan *Apache*. (Hasanuddin et al., 2022).

2.9 JWT (JSON Web Token)

JWT ini adalah *token* dalam bentuk *string* informasi *string JSON* yang sangat padat (ukurannya) yang digunakan untuk otentikasi dan pertukaran informasi.. Karena ukurannya yang kecil, token *JWT* dapat dikirim dalam *URL*, parameter *HTTP POST*, atau *header HTTP*, dan ukurannya yang kecil memungkinkan transfer yang lebih cepat. Konten token yang dihasilkan disebut informasi independen karena berisi informasi pengguna yang diperlukan, sehingga tidak perlu menanyakan database lebih dari satu kali. Token tersebut dapat diverifikasi dan dipercaya karena sudah di-*sign* secara digital (Rahmatulloh et al., 2018).

2.10 Web Service

Web service adalah teknologi yang mengubah fungsi *Internet* dengan menambahkan fungsionalitas *Web* transaksional, yaitu kemampuan *Web* untuk berkomunikasi satu sama lain dalam pola program-ke-program (P2P). Sampai saat ini, fokus web didominasi oleh komunikasi program-to-program melalui interaksi business-to-consumer (B2C), dan web transaksional akan didominasi oleh program-to-program dengan interaksi business-to-business (Setiyawati et al., 2016). Layanan *web* adalah teknologi alternatif untuk sistem terdistribusi yang beroperasi pada protokol *HTTP* dan bertukar pesan melalui jaringan menggunakan standar *SOAP*. Pesan *SOAP* dapat berisi informasi sensitif yang harus dilindungi ketika mengimplementasikan layanan *web* di lingkungan heterogen seperti *Internet*. (Eko Win Kenali, 2010).

Web service terdiri dari interaksi antara 3 komponen yang berperan dalam proses pertukaran data yaitu: *service provider*, *service registry*, *service requestor* (Pratomo & Haryono, 2020). Berikut adalah penjelasan kapan komponen-komponen tersebut:

1. *Service Provider* (penyedia layanan)
Berfungsi sebagai pemilik sekaligus *platform* yang mengakomodir semua akses layanan
2. *Service Requestor* (peminta layanan)
Berfungsi sebagai permintaan atau memulai interaksi dengan layanan. Peran peminta layanan ini dapat dijalankan oleh program
3. *Service Registry*
Berfungsi sebagai penyedia daftar deskripsi layanan tempat penyedia layanan menerbitkan deskripsi layanan mereka

2.11 Arsitektur Web Service

Web service didefinisikan mekanisme komunikasi antara sistem komputer yang berbeda. Tanpa layanan *web*, komunikasi *peer-to-peer* khusus menjadi rumit dan bergantung pada platform yang dipahami dan ditafsirkan oleh *web* seperti ratusan hal yang berbeda. Ini sangat berguna ketika sistem komputer sesuai dengan protokol yang dapat dipahami dengan mudah oleh *web*. (Choirudin & Adil, 2019). Tiga komponen standar *Web Service* adalah sebagai berikut :

1. *Simple Object Access Protocol (SOAP)*, sebuah protokol yang menangani pertukaran informasi dalam jaringan terdistribusi..
2. *Web Service Definition Language (WSDL)*, Menentukan keberadaan layanan web di jaringan serta dokumen standar yang ditulis dalam format *XML*.
3. *Universal Description, Discovery, and Integration (UDDI)*, dibangun di atas *XML* dan dapat diakses oleh entitas dalam jaringan dan juga memaparkan lokasi direktori tanpa platform yang berisi layanan..

2.12 Unified Modelling Language (UML)

Unified Modeling Language (UML) adalah pemodelan visual yang menggambarkan, mengilustrasikan, membangun dan mendokumentasikan pengembangan sistem informasi yang memiliki paradigma berorientasi objek. Dalam *UML* terdapat beberapa diagram yang dapat digunakan untuk memudahkan setiap pihak dalam memodelkan sistem yang akan dikembangkan (Waruwu & Nasution, 2018).

2.13 *Web API*

Web API adalah antarmuka program sistem yang dapat diakses melalui metode dan header protokol *HTTP* standar. *API Web* dapat diakses dari berbagai klien *HTTP*, seperti *browser* dan perangkat seluler. *Web API* juga memiliki keuntungan menggunakan infrastruktur yang sama dengan yang digunakan *web*, terutama untuk *caching* dan *concurrency* (Kurniawan, 2018.).

2.14 *XAMPP*

XAMPP yaitu *APACHE*, *PHP*, *MySQL* dan *PHPMyAdmin*. *XAMPP* adalah alat untuk mengintegrasikan paket perangkat lunak ke dalam satu paket yang menginstal *XAMPP*, menghilangkan kebutuhan untuk menginstal dan mengkonfigurasi *server web APACHE*, *PHP* dan *MySQL* secara manual. *XAMPP* menginstal dan mengkonfigurasi sendiri, atau mengkonfigurasi sendiri untuk Anda. (Prasetyo et al., 2023).

2.15 *MySQL*

MySQL adalah sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara bebas di bawah *GPL (General Public License)*. Setiap orang bebas menggunakan *MySQL*, tetapi bukan sebagai sumber tertutup atau turunan komersial (Arhandi, 2016). *MySQL* Ini sebenarnya berasal dari salah satu konsep utama basis data sejak lama, *SQL (Structured Query Language)* (Prasetyo et al., 2023).

2.16 *Metode Pengembangan Sistem*

Metode Pengembangan Sistem *Agile* yaitu salah satu jenis pengembangan sistem *Software Development Life Cycle (SDLC)*. *SDLC* adalah sebuah metode atau proses pengembangan sistem agar pekerjaan berjalan dengan terstruktur, efektif dan mencapai tujuan dengan sempurna.

Metode *Agile* adalah metodologi pengembangan tambahan yang berfokus pada pengembangan cepat, rilis perangkat lunak tambahan, pengurangan biaya, produksi kode berkualitas tinggi, dan proses pengembangan berbasis pelanggan. (Mahendra & Yanto, 2018).

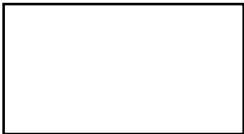
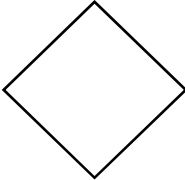
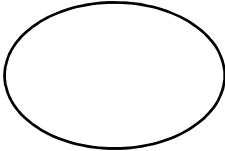

2.17 *Entity Relationship Diagram (ERD)*

Entity Relationship Diagram atau *ERD* adalah diagram struktural yang digunakan untuk merancang basis data. *ERD* menggambarkan data yang disimpan

dalam sistem dan keterbatasannya. *ERD* adalah himpunan entitas, himpunan relasi, dan batasan. Selain itu juga dapat menggambarkan hubungan-hubungan yang ada dalam pengolahan data, seperti hubungan *many to many*, *one to many*, *one to one* (Latukolan et al., 2018.).

Simbol-simbol ERD ditunjukkan pada Tabel 1.

Tabel 1. Simbol-simbol ERD

Simbol	Nama	Keterangan
(1)	(2)	(3)
	Entitas	Kumpulan suatu object diidentifikasi secara unik
	Relasi	Menunjukkan adanya hubungan di antara sejumlah entitas relasi yang berbeda
	Atribut	Properti unit atau hubungan itu penjelasan rinci tentang unit tersebut
	Garis Relasi	Sebagai pemhubung antara relasi dengan entitas relasi dan entitas dengan atribut

Sumber: (Solikhin et al., 2018)

2.18 Postman

Postman adalah perangkat lunak yang memiliki fungsionalitas pengembangan sistem penuh dalam mengirim dan menerima *respons server*. Perangkat lunak ini mendukung pengembangan sistem *REST API* dengan mengkategorikan permintaan berdasarkan metode permintaan, *URL* dan parameter permintaan (Pamuji, n.d., 2020). *Software* ini banyak digunakan oleh penguji dan pengembang aplikasi.

2.19 Artikel Ilmiah Terkait

Berikut ini beberapa jurnal yang menjadi bahan referensi penulis menyusun karya tulisnya :

1. Hasanuddin et al, (2022). Dengan judul “*Rancang Bangun Rest Api Aplikasi Weshare Sebagai Upaya Mempermudah Pelayanan Donasi Kemanusiaan*”. Karya ilmiah ini di tulis untuk aplikasi *WeShare* terdapat beberapa fungsi metode *respon request*, terdapat *endpoint* sebagai alamat *API* untuk memudahkan *Hitting API* bagi tim pengembang khususnya yang bekerja sebagai *Front End Team* dan terdapat penjelasan sebagai panduan yang digunakan oleh tim untuk menggunakan setiap *API*.
2. Pratomo & Haryono, (2020). Dengan judul “*Perancangan RESTful Web Service Satuan Kredit Partisipasi di Universitas Islam Indonesia*”. Karya ilmiah ini di tulis untuk juga dapat digunakan dalam kebanyakan kasus protokol *HTTP/HTTPS* digunakan dan data yang dipertukarkan dapat berupa *XML*, *JSON*, atau beberapa tipe data lainnya. Hal ini karena *REST* adalah arsitektur untuk mengembangkan layanan *web* sedangkan *SOAP* adalah protokol untuk mengembangkan layanan *web*.
3. Kurniawan, (2018). Dengan judul “*Implementasi Rest Web Service Untuk Sales Order Dan Sales Tracking Berbasis Mobile*”. Karya ilmiah ini di tulis untuk Penerapan *REST services* dengan format *JSON* sebagai backend pada sistem ini sangat cocok karena dengan format dokumen *JSON* yang kecil, proses pengunduhan data dari *web services* lebih cepat dibandingkan menggunakan dokumen *XML* yang ukurannya relatif besar.