

BAB II

LANDASAN TEORI

2.1 Tinjauan Studi

Penelitian yang dilakukan berkaitan dengan milik Dedofin dan Bagus Mulyawan membuat sistem penjualan untuk menggantikan proses pencatatan penjualan yang masih manual dengan mengimplementasikan *Restful API* sebagai *web service* dan metode Regresi Linear untuk peramalan stok barang bulan berikutnya [5].

Pada penelitian yang berjudul “RESTFul Web Service Untuk Sistem Pencatatan Transaksi Studi Kasus PT. XYZ” yang dilakukan oleh Penidas Fiodinggo Tanaem, Danny Manongga dan Ade Iriani. Menghasilkan sistem pencatatan transaksi yang saling terintegrasi dengan *platform* dan *database* yang berbeda pada PT. XYZ menggunakan *Restful API* [6]. Hasil dari penelitian yang didapat bahwa Restful API dapat menghubungkan *platform* yang berbeda dan *database* yang berbeda pula.

Sedangkan penelitian milik Alam Rahmatulloh, Heni Sulastri dan Rizal Nugroho yang berjudul “Keamanan RESTful Web Service Menggunakan JSON Web Token (JWT) HMAC SHA-512” yang membandingkan algoritma SHA-512 dengan algoritma SHA-256 dalam pembuatan token JWT (JSON Web Token). Hasilnya penggunaan algoritma SHA-512 lebih cepat 1% dibandingkan dengan algoritma SHA-256 dan menghasilkan nilai hash 2% lebih besar dibandingkan algoritma SHA-256, menjadikan keamanan pertukaran data lebih baik karena token lebih panjang.

Tabel 2.1: State of The Art

No	Nama Penulis	Tahun	Judul	Masalah	Metode	Hasil
1	Dedofin dan Bagus Mulyawan.	2018	Perancangan Sistem Penjualan Barang Dan Jasa Untuk Toko Intikarya Aluminium	Proses pencatatan penjualan dari perusahaan tersebut menggunakan Kertas, Microsoft Word dan Excel.	Menggunakan REST API dan mengimplementasikan metode Regresi Linear.	Sistem Penjualan Barang dan Jasa yang mengimplementasikan REST API dan metode Regresi Linear untuk peramalan stok bulan berikutnya.
2	Penidas Fiodinggo Tanaem, Danny Manongga dan Ade Iriani.	2016	RESTFul Web Service Untuk Sistem Pencatatan Transaksi Studi Kasus PT. XYZ	Pada PT. XYZ memiliki banyak sistem dan <i>platform</i> yang berbeda	Penerapan Restful Web Service dan JWT (JSON Web Token) sebagai akses keamanan.	Penggunaan Restful dapat mengintegrasikan data pada sistem dan <i>platform</i>

				sehingga memberikan dampak pada data gudang dan pencatatan transaksi, karena <i>database</i> yang digunakan tiap aplikasi berbeda.		yang berbeda pada PT. XYZ yang dijalankan pada jaringan local.
3	Alam Rahmatulloh, Heni Sulastri dan Rizal Nugroho.	2018	Keamanan RESTful Web Service Menggunakan JSON Web Token (JWT) HMAC SHA-512	Perbandingan kinerja algoritma SHA-512 dan SHA-256 dalam proses pembuatan token.	Algoritma SHA-512 dan SHA-256	Penerapan algoritma SHA-512 lebih cepat 1% dengan algoritma SHA-256 dan menghasilkan nilai <i>hash</i> 2% lebih besar dari SHA-256

						sehingga token lebih panjang.
--	--	--	--	--	--	-------------------------------

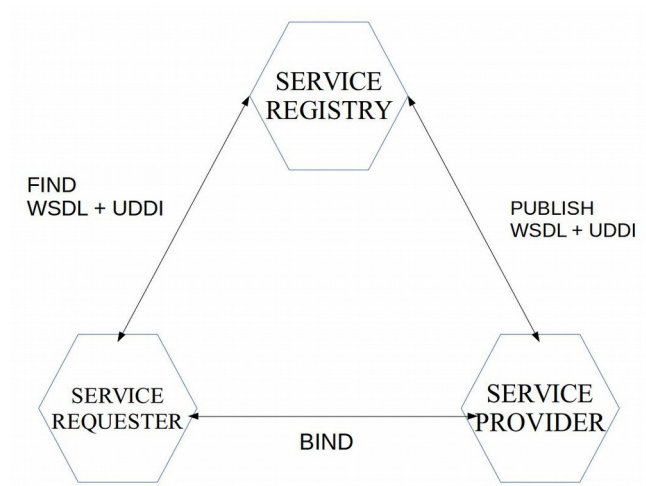
Kesimpulan:

Penggunaan REST API pada *Web service* telah banyak diterapkan pada pembuatan sistem, karena kemampuannya yang dapat mengintegrasikan data pada *platform* yang berbeda dengan menggunakan protokol HTTP. Dalam keamanan integrasi data menggunakan JWT (JSON Web Token) sebagai otentikasi dalam mengakses data pada *server*.

2.2 Tinjauan Pustaka

2.2.1 Web Service

Web Service adalah sebuah entitas komputasi yang dapat diakses melalui jaringan internet maupun intranet dengan standar protokol tertentu dalam platform dan antarmuka bahasa pemrograman yang independen [2]. Web Service mampu menjembatani komunikasi antar program karena tidak dipengaruhi oleh platform, arsitektur, maupun bahasa pemrograman. Sehingga setiap platform yang berada pada satu jaringan sama atau pada jaringan yang berbeda dapat saling berkomunikasi dengan menggunakan protokol yang ditetapkan oleh web service seperti HTTP.



Gambar 2.1: Arsitektur Web Service

Peran service pada arsitektur:

1. Service Provider

Service provider adalah penyedia layanan web. Penyedia layanan mengimplementasikan layanan dan membuatnya tersedia di Internet.

2. Service Requester

Ini adalah konsumen dari web Service. Peminta layanan yang mencari dan menemukan layanan yang dibutuhkan serta menggunakan layanan tersebut.

3. Service Registry

Berfungsi sebagai lokasi central yang mendeskripsikan semua layanan/service yang telah di-register.

2.2.2 REST

Konsep REST pertama kali diperkenalkan oleh Roy Fielding pada tahun 2000. REST merupakan standar arsitektur komunikasi yang digunakan pada pengembangan layanan berbasis web. Sistem yang menggunakan prinsip-prinsip dari REST dapat disebut dengan “RESTful”. Cara kerja RESTful dimulai dari client melakukan *request* melalui *HTTP Request*, kemudian server merespon melalui *HTTP Response* [7]. Komponen dari *HTTP Request* adalah sebagai berikut:

- **Verb. HTTP Method**, yang digunakan di antaranya GET, PUT, POST, DELETE.
- **Uniform Resource Identifier (URI)**, untuk mengidentifikasi lokasi *resource* pada *server*.
- **HTTP Version**, menjelaskan versi dari HTTP yang akan digunakan, contohnya HTTP v1.1.
- **Request Header**, berisi metadata untuk HTTP Request. Contohnya adalah tipe client/browser, format yang didukung oleh client, format dari body pesan, dan setting cache.
- **Request Body**, konten data yang dikirimkan.

Sedangkan komponen pada HTTP Response berisikan:

- **Status/Response Code**, menjelaskan status *server* pada resource yang di *request*.
- **HTTP Version**

- **Response Header**, berisi metadata untuk HTTP Response. Contohnya tipe *server*, tipe content, panjang content, dan waktu *response*.
- **Response Body**, berisi konten data dari hasil *request*.

2.2.3 JSON

JSON (JavaScript Object Notation) merupakan suatu format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan oleh komputer. Sebelum adanya JSON, XML biasa digunakan untuk saling bertukar data. Format penulisan JSON tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh bahasa pemrograman keluarga C seperti C, C++, C#, Perl, Java, JavaScript, Python dan sebagainya. Oleh karena sifat – sifat tersebut, hal ini menjadikan JSON ideal sebagai bahasa pertukaran data [5].

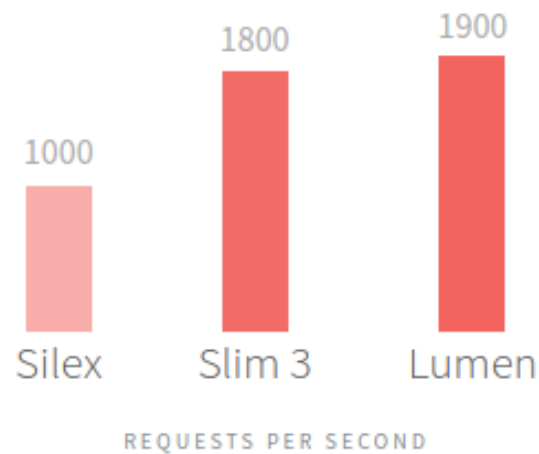
2.2.4 Laravel

Laravel dikembangkan oleh Taylor Otwell, dirilis pada tanggal 5 Juni 2011 dan masih terus berkembang sampai sekarang. Laravel merupakan web application framework berbasis PHP yang open source dan menggunakan konsep MVC (Model, View, Controller) [8]. MVC memisahkan aplikasi berdasarkan komponen – komponen aplikasi, seperti manipulasi data, controller dan user interface.

- Model merupakan bagian yang berfungsi untuk mengakses database.
- View merupakan desain antarmuka yang langsung berinteraksi dengan pengguna.
- Controller merupakan bagian yang memproses permintaan dari pengguna.

2.2.5 Lumen

Lumen merupakan *micro-framework* yang dibuat dari Laravel. Lumen merupakan versi kecilnya Laravel yang difokuskan pada pengembangan web dengan menggunakan konsep Rest API. Terdapat beberapa micro-framework yang dapat digunakan untuk membuat Rest API pada PHP seperti Slim Framework, Sillex, dan lain – lain. Perbedaan Lumen dari micro-framework lainnya adalah kemampuannya dalam menangani Request hingga 1900 Request/detik.



Gambar 2.2: Benchmark Request Lumen

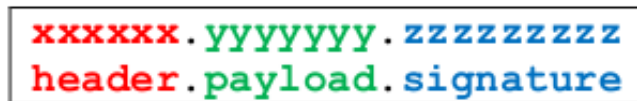
2.2.6 API

Application Programming Interface atau disingkat API yang memungkinkan developer untuk mengintegrasikan dua bagian dari aplikasi atau dengan aplikasi yang berbeda secara bersamaan. API terdiri dari berbagai elemen seperti fungsi, protokol dan tools lainnya yang memungkinkan pengembang untuk membuat aplikasi. Tujuan penggunaan API yaitu untuk mempercepat proses development dengan menyediakan function secara terpisah sehingga pengembang

tidak perlu membuat fitur yang serupa [2]. Pada pengembangan web integrasi data menggunakan format penulisan yang sering digunakan seperti XML atau JSON, pada penelitian ini menggunakan format JSON sebagai format integrasi data yang digunakan.

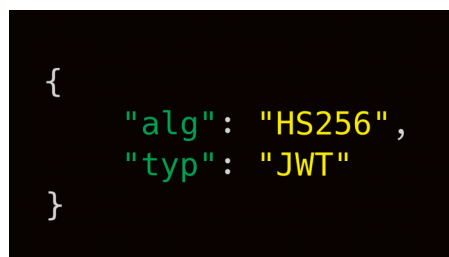
2.2.7 JSON Web Token (JWT)

JWT adalah sebuah token dengan format JSON yang dienkripsi, isi dari token tersebut berupa informasi pengguna yang gunanya untuk melakukan sistem autentikasi dan pertukaran informasi [7]. Sebuah token umumnya diterbitkan oleh penyedia layanan dan dikonsumsi oleh pihak yang mengandalkan konten untuk mengidentifikasi subjek token dengan tujuan yang berhubungan dengan keamanan [6].



Gambar 2.3: Struktur JSON Web Token (JWT)

JSON Web Token terdiri dari tiga bagian yang dipisahkan oleh titik “.” yaitu Header, Payload dan Signature. Header biasanya terdiri dari dua bagian: jenis token, yaitu JWT, dan algoritma hashing seperti HMAC SHA256.



Gambar 2.4: JWT Header

Bagian kedua dari JWT adalah payload, yang berisi klaim. Klaim adalah pernyataan tentang suatu entitas (biasanya, pengguna) dan *metadata* tambahan. Ada tiga jenis klaim, yaitu *registered*, *public*, dan *private claims*.

- Registered claims

Ini adalah seperangkat klaim yang telah ditentukan sebelumnya yang tidak wajib tetapi direkomendasikan, untuk memberikan serangkaian klaim yang berguna dan dapat dioperasikan. Beberapa di antaranya adalah:

- *iss* (Penerbit) mengidentifikasi pokok yang menerbitkan JWT. Pemrosesan klaim ini umumnya khusus untuk aplikasi. Nilai "*iss*" adalah string case-sensitive yang berisi nilai String atau URI. Penggunaan klaim ini adalah OPTIONAL atau tidak wajib.
- *exp* (Waktu kedaluwarsa) mengidentifikasi waktu kedaluwarsa pada atau setelah mana JWT tidak harus diterima untuk diproses. Pemrosesan klaim "*exp*" mensyaratkan bahwa tanggal / waktu saat ini harus sebelum tanggal / waktu kedaluwarsa yang tercantum dalam klaim "*exp*".
- *sub* (Subjek) mengidentifikasi pokok yang menjadi subjek JWT. Klaim dalam JWT biasanya merupakan pernyataan tentang subjek. Nilai subjek harus dicakup untuk menjadi unik secara lokal dalam konteks penerbit atau unik secara global. Pemrosesan klaim ini umumnya khusus untuk aplikasi. Nilai "*sub*" adalah string case-sensitive yang berisi nilai String atau URI. Penggunaan klaim ini adalah optional atau tidak wajib.
- *aud* (Audiens) mengidentifikasi penerima yang dimaksudkan untuk JWT. Setiap pelaku dimaksudkan untuk memproses JWT harus mengidentifikasi dirinya dengan nilai dalam klaim audiens. Jika pelaku

memproses klaim tidak mengidentifikasi dirinya dengan nilai dalam klaim "aud" ketika klaim ini hadir, maka JWT harus ditolak. Dalam kasus umum, nilai "aud" adalah array dari string *case-sensitive*, masing-masing berisi nilai String atau URI. Dalam kasus khusus ketika JWT memiliki satu pemirsa, nilai "aud" mungkin menjadi *case-sensitive* huruf tunggal yang berisi nilai String atau URI. Penafsiran nilai-nilai audiens umumnya spesifik aplikasi. Penggunaan klaim ini adalah optional atau tidak wajib.

- nbf (Bukan sebelumnya) mengidentifikasi waktu sebelum JWT tidak harus diterima untuk diproses. Pemrosesan klaim "nbf" mensyaratkan bahwa tanggal / waktu saat ini HARUS setelah atau sama dengan tanggal / waktu tidak-sebelum yang tercantum dalam klaim "nbf". Pelaksana mungkin menyediakan beberapa kelonggaran kecil, biasanya tidak lebih dari beberapa menit, untuk memperhitungkan kemiringan jam. Nilainya harus berupa angka yang berisi nilai Tanggal Numerik. Penggunaan klaim ini adalah optional atau tidak wajib.
- iat (Dikeluarkan pada) mengidentifikasi waktu ketika JWT diterbitkan. Klaim ini dapat digunakan untuk menentukan usia JWT. Nilainya harus berupa angka yang berisi nilai Tanggal Numerik. Penggunaan klaim ini adalah optional atau tidak wajib.
- jti (JWT ID) menyediakan pengidentifikasi unik untuk JWT. Nilai pengidentifikasi harus ditetapkan dengan cara yang memastikan bahwa ada kemungkinan yang dapat diabaikan bahwa nilai yang sama akan secara tidak sengaja ditetapkan ke objek data yang berbeda; jika aplikasi menggunakan beberapa emiten, tabrakan harus dicegah di antara nilai yang dihasilkan oleh emiten yang berbeda juga. Klaim "jti" dapat digunakan untuk mencegah JWT diputar ulang. Nilai "jti" adalah

string case-sensitive. Penggunaan klaim ini adalah optional atau tidak wajib.

- Public claims

Ini dapat didefinisikan sesuka hati oleh mereka yang menggunakan JWT. Tetapi untuk menghindari tabrakan mereka harus didefinisikan dalam *IANA JSON Web Token Registry* atau didefinisikan sebagai URI yang berisi *namespace* tahan tabrakan.

- Private claims

Ini adalah klaim khusus yang dibuat untuk membagikan informasi antara pihak-pihak yang setuju untuk menggunakannya dan bukan merupakan klaim terdaftar atau publik.

```
{
  "iss": "Tukutu",
  "exp": 12345678,
  "sub": 123,
  "jti": "elns-slkn-ss",
  "name": "elfahos",
  "merchant": true
}
```

Gambar 2.5: JWT Payload

Bagian ketiga dari JWT adalah signature, berisi hash dari komponen-komponen header, payload, dan kunci rahasia. Contoh JWT Signature ini menggunakan algoritme HMAC SHA-256.

```
HMACSHA256(  
    base64UrlEncode(header)+"."+br/>    base64UrlEncode(payload),  
    secret)
```

Gambar 2.6: JWT Signature

Hasil keluaran berupa token yang berisi struktur data JSON yang dienkripsikan dengan panjang 256 bit dari hasil enkripsi menggunakan algoritme HMAC SHA-256. Penerapan JWT dapat digunakan pada 2 kondisi yaitu kondisi saat autentikasi dan pertukaran informasi. Autentikasi merupakan kondisi yang umum ditemukan untuk menggunakan JWT, setelah pengguna login. Setiap permintaan yang dilakukan oleh client harus menyertakan JWT, yang dapat memungkinkan pengguna untuk akses router, service, dan sumber daya yang diizinkan dengan token yang dibuat.

2.2.8 Firebase Cloud Messaging (FCM)

Firebase Cloud Messaging (FCM) adalah solusi perpesanan lintas platform yang memungkinkan Anda mengirimkan pesan secara andal tanpa biaya.

Menggunakan FCM, Anda dapat memberi tahu aplikasi klien bahwa email baru atau data lain tersedia untuk disinkronkan. Anda dapat mengirim pesan notifikasi untuk mendorong keterlibatan dan retensi pengguna. Untuk kasus penggunaan seperti olahpesan cepat, pesan dapat mentransfer muatan hingga 4KB ke aplikasi klien [12].

2.2.9 MySQL

MySQL dikembangkan oleh sebuah perusahaan Swedia yang bernama MySQL AB, yang pada saat itu bernama TcX DataKonsult AB, sejak sekitar tahun 1994-1995. MySQL mempunyai fitur yang mudah dipelajari bagi para penggunanya dan dikembangkan untuk menangani database yang besar dengan waktu yang lebih singkat. Kecepatan, konektivitas, dan keamanannya yang lebih baik membuat MySQL sangat dibutuhkan untuk mengakses database di internet [13].

MySQL merupakan perangkat lunak dibawah lisensi GPL (General Public License). Pengguna dapat secara bebas menggunakan MySQL, tetapi dengan batasan perangkat lunak tersebut, pengguna tidak boleh dijadikan produk turunan yang bersifat komersial MySQL dapat berjalan di atas banyak sistem operasi seperti Solaris, MAC OS X, linux, dan FreeBSD [14].



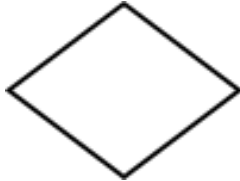
MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *database* yang terbagi menjadi tiga bagian, yaitu DDL, DML dan DCL [15].



1. Data Definition Language (DDL) merupakan kelompok perintah yang berfungsi untuk mendefinisikan atribut – atribut basis data, tabel, atribut (kolom), batasan – batasan terhadap suatu atribut, serta hubungan antar tabel.
2. Data Manipulation Language(DML) adalah kelompok perintah yang berfungsi untuk memanipulasi data dalam *database*, misalnya untuk pengambilan, penyisipan, pengubahan, dan penghapusan data.
3. Data Control Language(DCL) berisi perintah-perintah untuk mengendalikan pengaksesan data. Pengendalian dapat dilakukan berdasarkan pengguna, tabel, kolom maupun operasi yang boleh dilakukan.

2.2.10 Flowchart

Flowchart yaitu penggambaran secara grafik dari setiap langkah – langkah dan urutan prosedur dari suatu program. Dengan membuat *flowchart* dapat membantu *analyst* dan *programmer* untuk memecahkan masalah ke dalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasiannya. *Flowchart* di visualkan dengan bentuk gambar diagram yang memiliki aliran satu atau dua arah secara sekuensial.

Tabel 2.2: Komponen Flowchart

Simbol	Nama	Keterangan
	Terminator	Menunjukkan awal atau akhir sebuah proses.
	Process	Menyatakan kegiatan yang akan terjadi dalam diagram
	Decision	Proses / langkah di mana perlu adanya keputusan atau adanya kondisi tertentu. Di titik ini selalu ada dua keluaran untuk melanjutkan aliran kondisi yang berbeda.


	Input/Output Data	Digunakan untuk mewakili data masuk, atau data keluar.
	Flow Line	Menunjukkan arah aliran algoritme, dari satu proses ke proses berikutnya.



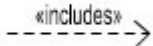

2.2.11 Use Case Diagram

Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML adalah sebuah tool yang dikembangkan dengan paradigma Object – Oriented. UML menyediakan beberapa diagram untuk memodelkan aplikasi berorientasi objek, salah satunya yaitu use case diagram.

Use Case mewakili bagaimana sistem berinteraksi dengan lingkungannya dengan menggambarkan aktivitas yang dilakukan oleh pengguna dan respon sistem. *Use case* diagram merupakan representasi interaksi pengguna dengan sistem dan menggambarkan spesifikasi dari kasus penggunaan. Elemen pada use case meliputi *actor*, *use case*, *system boundary*, dan *relationship*.

Tabel 2.3: Komponen Use Case Diagram

Gambar	Nama	Keterangan
	Actor	Mempresentasikan seseorang/himpunan peran yang dimainkan ketika berinteraksi dengan use case.

	Use Case	Deskripsi dari urutan aksi - aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	Association	Sebagai penghubung antara satu objek dengan objek lainnya.
	Includes	Kelakuan yang harus terpenuhi agar sebuah event dapat terjadi, dimana pada kondisi ini sebuah <i>use case</i> adalah bagian dari <i>use case</i> lainnya.
	Boundary	Batasan yang dari sistem terhadap lingkungan.

2.3 Deskripsi Tempat KKI

PT. Dian Nuswantoro Teknologi dan Informasi merupakan perusahaan di bidang teknologi komputerisasi yang berdiri sejak tahun 2006. PT Dian Nuswantoro Teknologi dan Informasi ini lebih dikenal dengan nama PT Dinustek berfokus pada pelayanan tentang informasi dan teknologi (IT) , Layanan jaringan internet (Internet Service Provider) dan layanan pembuatan aplikasi (Software Development).

Informasi tentang PT Dinustek :

Alamat : Jl. Arjuna No 36 ,Kota Semarang, Jawa Tengah, Indonesia.
Telepon : (024) 3568492
Email : marketing@dinustek.com
Website : www.dinustek.com

2.3.1 Logo dan Makna Tempat KKI



Gambar 2.7: Logo PT Dian Nuswanto Teknologi dan Informasi

Logo PT Dinustek terbagi menjadi 3 (tiga) bagian yaitu :

1. Simbol Api

Simbol api berwarna oranye melambangkan solusi untuk klien yang bermaksud memberikan penyelesaian masalah terbaik dari sisi teknologi dan informasi bagi bisnis klien.

2. Warna Biru

Warna biru melambangkan solusi teknologi informasi yang kami tawarkan dapat dipercaya kehandalan bagi klien.

3. Bentuk Huruf

Huruf-huruf yang tersusun berbentuk small caps melambangkan keramah-tamahan perusahaan kepada klien, dan nuansa kekeluargaan di perusahaan. Bentuk huruf yang dinamis namun kuat menggambarkan tim perusahaan yang kreatif dan solid.

2.3.2 Struktur Organisasi Tempat KKI

Struktur organisasi PT Dinustek:

1. Direktur Utama
Mohamad Sidiq, S.Si., M. Kom
2. Direktur
Dr. Pulung Nurtantio Andono, ST, M. Kom
3. Manager of Software Department
Abu Salam, M. Kom
4. Kepala Divisi Stratup Software
Moh. Tofa Nurzaki, S. Kom

2.3.3 Visi dan Misi Tempat KKI

Visi : Menjadi perusahaan teknologi informasi dan komunikasi kelas dunia.

Misi : Memberikan solusi keseluruhan teknologi informasi dan komunikasi untuk klien.