SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

IMPLEMENTASI SISTEM LOGISTIK YANG DAPAT MENAMPUNG LONJAKAN PADA INSTANSI KESEHATAN DENGAN MEMANFAATKAN TEKNOLOGI *SERVERLESS* PADA LAYANAN *AWS SERVICES*

TUGAS AKHIR

Arya Maulana Hidayatullah
0110220144

PROGRAM STUDI TEKNIK INFORMATIKA
DEPOK
AGUSTUS 2024

SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

IMPLEMENTASI SISTEM LOGISTIK YANG DAPAT MENAMPUNG LONJAKAN PADA INSTANSI KESEHATAN DENGAN MEMANFAATKAN TEKNOLOGI *SERVERLESS* PADA LAYANAN *AWS SERVICES*

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana

Arya Maulana Hidayatullah

0110220144

PROGRAM STUDI TEKNIK INFORMATIKA

DEPOK

AGUSTUS 2024

HALAMAN PERNYATAAN ORISINALITAS

Skripsi/Tugas Akhir ini adalah hasil karya penulis, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama: Arya Maulana Hidayatullah

NIM : 0110220144

Depok , Agustus 2024

Tanda Tangan

Arya Maulana Hidayatullah

HALAMAN PENGESAHAN

Skripsi/Tugas Akhir ini diajukan oleh :

Nama :Arya Maulana Hidayatullah

NIM : 0110220144

Program Studi : Teknik Informatika

Judul Skripsi : IMPLEMENTASI SISTEM LOGISTIK yang dapat MENAMPUNG LONJAKAN pada INSTANSI KESEHATAN dengan MEMANFAATKAN TEKNOLOGI *SERVERLESS* Pada LAYANAN *AWS SERVICES*

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana komputer pada Program Studi Teknik Informatika , Sekolah Tinggi Teknologi Terpadu Nurul Fikri

DEWAN PENGUJI

Pembimbing	Penguji
(April Rustianto S.Komp., M.T.)	(Imam Haromain, S.Si., M.Kom.)

Ditetapkan di : Depok

Tanggal : 13 Agustus 2024

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan skripsi/Tugas Akhir ini. Penulisan skripsi/Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana komputer Program Studi Teknik Informatika pada Sekolah Tinggi Teknologi Terpadu Nurul Fikri Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi penulis untuk menyelesaikan skripsi/tugas akhir ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Allah SWT.
2. Orang tua dan semua anggota keluarga yang telah memberikan dorongan baik secara moril maupun materil dalam penyelesaian tugas ini.
3. Bapak Dr. Lukman Rosyidi, M.T., M.M. selaku Ketua Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
4. Ibu Tiffany Nabarian, S.Kom., M.T.I selaku Ketua Program Studi Teknik Informatika Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
5. Ibu Nurul Janah, S.IIP., M.Hum. selaku Dosen Pembimbing Akademik yang telah membimbing penulis selama berkuliah di Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
6. Bapak April Rustianto, S.Komp., M.T. selaku Dosen Pembimbing Tugas Akhir penulis dalam menyelesaikan penulisan ilmiah ini.
7. Bapak Imam Haromain, S.Si., M.Kom. selaku Dosen Penguji Tugas Akhir di Sekolah Tinggi Teknologi Terpadu Nurul Fikri .
8. Para Dosen di lingkungan Sekolah Tinggi Teknologi Terpadu Nurul Fikri yang telah membimbing penulis dalam menuntut ilmu yang telah diberikan.

Dalam penulisan ilmiah ini tentu saja masih banyak terdapat kekurangan-kekurangan yang mungkin disebabkan oleh keterbatasan kemampuan dan pengetahuan yang penulis miliki. Walaupun demikian, penulis telah berusaha menyelesaikan penulisan ilmiah ini sebaik mungkin. Oleh karena itu apabila

terdapat kekurangan di dalam penulisan ilmiah ini, dengan rendah hati penulis menerima kritik dan saran dari pembaca.

Akhir kata, penulis berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 13 Agustus 2024

Arya Maulana Hidayatullah

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Sekolah Tinggi Teknologi Terpadu Nurul Fikri, saya yang bertanda tangan di bawah ini:

Nama : Arya Maulana Hidatullah

NIM : 0110220144

Program Studi : Teknik Informatika

Jenis karya : Skripsi / Tugas Akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada STT-NF **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty - Free Right*)** atas karya ilmiah saya yang berjudul :

Implementasi Sistem Logistik yang dapat Menampung Lonjakan pada Instansi Kesehatan dengan Memanfaatkan Teknologi *Serverless* pada Layanan *AWS Services*

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini STT-NF berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 13 Agustus 2024

Yang Menyatakan

(Arya Maulana Hidayatullah)

ABSTRAK

Nama : Arya Maulana Hidayatullah
NIM : 0110220144
Program Studi : Teknik Informatika
Judul : Implementasi Sistem Logistik yang dapat Menampung Lonjakan pada Instansi Kesehatan dengan Memanfaatkan Teknologi *Serverless* pada Layanan *AWS Services*

Peningkatan kebutuhan layanan kesehatan dalam beberapa tahun terakhir sangat tinggi terutama pada akses yang cepat, efisien, dan andal terhadap layanan kesehatan. Salah satu tantangan utama yang dihadapi oleh penyedia layanan kesehatan adalah kemampuan untuk menanggapi lonjakan pengguna secara efektif, terutama dalam konteks penyediaan layanan logistik. Penelitian ini bertujuan untuk mengimplementasikan sistem logistik yang dapat menampung lonjakan permintaan disebuah instansi kesehatan dengan memanfaatkan teknologi serverless pada layanan *AWS* (*Amazon Web Services*). Teknologi *serverless* memungkinkan pengelolaan sumber daya yang fleksibel dan skalabilitas yang tinggi tanpa perlu manajemen server secara manual. Metode yang akan dilakukan untuk penelitian ini adalah *Load Testing* dengan mengirimkan data dalam ukuran yang besar. Hasil penelitian menunjukkan bahwa sistem yang diusulkan mampu meningkatkan efisiensi operasional, mengurangi waktu respons, dan memastikan ketersediaan logistik yang tepat waktu selama periode kritis. Dengan demikian, implementasi teknologi serverless pada layanan AWS terbukti sebagai solusi inovatif dan efektif untuk menghadapi tantangan logistik di sektor kesehatan.

Kata kunci :Layanan kesehatan, *AWS* (*Amazon Web Services*), *Load testing*, *Serverless*, logistik

ABSTRACT

Name : *Arya Maulana Hidayatullah*
NIM : *0110220144*
Study Program : *Informatic Enginering*
Title : *Implementation Of A Logistics System That Can Accept Surges In Health Agency By Utilizing Serverless Technology On Aws Services*

The increasing need for health services in recent years has increased public demand for fast, efficient and reliable access to health services. One of the main challenges faced by healthcare providers is the ability to respond effectively to surges in users, especially in the context of logistics service provision. This research aims to implement a logistics system that can accommodate surges in demand in a health agency by utilizing serverless technology on AWS (Amazon Web Services) services. Serverless technology enables flexible resource management and high scalability without the need for manual server management. The method that will be used for this research is Load Testing by sending data in large sizes. The research results show that the proposed system is able to increase operational efficiency, reduce response time, and ensure timely availability of logistics during critical periods. Thus, the implementation of serverless technology on AWS services has proven to be an innovative and effective solution to face logistics challenges in the health sector.

Key words : Health services, AWS (Amazon Web Services), Load testing, Serverless, logistics

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar belakang

Peningkatan kebutuhan akan layanan kesehatan dalam beberapa tahun terakhir selaras dengan meningkatnya permintaan masyarakat akan akses yang cepat seiring dengan perkembangan teknologi, efisien, dan dapat diandalkan terhadap layanan kesehatan. Salah satu tantangan yang dihadapi oleh suatu penyedia layanan kesehatan adalah kemampuan untuk menanggapi lonjakan pengguna secara efektif, terutama dalam konteks penyediaan layanan logistik yang mendukung rantai pasok kesehatan. Sistem logistik penyedia layanan kesehatan berperan penting dalam memastikan distribusi yang efisien dan tepat waktu dari berbagai produk dan layanan kesehatan ke instansi kesehatan yang membutuhkan.

Logistik adalah suatu rangkaian upaya yang mencakup efektivitas perencanaan, implementasi, sampai pengawasan atas suatu proses perpindahan produk barang atau jasa, energi, atau sumber daya lain, dari mulai titik awal hingga titik pengguna. Seluruh aktivitas logistik dilakukan untuk mencapai tujuan utama, yaitu memastikan ketersediaan barang dan pengiriman tepat waktu ke lokasi instansi kesehatan yang dituju [1]. Sesuai dengan tuntutan, peningkatan kebutuhan kesehatan yang diharapkan terorganisir dengan tetap menyesuaikan dengan era digital saat ini, Teknologi Informasi (TI) telah menjadi salah satu aspek kunci dalam berbagai organisasi dan bisnis. Infrastruktur TI yang kuat dan efisien sangat penting untuk mendukung operasional sehari-hari, inovasi, dan pertumbuhan bisnis. Namun, dengan perkembangan yang cepat dalam dunia TI [2], perusahaan sering menghadapi tantangan-tantangan untuk menjaga infrastruktur mereka agar selalu efisien, aman, dan mampu menangani beban kerja yang semakin besar. Oleh karena itu pemanfaatan teknologi *cloud computing* telah muncul sebagai solusi yang inovatif dan efektif untuk mengatasi tantangan ini.

Cloud computing memungkinkan perusahaan untuk menyimpan, mengelola, dan mengakses sumber daya komputasi dan data melalui internet,

yang memungkinkan skalabilitas, fleksibilitas, dan efisiensi yang lebih besar. Di sektor publik, cloud computing memanajemen logistik dan melibatkan peran organiasi kesehatan dari berbagai level, mulai dari tingkat pusat, provinsi, kabupaten/kota dan fasilitas pelayanan kesehatan, yang dimana masing-masing level memiliki wewenang dan tanggung jawab yang berbeda terkait dengan pengelolaan logistik. Oleh karena itu *Serverless* merupakan layanan *Cloud Computing* yang banyak dimanfaatkan untuk membangun dan menjalankan berbagai aplikasi dan layanan tanpa perlu mengelola infrastruktur.

Beberapa server masih menjalankan suatu aplikasi, tetapi semua manajemen server dilakukan oleh penyedia layanan. Seiring dengan pesatnya pertumbuhan pengembangan tanpa server (*Serverless*) dalam beberapa tahun terakhir, layanan yang tersedia bagi pengembang semakin bertambah jumlah dan cakupannya. Adapun layanan yang menyediakan jasa *cloud service provider* salah satunya *AWS* (*Amazon Web Services*). Keberhasilan penyedia layanan kesehatan bergantung pada kemampuan infrastruktur logistik. Infrastruktur logistik yang efesien harus mampu mengantisipasi perubahan permintaan untuk memenuhi kebutuhan dalam waktu singkat. salah satu layanan *cloud service provider* yang disediakan oleh *AWS* ialah *serverless*, yang bertujuan untuk memonitoring dan mengatasi kelonjakan yang dapat mempengaruhi responsitivitas dan efektifitas terhadap penyediaan layanan kesehatan. Berdasarkan latar belakang tersebut, maka penulis akan menjelaskan lebih lanjut mengenai "**IMPLEMENTASI SISTEM LOGISTIK YANG DAPAT MENAMPUNG LONJAKAN PADA INSTANSI KESEHATAN DENGAN MEMANFAATKAN TEKNOLOGI *SERVERLESS* PADA LAYANAN *AWS SERVICES***"

1.2 Rumusan Masalah

Rumusan masalah merujuk kepada latar belakang yang sudah dikemukakan oleh peneliti adalah sebagai berikut:

- Bagaimana rancangan sistem berbasis serverless *AWS* yang dapat memantau penggunaan database terhadap sistem logistik pada instansi kesehatan ?
- Bagaimana cara menanggulangi kelonjakan penggunaan database di sektor logistik pada suatu instansi kesehatan dengan menggunakan layanan Serverless *AWS*?

1.3 Tujuan dan Manfaat Penelitian

Tujuan dari penelitian diantaranya:

- Melakukan perancangan serverless berbasis *AWS* yang dapat membantu memonitoring atau memantau pengunaan database logistik kesehatan di sebuah rumah sakit
- Melakukan uji coba terhadap rancangan serverless yang dapat memonitoring penggunaan database logistik kesehatan di sebuah rumah sakit tertentu yang sudah mendapatkan izin terkait ujicoba serverless berbasis *AWS*, sehingga peniliti dapat memberikan kesimpulan dari rancangan serverless berbasis *AWS* tersebut.

Manfaat yang ingin disampaikan yaitu dapat memberikan manfaat positif bagi instanti kesehatan (rumah sakit) di sektor logistik kesehatan, membantu kinerja instansi kesehatan (rumah sakti) untuk memonitoring database yang dimiliki terutama di sektor logistik kesehatan. Apabila terjadi kelonjakan pengunaan database, maka instansi kesehatan dapat melakukan penanggulangan pertama dalam mengatasi kelonjakan yang cukup pesat.

1.4 Batasan Masalah

Batasan permasalah dari penilitan ini adalah sebagai berikut:

1. Peniliti hanya melakukan perancangan *serverless* berbasis *AWS* yang memonitoring barang yang di order oleh instansi kesehatan serta memonitoring apabila terjadi kelonjakan penggunaan database.
2. Adapun tools yang akan digunakan dalam pembuatan *serverless* berbasis *AWS* yang di gunakan seperti *Amazon S3, AWS Lambda, Amazon API Gateway, Amazon Cloudwatch dan Amazon database*.
3. Untuk penelitian ini penulis akan menggunakan salah satu tools yang ada di *AWS* sebagai *DB* (*Databases*) yaitu *DynamoDB* yang akan digabungkan oleh *Amazon Cloudwatch* untuk memberikan alert dari *DynamoDB* apabila terjadi kelonjakan penggunaan atau terjadi kesalahan pada *DB*.

1.5 Sistematika Penulisan

Agar penulisan ini dapat tersusun dengan baik maka sistematika penulisan terdiri dari :

BAB I : PENDAHULUAN

Bab ini membahas mengenai latar belakang penulisan, rumusan masalah, tujuan penelitian , manfaat penelitian , batasan masalah, dan sistematika penulisan penelitian yang berjudul "IMPLEMENTASI SISTEM LOGISTIK YANG DAPAT MENAMPUNG LONJAKAN PADA INSTANSI KESEHATAN DENGAN MEMANFAATKAN TEKNOLOGI *SERVERLESS* PADA LAYANAN *AWS SERVICES"*

BAB II : KAJIAN LITERATUR

Bab ini penulis akan menyajikan landasan-landasan teori dan jurnal terkait dengan topik penelitian. Teori-teori ini bertujuan untuk mempertajam pemahaman peneliti, serta dalam bab ini juga peneliti menyajikan beberapa penelitian terdahulu yang terkait dengan penelitian yang akan dilaksanakan. Adapun teori yang dibahas pada bab ini yaitu penjelasan tentang sistem, penjelasan tentang *Cloud Computing*, layanan *Serverless Computing*,

penjelasan terkait *AWS tools* yang akan di gunakan salah satu nya *Amazon S3 Buckets*, *Amazon DynamoDB*, *AWS Lambda*, *Amazon API Gateway*, *AWS IAM*, dan *Amazon CloudWatch*, metode *load testing*, dan penjelasan tentang sistem logistik .

BAB III : METODOLOGI PENELITIAN

Bab ini akan mejelaskan waktu, tempat, dan metode yang akan digunakan peneliti dalam melaksanakan penelitian. Pada tahap ini dijelaskan juga rancangan penelitian yang terdiri dari beberapa metode yang digunakan yaitu, metode analisis yang digunakan ialah metode kuantitatif, metode pengumpulan data yang digunakan adalah metode eksperimen, metode pengujian yang digunakan adalah metode *Load Testing*, dan jenis penelitian yang digunakan yaitu penelitian dan pengembangan atau dapat disebut (*Research and Development (R&D)).*

BAB IV : IMPLEMENTASI dan EVALUASI

Bab ini akan menjelaskan tentang implementasi dari rancangan yang telah dibuat berserta bukti pengerjaanya serta memberikan evaluasi terhadap rancangan yang telah dibuat. Dalam hal ini setelah di lakukan perancangan sebuah sistem oleh peneliti lalu di lanjutkan dengan melakukan implementasi kepada sistem tersebut dengan tujuan apakah sistem yang sudah di rancang tersebut berjalan dengan lancar atau tidak berjalan dengan lancar.

BAB V : KESIMPULAN dan SARAN

Di bab ini berisikan kesimpulan dari hasil implementasi yang sudah dilakukan oleh peneliti dan saran untuk penelitian kedepannya atas apa yang telah dilakukan selama proses penelitian dan penulisan berlangsung serta harapan oleh peneliti terhadap hasil penelitan ini.

BAB II
KAJIAN LITERATUR

2.1 Sistem

Menurut Meriam-Webster sistem adalah “interaksi secara teratur atau kelompok item yang saling bergantung membentuk satu kesatuan yang utuh. Sistem merupakan seperangkat ajaran, gagasan, atau asas yang terorganisasi biasanya dimaksudkan untuk menjelaskan pengaturan atau cara kerja dari keseluruhan yang sistematis”. Sistem merupakan suatu wadah untuk menjalankan seluruh perintah yang telah dibuat untuk mencapai suatu tujuan utama. Menurut Azhar Susanto sistem adalah “Kumpulan atau grup dari sub sistem atau bagian atau komponen apapun baik fisik maupun non fisik yang saling berhubungan satu sama lain dan dapat bekerja sama untuk mencapai satu tujuan tertentu”. Sistem menurut Adani ialah “sekelompok atau sekumpulan proses yang dimana kata data dapat diolah, dianalisis, dan di tampilkan supaya data tersebut berguna untuk menunjang pengambilan suatu keputusan” [3]. Dari penjelasan oleh para ahli diatas dapat disimpulkan bahwa sistem adalah kumpulan dari beberapa elemen, himpunan dari suatu unsur, dan komponen fungsional yang saling berhubungan satu sama lainnya.

2.2 *Cloud Computing*

Cloud Computing, juga dikenal sebagai komputasi awan, adalah kombinasi komputasi dan pengembangan internet. *Cloud computing* dapat diartikan sebagai penyimpanan data yang menggunakan internet sebagai media transmisinya. Ada tiga jenis komputasi awan diantaranya:

a. SAAS (*Software As A service*), layanan cloud dalam bentuk perangkat lunak (*software*)

b. PAAS (*Platform As A Service*) merupakan layanan cloud berupa platform untuk membangun aplikasi.

c. IAAS (*Infrastructure As a Service*) adalah layanan cloud yang dirancang sesuai kebutuhan pengguna.Layanan ini sederhana.

Artinya pengguna dapat mengakses datanya kapan saja dan dimana saja melalui jaringan koneksi internet yang tersedia. Bagi pengguna yang menggunakan komputer cloud, kapasitas yang disediakan cloud sangatlah

besar sehingga data dapat disimpan tanpa memakan banyak memori komputer [4].

2.3 Layanan *Serverless Computing*

Serverless Computing adalah model pemrograman di mana berintegrasi dengan *cloud computing* yang disediakan oleh *cloud provider*. Dalam hal ini, *cloud provider* berfungsi dalam penyediaan infrastruktur server, manajemen sistem, dan segala hal berkaitan dengan penyimpanan dan pengoperasian sistem. Singkatnya, *user* tidak perlu memikirkan pemeliharaan dan kendala pada *server* karena hal tersebut sudah menjadi tanggung jawab *developer* atau dalam hal ini adalah *cloud provider*. *User* hanya fokus mengembangkan kode fungsi dan desain *website* supaya terlihat menarik dan berfungsi dengan baik dalam menjalankan perintah. Pengguna tidak harus direpotkan untuk mengatur operasional *server*, *scaling* (penambahan dan pengurangan jaringan), atau install *OS management servers*. Dengan kata lain, para *user* hanya terima jadi *server* yang disediakan[5].

Adapun komponen Serverless Computing diantaranya : *Cloud Computing*, *API* (*Application Programming Interface*), dan Penyimpanan Objek. *Serverless Computing* juga merupakan model *cloud computing* yang memungkinkan *developer software* untuk mengembangkan dan menjalankan suatu aplikasi di *server* tanpa harus melakukan *provisioning* atau pemeliharasan infrastruktur *back-end*. Hal itu lantaran tugas rutin seperti melakukan manajemen dan perawatan terhadap infrastruktur, termasuk melakukan *update OS*, *monitoring* sistem, dan merencanakan kapasitas penggunaan telah dilakukan oleh *vendor cloud*, sehingga *developer* dapat fokus melakukan *coding*. Serverless Computing memliki kelebihan yaitu: scalling otomatis, tarif server yang terjangkau, dan ketersediaan data dan sistem yang stabill. Namun Serverless Computing memiliki beberapa kekurangan, diantaranya: kode pemrograman yang terbatas, pembatasan sumber daya (kapasitas memori), dan kesulitan mengatasi bug ketika terjadi nya bug pada Serverless Computing.

Serverless Computing juga dapat dilaksanakan dalam ruang lingkup yang terisolasi, oleh karena itu hal tersebut dimanfaatkan dalam pengujian dari beberapa hipotetis yang dilakukan dengan menggunakan metode paralel yang memberikan keuntungan oleh pengembangnya. Namun adanya halangan di *Serverless Computing* tersebut diantaranya jangka waktu dan kapasitas penyimpanan (*Memory*) yang harus ditangani. Di era modern *Serverless Computing* menjadi alternatif yang pantas untuk melakukan latihan model yang sudah ada pada saat ini[6].

2.4 *Amazon S3 Buckets*

Amazon Simple Storage Service atau di singkat sebagai *Amazon S3* adalah sebuah layanan yang ada di *AWS* yang memberikan layanan penyimpanan dalam bentuk objek manapun, seperti: data, gambar, dan masih banyak lagi. *Amazon S3* juga menawarkan fasilitas layanan seperti menyimpan dan mengambil data dalam jumlah yang bervariasi dan waktu yang bisa di sesuaikan oleh pengguna-nya, skalabilitas, ketersediaan data , keamanan, dan kinerja yang terdepan [7].

2.5 *AWS Lambda*

AWS Lambda merupakan sebuah layanan komputasi yang memungkinkan Anda menjalankan kode tanpa menyediakan atau mengelola server. *Lambda* akan menjalankan kode kita pada infrastruktur komputasi dengan ketersediaan tinggi dan melakukan semua administrasi sumber daya komputasi, termasuk pemeliharaan server dan sistem operasi, penyediaan kapasitas dan penskalaan secara otomatis, dan pencatatan. Dengan *Lambda*, yang perlu kita lakukan hanyalah menyediakan kode kita di salah satu *runtime* bahasa yang didukung Lambda [8].

2.6 *Amazon API Gateway*

Amazon API Gateway adalah *AWS* layanan untuk membuat, menerbitkan, memelihara, memantau, dan mengamankan *REST*, *HTTP*, dan *WebSocket API* dalam skala apa pun. Pengembang *API* dapat membuat *API* yang mengakses satu layanan *AWS* atau layanan web lainnya, serta data yang

disimpan di AWSCloud. Sebagai pengembang *API Gateway*, kita dapat membuat *API* untuk digunakan dalam aplikasi klien Anda sendiri. Atau Anda dapat membuat *API* Anda tersedia untuk pengembang aplikasi pihak ketiga [9].

API Gateway membuat sebuah *API RESTful* yang:

- Berbasis *HTTP*.
- Menerapkan metode *HTTP* standar seperti *GET*, *POST*, *PUT*, *PATCH*, dan *DELETE*.

Amazon API Gateway dapat diartikan juga sebagai salah satu tools dari *AWS (Amazon Web Service)* yang memungkinkan pengguna/pengembang menghubungkan aplikasi *non-AWS* dengan sumber daya *back-end AWS*, seperti kode dan server. Meningkatkan akses pelanggan AWS ke aplikasi yang kompatible dan keseluruhan utilitas layanan *Amazon Cloud Service* lainnya. *API* (*Application Program Interface*) memungkinan program software untuk berkomunikasi, membuat lebih banyak fungsional. Pengguna AWS dapat membuat, mengelola, dan menjaga *API* yang sudah ada dengan menggunakan *Amazon API Gateway*.

2.7 *Amazon Cloudwatch*

Amazon CloudWatch merupakan layanan yang memantau sumber daya dan aplikasi *Amazon Web Services* (*AWS*) yang berjalan di *AWS* dari waktu ke waktu. *CloudWatch* dapat mengumpulkan dan melacak metrik, yang merupakan indikator sumber daya dan aplikasi pengguna yang dapat diukur. Halaman beranda *CloudWatch* secara otomatis menampilkan metrik untuk setiap layanan *AWS* yang pengguna akan gunakan. Pengguna dapat membuat *dashboard* khusus untuk menampilkan metrik untuk aplikasi tertentu dan menampilkan kumpulan metrik khusus pilihan Anda. Anda dapat memantau metrik dan membuat alarm untuk mengirimkan pemberitahuan atau secara otomatis mengalihkan sumber daya untuk memantau ketika ambang batas dilanggar. Misalnya, Anda dapat memantau CPU dan penggunaan baca dan disk dengan memokuskan bagian apa yang akan dipantau atau dimonitoring

oleh *CloudWatch* , lalu menggunakan data tersebut untuk memutuskan apakah Anda perlu meluncurkan instans tambahan untuk menanganinya seiring bertambahnya beban. Anda juga dapat menggunakan data ini untuk menghemat uang dengan menghapus waktu yang tidak terpakai. *CloudWatch* memberikan visibilitas sistem terhadap pemanfaatan sumber daya, kinerja aplikasi, dan kesehatan operasional [10].

2.8 *Amazon DynamoDB*

Amazon DynamoDB adalah layanan database *NoSQL* yang dikelola sepenuhnya yang memberikan kinerja yang cepat dan dapat diprediksi serta skala yang belum pernah terjadi sebelumnya. *DynamoDB* mengurangi beban administratif dalam mengelola dan menskalakan database terdistribusi sehingga Anda tidak perlu khawatir tentang penyediaan perangkat keras, konfigurasi dan konfigurasi, replikasi, patching perangkat lunak, atau penskalaan klaster. *DynamoDB* juga menyediakan enkripsi saat istirahat, menghilangkan beban administratif dan kompleksitas yang terkait dengan perlindungan data sensitif. *DynamoDB* dapat membuat tabel database yang dapat menyimpan dan mengambil data dalam jumlah besar dan menangani berbagai tingkat lalu lintas aplikasi. Pengguna dapat menambah atau mengurangi kapasitas akses tabel Anda tanpa downtime atau penurunan kinerja. Anda dapat memantau penggunaan sumber daya dan metrik kinerja menggunakan *AWS Management Console* [11].

DynamoDB menyediakan kemampuan aplikasi manual. Hal ini dapat memungkinkan pengguna membuat cadangan lengkap spreadsheet pengguna untuk penyimpanan dan pengarsipan jangka panjang sesuai dengan persyaratan kepatuhan peraturan. pengguna dapat membuat cadangan sesuai permintaan yang dapat melakukan pemulihan tabel *Amazon DynamoDB* Anda secara *real-time*. Pemulihan *point-in-time* membantu melindungi tabel Anda dari operasi penulisan atau penghapusan yang tidak disengaja. Pemulihan titik waktu memungkinkan Anda memulihkan tabel ke lokasi tertentu dalam 35 hari terakhir. *DynamoDB* juga dapat secara otomatis menghapus item basi dari tabel Pengguna, serta membantu mengurangi

penggunaan penyimpanan dan biaya penyimpanan data basi alias data yang sudah tidak terpakai.

2.9 *AWS IAM*

AWS Identity and Access Management atau disingkat sebagai *AWS IAM* adalah sebuah layanan yang disediakan oleh *AWS* untuk membantu pengguna untuk memberikan akses kepada *AWS tools* yang akan digunakan, serta mengontrol akses kepada *AWS tools* yang akan digunakan [12]. Contoh nya seorang pengguna akan memberikan akses *AWS Lambda* untuk mengintergerasi pengiriman atau mengupload data dari *Amazon S3 (Simple Storage Service)* ke *DynamoDB* dengan menggunakan *IAM Role*. *IAM Role* sendiri ialah sebuah identitas dari *IAM* (*IAM Identity*) yang dimana pengguna dapat membuat nya pada akun *AWS Console* pengguna yang mempunyai izin yang spesifik kepada layanan *AWS* yang akan digunakan.

2.10 *Research And Development (R&D)*

Research and Development disingkat sebagai *R&D* ialah sebuah metode penelitian yang digunakan sebagai pengembangan dan pengujian kepada suatu produk yang kedepan-nya akan dikembangkan dalam dunia pendidikan [7]. Di sektor lain yaitu di sektor perusahaan metode *R&D* bertujuan untuk meluncurkan sebuah produk baik dalam bentuk perangkat lunak yang akan di buat oleh peruhsaan tersebut atau jasa yang orisinal dan dapat berjalan secara optimal dan berguna untuk pengguna-nya. Adapun di sektor teknologi metode *R&D* tersebut berguna untuk menghasilkan sebuah aplikasi atau sistem yang digunakan oleh pengguna-nya dan berjalan secara optimal

2.11 *Load Testing*

Load Testing merupakan jenis pengujian sistem yang atau perangkat lunak yang sudah di rancang untuk menentukan apakah sistem atau perangkat tersebut dalam menanggulangi sebuah data yang mempunyai beban yang berlebih bertujuan untuk memastikan apakah sitem atau perangkat lunak yang sudah di buat terjadi kelemahan (*vulnerbillity*) baik dalam bentuk pengurangan perfoma ketika membaca suatu data atau kapasitas melebihi batas yang sudah di tetapkan

dari sistem atau perangkat lunak tersebut. Adapun cara kerja metode tersebut ialah penguji membuat sebuah *source code* dalam bentuk *script* baik dalam bentuk bahasa pemrograman *python* atau *javascript* yang akan dimasukan kedalam sistem atau perangkat lunak tersebut dan metode ini di lakukan secara bertahap [8].

2.12 Sistem Logistik

Sistem Logistik berasal dari 2 jenis penjelasan yaitu Sistem dan Logistik . Sistem merupakan kumpulan-kumpulan dari beberapa elemen, komponen fungsional, dan himpunan dari suatu unsur yang berkaitan satu sama lain nya, sedangakan Logistik menurut Chirstoper bahwa "Logistik merupakan suatu proses yang dilakukan secara strategis dalam mengelola pengadaan, pergerakan dan penyimpanan material suku cadang dan aliran informasi terkait dengan organisasi" [9]]. Hal ini dapat disimpulkan bahwa logistik merupakan ilmu pengetahuan dalam melakukan kegiatan penyimpanan, pemeliharaan dan penyaluran , serta penghapusan suatu barang tertentu. Dari penjelasan diatas dapat disimpulkan bahwa sistem logistik ialah sebuah sistem yang dapat menyimpan data, memelihara dan menyalurkan data, serta menghapus suatu data tertentu jika tidak diperlukan.

2.13 Metode Eksperimen

Metode eksperimental ialah salah satu dari pendekatan penelitian dimana peneliti mengontrol dan memanipulasi *variabel* independen untuk mengamati efeknya terhadap *variabel* dependen. Eksperimental adalah metode yang paling umum digunakan dalam penelitian ilmiah untuk menentukan hubungan sebab-akibat antara *variabel*. Metode eksperimental banyak digunakan dalam ilmu alam dan sosial, seperti psikologi, pendidikan, pengembangan teknologi, dan ilmu kesehatan, karena dapat memberikan bukti yang kuat tentang hubungan sebab-akibat, dalam sektor teknologi metode eksperimen menjadi penentu apakah solusi dari perancangan suatu teknologi efektif dan sesuai dengan tujuan yang di harapkan atau belum efektif dan belum sesuai dengan tujuan yang di harapkan [10]

2.14 Penelitian Terkait

Tabel 2. 1 Penelitian Terkait

No	Nama dan Tahun	Judul	Topik	Subjek	Hasil
1	Faizal Ramadhan, 2020	Manajemen Logistik Alat Kesehatan Di Puskesmas	*Logistik Kesehatan*	Puskesmas	
2	Hanif Hidayatulloh, 2021	SEMINAR PEMANFAATAN CLOUD COMPUTING DI LINGKUNGAN MASYARAKAT	*Cloud Computing*	Lingkungan masyarakat	Metode Pemanfaatan Cloud Computing Di Lingkungan Masyarakat
3	Cahya Mulyadi, 2021	Rancang Bangun Web Aplikasi Menggunakan *Serverless Architecture* Dan Metode *Web Scraping* Sebagai Sumber Data	*Serverless*	Mahasiswa	Aplikasi yang menampilkan nilai mahasiswa secara informatif berdasarkan data

Daftar penelitian terkait akan mencakup studi-studi yang telah melakukan penelitian sebelumnya. Penelitian ini akan membahas metode, teknologi yang digunakan, hasil, dan temuan penting dari penelitian tersebut. Tabel perbandingan penelitian kemungkinan akan mencakup kolom-kolom seperti judul penelitian, metode yang digunakan, teknologi atau alat yang digunakan, hasil utama, dan kesimpulan. Dalam penjelasannya, akan dijelaskan bagaimana penelitian-penelitian tersebut relevan dengan penelitian yang sedang dilakukan oleh mereka dan bagaimana kontribusi mereka dalam memahami dan menyelesaikan permasalahan yang sama. Peneliti akan memaparkan dari ***Tabel 2.1*** sebagai berikut :

1. Manajemen Logistik Alat Kesehatan Di Puskesmas

Penelitian ini bertujuan untuk menganalisis manajemen logistik alat kesehatan di Puskesmas Boja II Kabupaten Kendal. Penelitian ini dilakukan dengan

menggunakan metode kualitatif yang meliputi wawancara mendalam, observasi, dan pengumpulan data dengan dokumen. Subjek penelitian ini adalah kepala Puskesmas Boja II, bendahara barang Puskesmas Boja II, 3 petugas pengurus barang Puskesmas Boja II, dan koordinator ruang balai pelayanan umum Puskesmas Boja II. Penelitian ini dilakukan di wilayah kerja Puskesmas Boja II yang meliputi 8 desa. Penelitian ini dilakukan dengan tujuan untuk memahami bagaimana manajemen logistik alat kesehatan di Puskesmas Boja II berpengaruh pada kualitas pelayanan kesehatan dan bagaimana sumber daya manusia yang terlibat dalam manajemen logistik alat kesehatan. Penelitian ini juga bertujuan untuk mengetahui bagaimana peraturan dan regulasi yang berlaku dalam pengelolaan alat kesehatan di Puskesmas Boja II.

2. SEMINAR PEMANFAATAN CLOUD COMPUTING DI LINGKUNGAN MASYARAKAT CIMANGGIS DENGAN AMAZON WEB SERVICES

Penelitian ini bertujuan untuk meningkatkan pengetahuan masyarakat Cimanggis tentang Cloud Computing dengan menggunakan Amazon Web Services (AWS) serta memanfaatkan fitur-fitur AWS dalam kegiatan bisnis. Metode yang digunakan dalam penelitian ini adalah seminar pengabdian masyarakat yang mencakup presentasi, pengajaran cara penggunaan AWS, dan sesi tanya jawab. Subjek penelitian adalah masyarakat Cimanggis yang berjumlah 14 peserta. Kegiatan ini diharapkan dapat menambah pemahaman peserta tentang cloud computing dan AWS, serta memotivasi mereka untuk lebih mengembangkan teknologi informasi dalam usaha mereka.

3. Rancang Bangun Web Aplikasi Menggunakan *Serverless Architecture* Dan Metode *Web Scraping* Sebagai Sumber Data

Penelitian ini bertujuan untuk merancang dan membangun sebuah aplikasi web menggunakan arsitektur serverless dan metode web scraping sebagai sumber data. Tujuannya adalah untuk membuat aplikasi yang dapat menampilkan nilai mahasiswa Politeknik Negeri Jakarta secara informatif. Metode yang digunakan dalam penelitian ini meliputi studi literatur, analisis kebutuhan sistem, pengembangan sistem menggunakan metodologi Rapid Application

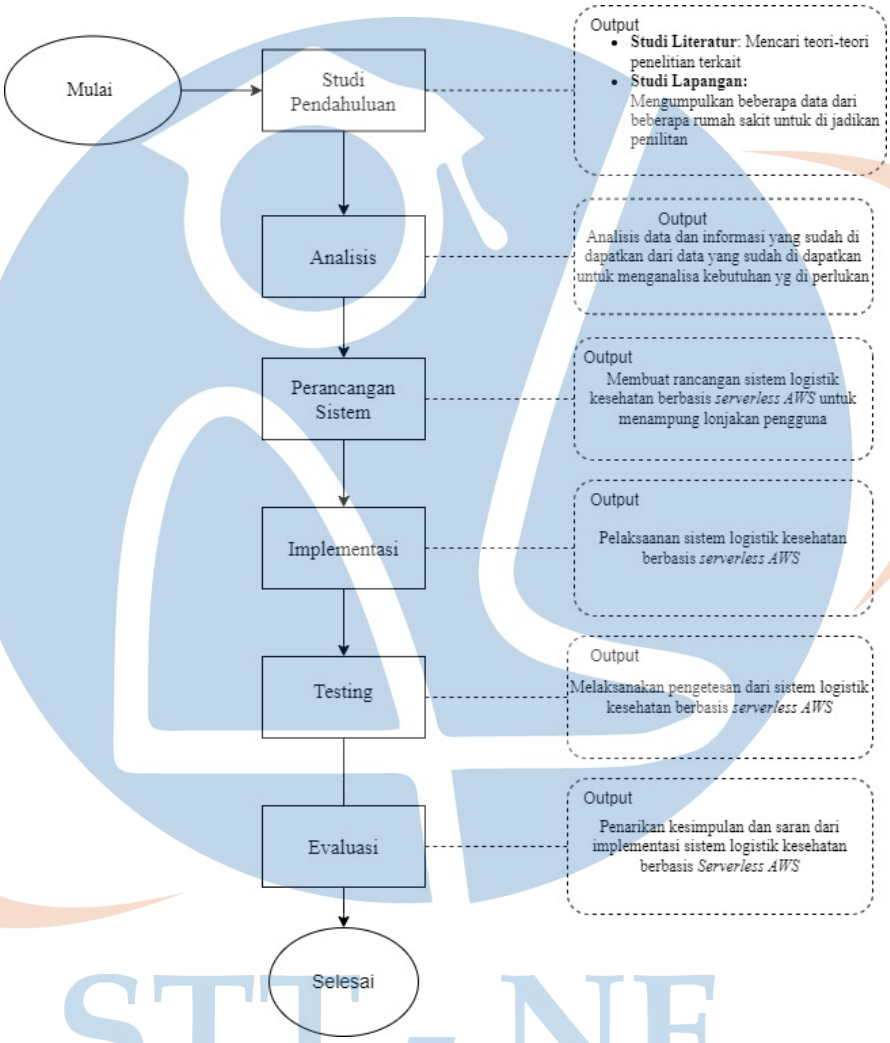
Development (RAD), desain dan pembuatan prototipe, pengujian, deployment, serta pemeliharaan aplikasi. Subjek penelitian adalah data nilai mahasiswa yang diperoleh melalui web scraping dari portal mahasiswa Politeknik Negeri Jakarta.

Keterkaitan penelitian tersebut dengan jurnal penelitian sebelum nya ialah peneliti melakukan sebuah perancangan sistem dengan menggunakan *tools* yang ada di layanan *AWS* terutama *AWS Console* yang bertujuan perancangan system yang peneliti rancang dapat me-monitoring *Database* ketika terjadi kelonjakan dan me-manajamen *Database* ketika terjadi kelonjakan pada waktu dan hari tertentu.

BAB III
METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Berikut adalah tahapan-tahapan yang akan dilaksanakan selama melaksanakan penelitian tersebut secara umum adalah sebagai berikut:



Gambar 3. 1 Tahapan Penelitian

Dari gambar diatas peniliti akan menjelaskan tahapan demi tahapan yang akan dilaksanakan oleh peneliti pada penelitian ini :

1. Studi Permasalahan

Pada tahap ini, peneliti mencari tau titik dari permasalahan dari penelitian ini yang diaman peneliti memahami permasalahan bagaimana kelonjakan

database itu terjadi pada suatu instansi kesehatan yang dimana menyebabkan pengurangan kecepatan membaca suatu data ketika mengimput suatu data kedalam *database* atau melebihi kapasitas *database*. Oleh dari itu peneliti mencari teori-teori yang menjadi pendukung pada penelitian tersebut beserta memikirkan solusi terbaik untuk permasalahan diatas. Peneliti melakukan studi lapangan yang dimana peniliti meminta izin kepada suatu instansi kesehatan/rumah sakit yang akan dijadikan sebagai penelitian ini.

2. Analisis

Pada tahap ini, peneliti sudah mendapatkan data dan informasi dari suatu instansi kesehatan/rumah sakit yang berupa data. Lalu peneliti akan menganalisa data tersebut untuk mencari solusi yang terbaik untuk memenuhi kebutuhan pada instansi kesehatan/rumah sakit tersebut yang dimulai dengan memahami data yang sudah di dapatkan dan membaca dari beberapa jurnal dan artikel ilmiah yang akan menjadi penunjang dari penetilian yang di laksakan.

3. Perancangan Sistem

Pada tahap ini, data yang sudah didapatkan peneliti dari instansi kesehatan/rumah sakit dan sudah dianalisis yang dimana peneliti akan melanjutkan dengan melakukan perancangan sebuah sistem yang harapan peneliti sistem ini akan menjadi solusi terbaik untuk permasalahan yang sedang dialami oleh instansi kesehatan/rumah sakit, baik dari pengerjaan *tools-tools* yang akan di gunakan pada perancangan sistem tersebut seprerti pembuatan *Amazon S3 Buckets* sebagai penyimpanan data yang sudah didapatkan, pembikinan *AWS Lambda Function*, pembikinan *Amazon API Gateway* yang dimana menggunakan tipe *Simple REST API*, pembuatan *Amazon DynamoDB* yang akan digunakan sebagai *Database* pada penelitian ini, pembuatan *AWS IAM Roles* yang akan digunakan sebagai aksess pada *AWS Lambda Function* untuk medapatkan *Command Access* dari *Amazon S3 Buckets* dapat mengirimkan data ke *Amazon DynamoDB*, pembikinan sebuah *script* yang dimana akan di gunakan pada penelitian ini untuk menunjukan apakah terjadi kerentanan pada *Amazon DynamoDB* yang sudah di buat oleh peneliti ,

pembuatan *AWS CloudWatch* yang akan digunakan sebagai monitoring pada *Amazon DynamoDB* yang sudah di buat oleh peneliti dan memberikan *Alert Trigger* apabila *Amazon DynamoDB* yang peneliti buat terjadi kerentanan baik dari segi kapasitas sudah melebihi batas atau mengalami penurunan performa pada DB tersebut.

4. Implementasi

   pada tahap ini, peneliti melakukan implementasi dari rancangan sistem yang sudah di buatkan dengan menggunakan *Script* yang sudah di buat oleh peneliti dengan menggunakan metode *Load Testing* yang dimana menggunakan sebuah script untuk pengetesan Database yang sudah dibikin oleh pengguna sebelum nya yang bertujuan apakah *Database* tersebut terjadi kerentanan dari yang sudah di sebutkan sebelum nya dan bagaimana *Database* tersebut menunjukan reaksi nya ketika di berikan sebuah script. Apakah terjadi kerentanan pada *Database* tersebut, apakah terjadi pengurangan performa baik dari kecepatan *Database* tersebut ketika membaca data atau kapasitas *Database* tersebut sudah melebihi kapasitas.

5. Evaluasi

   Pada tahap ini, setelah peneliti sudah merancang sebuah sistem dan melakukan implementasi dari sistem yang sudah dikerjakan, peneliti akan menyimpulkan hasil dari implementasi yang sudah dikerjakan , serta peneliti memberikan saran untuk kedepannnya penelitian ini perlu di kembangkan secara bertahap.

3.2 Rancangan Penelitian

Pada rancangan penelitian ini, peneliti memberikan penjabaran lebih detail mengenai tahapan-tahapan yang akan di lakukan pada penelitian ini, hal ini meliputi jenis penelitian, metode analisis data, metode pengumpulan data,metode pengujian, metode impelentasi dan evaluasi

3.2.1 Jenis Penelitian

Jenis Penelitian yang akan dilaksanakan oleh peneliti merupakan jenis penelitian dan pengembangan atau biasa di sebut dengan *Research and Development* (R&D). Dalam konteks penelitian tersebut , pengembangan produk digital atau perangkat lunak tersebut nantinya akan dikembangkan kembali secara bertahap yang dapat memberikan manfaat, terutama di sektor logistik kesehatan.

3.2.2 Metode Analisis Data

Metode analisis data yang akan di gunakan oleh peniliti pada penelitan ini ialah metode kuantitatif. Metode kuantitatif merupakan sebuah metode yang mendapatkan sebuah data yang dimana data tersebut berupa angka. Metode kuantitatif di pilih oleh peneliti dikarenakan data yang didapatkan oleh peneliti ialah database di sektor logistik kesehatan. Data akan di analisis dan akan di lakukan simulasi yang dimana akan dimasukan beberapa data dengan skala besar dengan menggunakan metode trigger *alert* . Metode trigger alert tersebut akan memberikan output berupa angka atau persentase. Angka atau persentase tersebut akan menjadi *threshold* atau batasan apakah database tersebut sedang terjadi lonjakan atau tidak terjadi lonjakan

3.2.3 Metode Pengumpulan Data

Metode pengumpulan data yang dilaksanakan yaitu dengan menggunakan metode eksperimen. Metode eksperimen adalah metode yang melakukan sebuah percobaan dengan tujuan membuktikan sendiri sesuatu yang di pelajari. Adapun dalam konteks database ialah pendekatan sebuah penelitian dengan tujuan untuk menguji pengaruh perlakuan tertentu terhadap suatu variabel lain dalam kondisi yang sudah terkendali. Dalam hal ini peneliti akan melakukan eksperiment terhadap database tersebut dengan memasukan data-data yang banyak untuk mengamati apakah database tersebut terjadi kelonjakan yang drastis atau tidak.

3.2.4 Metode Pengujian

Metode pengujian yang akan peneliti gunakan kepada penelitian tersebut ialah akan dilakukannya dengan menggunakan metode *Load Testing*. Pada pengujian ini peneliti akan melakukan pengujian terhadap database yang sudah di aplikasikan menggunakan layanan *Serverless AWS*. Pengujian ini bertujuan apakah database tersebut terjadi permasalahan pada kinerja database tersebut Ketika database di berikan beban yang tinggi. Hasil dari pengujuan ini akan menjadi titik acuan untuk pengembangan selanjutnya.

3.2.5 Metode Implementasi dan Evaluasi

Metode implementasi yang akan peneliti lakukan dalam penelitian ini yaitu menggunakan layanan *serverless AWS* dan database yang dimana dari kedua implementasi tersebut harapan peneliti ialah apakah database tersebut yang telah tersambung dengan layanan *Serverless AWS* tersebut sedang terjadi kelonjakan pengguna atau kelonjakan database, serta memberikan *alert* bahwasahnya database tersebut sedang terjadi kelonjakan, adapun evaluasi yang akan dilaksanakan oleh peneliti merupakan evaluasi dari hasil pengujian dengan menggunakan metode *Load Testing*, yang dimana peneliti akan membikin sebuah script yang akan di jalankan di dalam sistem yang sudah di buat dalam metode *Load Testing* untuk memastikan apakah hasil dari script tersebut terjadi kelonjakan atau tidak terjadi kelonjakan .

3.2.6 Lingkungan Pengembangan

Dalam penelitian dan pengembangan aplikasi ini, diperlukan software serta hardware untuk pengembangan perangkat lunak berbasis serverless . Software yang digunakan, yaitu *Visual Sudio Code*, dan *AWS Tools*. Selain membutuhkan *software*, penelitian ini juga membutuhkan hardware. Adapun hardware yang akan digunakan, yaitu laptop dengan spesifikasi sebagai berikut:

a. Model : Acer Nitro 5 AN515-45

b. Processor : AMD Ryzen 5 5600H

c. VGA : NVIDIA GTX 1650

d. RAM : 16 GB

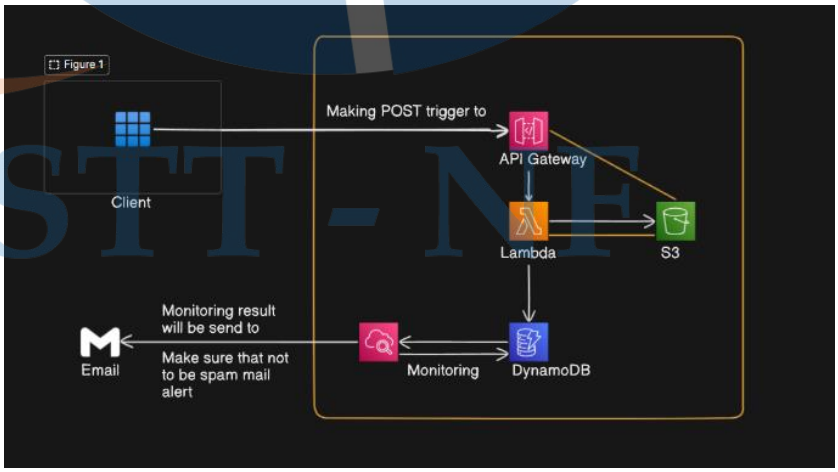
e. System Type : 64-bit windows operating system

BAB IV
IMPLEMENTASI DAN EVALUASI

Pada Bab IV ini peneliti akan menjelaskan tahapan-tahapan implementasi yang akan dikerjakan dan di lanjutkan dengan evaluasi dari implementasi dari sistem yang peneliti laksanakan pada penelitian tersebut.

4.1 Analisa Perancangan sistem

Pada tahap ini peneliti merencanakan tahapan perencanaan pada implementasi yang peneliti akan buat. Tahapan ini dimulai dengan memahami kinerja dari *tools-tools* yang akan terlibat dalam implementasi pada penelitian tersebut, yang bertujan untuk mendapatkan titik terang dari permasalahan yang terjadi pada sebuah instansi kesehatan. Persiapan tools yang akan di gunakan selama penelitian dimulai memahami kinerja dan kegunaan dari tools tersebut, serta memikirkan solusi terbaik untuk menyelesaikan permasalahan yang terjadi pada instansi kesehatan dan memberikan manfaat yang positif kepada intsansi kesehatan tersebut. Setelah peneliti menemukan solusi terbaik untuk permasalahan pada instansi tersebut, selanjut peneliti merancang sebuah diagram arsitektur yang bertujuan sebagai cetak biru dari rancangan implementasi sistem yang dilaksanakan pada penelitian ini:



Gambar 4. 1 Diagaram Arsitektur rancangan implementasi sistem yang dilaksanakan

Gambar 4.1 merupakan sebuah diagram arsitektur yang akan digunakan pada perancangan implementasi sistem yang peneliti laksanakan, untuk skema yang

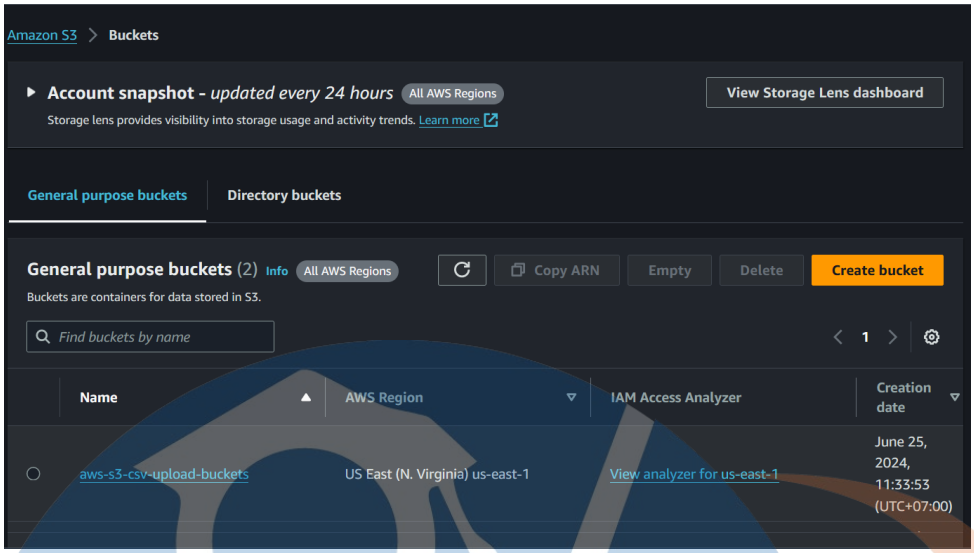
akan di laksanakan pada penelitian tersebut yaitu ketika instansi kesehatan akan melakukan pengimputan *data file* berbentuk *.csv* ke sebuah *website* yang sudah terkoneksi dengan sistem yang dirancang dengan menggunakan perantara *API Endpoint* dari *Amazon API Gateway*, lalu data tersebut pertama kali akan di simpan kedalam *Amazon S3 Bucket*. Setelah *data file* tersebut sudah tersimpan kedalam *Amazon S3 Bucket* selanjutnya *data file* tersebut akan di lanjutkan pengimputannya ke *Amazon DynamoDB* dengan menggunakan integerasi *function* dari *AWS Lambda Funtion* yang sudah mendapatkan aksess dari *AWS IAM Roles* untuk melakukan integerasi. Setelah *data file* sudah terinput kedalam *Amazon DynamoDB*, *Amazon CloudWatch* di rancang untuk memonitoring hasil dari penginputan *data file* dari *Amazon S3 Buckets* menuju *Amazon DynamoDB*. Apabila terjadi kelonjakan maka dari *Amazon DynamoDB* makan *Amazon CloudWatch* akan memberikan pemberitahuan melalui *E-mail* instansi kesehatan apabila sistem tersebut terjadi kelonjakan.

4.2 Implementasi Sistem

Pada tahapan ini peneliti akan menjelaskan tahapan-tahapan pengerjaan implementasi yang akan dilaksanakan pada penelitian ini dengan harapan implentasi tersebut memberikan hasil yang cukup berdasarkan *Gambar 4.1* diagram arsitektur yang peneliti rancang pada penelitan ini yang bertujuan peneliti tidak menggunakan tools-tools yang lain pada *AWS Console* yang berakibatkan perancangan sistem tersebut menjadi terhambat .

4.2.1 Perancangan *Amazon S3 Buckets*

Pada tahapan ini peneliti akan membuat sebuah *S3 Buckets* yang dimana akan di gunakan oleh peneliti untuk menyimpan dan mengupload database dalam bentuk `.csv` yang peneliti beri nama dengan *s3-dynamodb-buckets* .



Gambar 4. 2 Amazon S3 Buckets

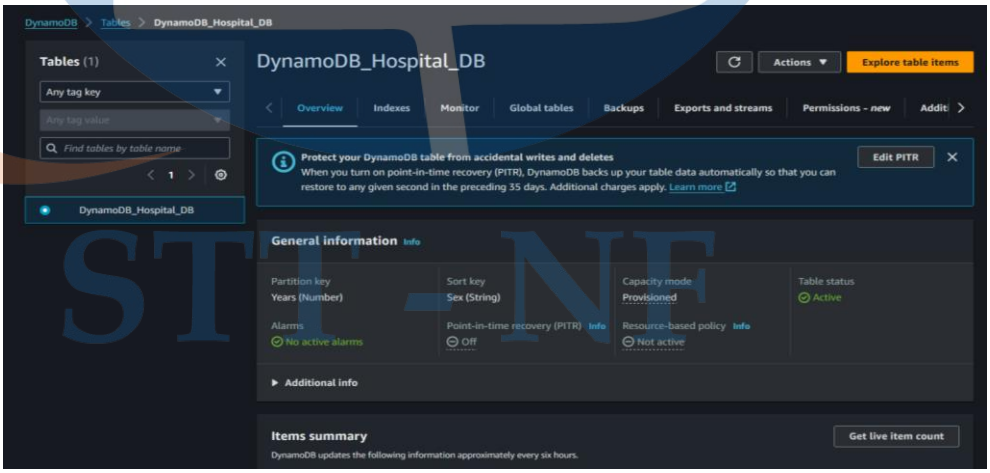
Pada *Gambar 4.2* merupakan sebuah *Amazon S3 Buckets* yang sudah di buat oleh peneliti dan akan digunakan untuk melakukan simulasi implementasi sistem dengan menggunakan metode *Load Testing* pada penenilitan ini yang dimana berisikan *data file* berbentuk *.csv*. Adapun *permision source code* untuk memastikan bahwa *S3 Bucket* tersebut bisa di akses peneliti menuliskan *source code* dalam *S3 Bucket* dengan tujuan *S3 Buckets* yang sudah di buat dapat diakses oleh *AWS Lambda Function* seperti berikut:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowLambdaAccess",
            "Effect": "Allow",
            "Principal": {
                "AWS":
"arn:aws:iam::665621628484:role/LambdaExecutionrolescode"
            },
            "Action": [
                "s3:GetObject",
                "s3:ListBucket",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::aws-s3-csv-upload-buckets",
                "arn:aws:s3:::aws-s3-csv-upload-buckets/*"
            ]
        }
    ]
}}
```

Pada sourcode yang disampaikan diatas merupakan source code yang di rancang oleh peneliti untuk memberikan akses dari *Amazon S3 buckets* kepada *AWS Lambda Function* dan *Amazon DynamoDB* untuk melakukan integerasi yaitu melakukan pengupload-tan *database file* dalam bentuk *.csv* yang di upload pertama kali kedalam *Amazon S3 Buckets* lalu di lanjutkan proses menguploadnya ke *Amazon DynamoDB* melalui integerasi *AWS Lambda Function*.

4.2.2 Perancangan *Amazon DynamoDB*

Pada tahap ini, peneliti membuat *Amazon DynamoDB* yang akan di gunakan oleh peneliti sebagai *Database* (*DB*) yang akan menampung data dalam bentuk `.csv` dari *Amazon S3 Buckets* , yang dimana *DynamoDB* tersebut sudah dikonfigurasi oleh peneliti dapat melakukan *Autoscale* pada *DynamoDB* yang bertujuan untuk menanggulani pengimputan data apabila terjadi permasalahan ketika pengirman data dalam bentuk `.csv` dari *Amazon S3 Buckets* ke *DynamoDB* untuk dilakukan pengetesan kepada *DynamoDB Table* yang sudah di buat oleh peneliti untuk dilakukan pengujian pada penelitian tersebut seperti *Gambar 4.3* dibawah ini.



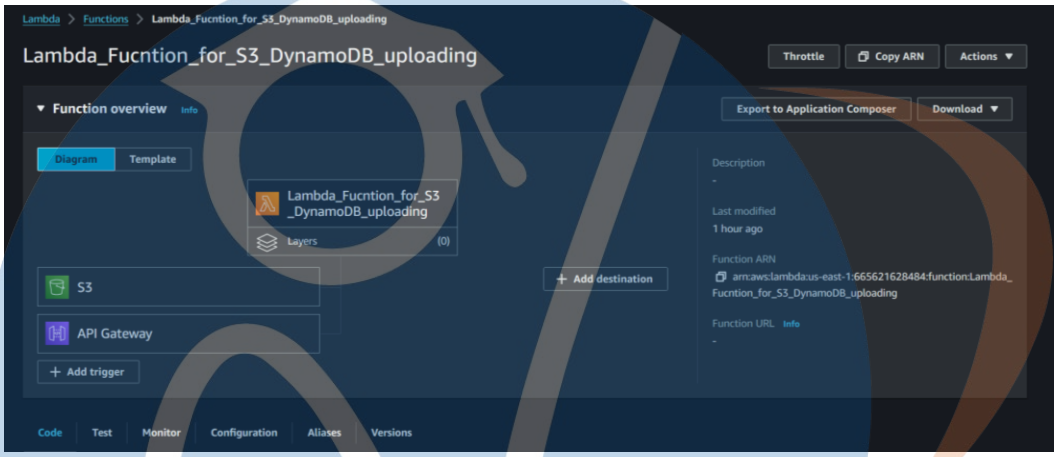
Gambar 4. 3 Amazon DynamoDB Table Database

Adapun *Amazon DynamoDB* yang dirancang oleh peneliti akan menjadi sebuah *Database* dan tempat pengimputan terakhir dari pengimputan

pertama *data file .csv* pertama kali kedalam *Amazon S3 Buckets* dengan menggunakan perantara oleh salah satu *tools* yang tersedia di *AWS Console.*

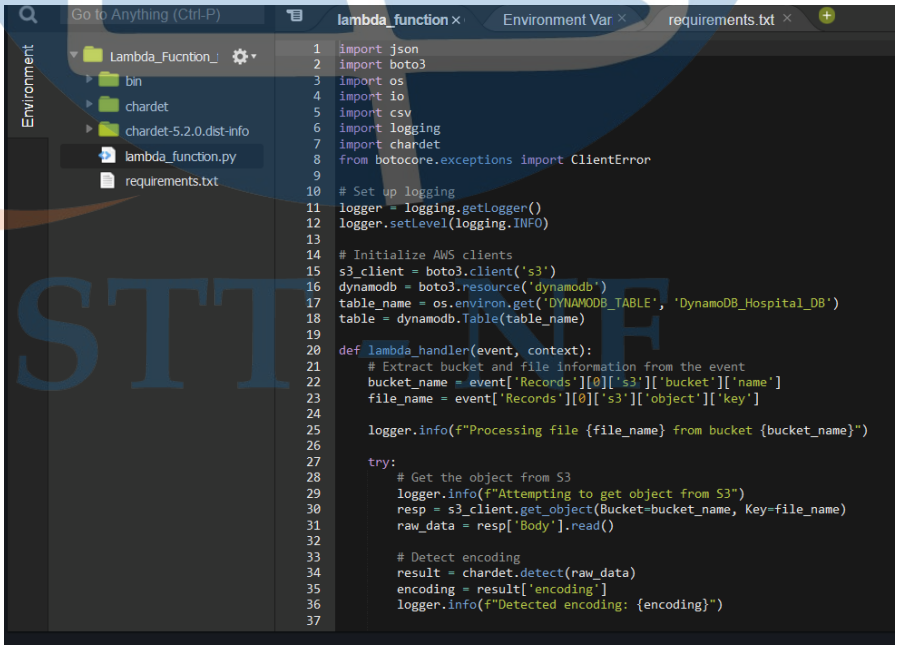
4.2.3 Perancangan *AWS Lambda Function*

Pada tahap ini peneliti membikin sebuah *Lambda Function* yang akan digunakan sebagai perintah command dimana data *.csv* yang sudah terinput pertama kali di *Amazon S3 Buckets* akan dilanjutkan ke *DynamoDB Database* yang sudah peneliti rancang sebelum nya pada *Gambar 4.4* Sebagai berikut.

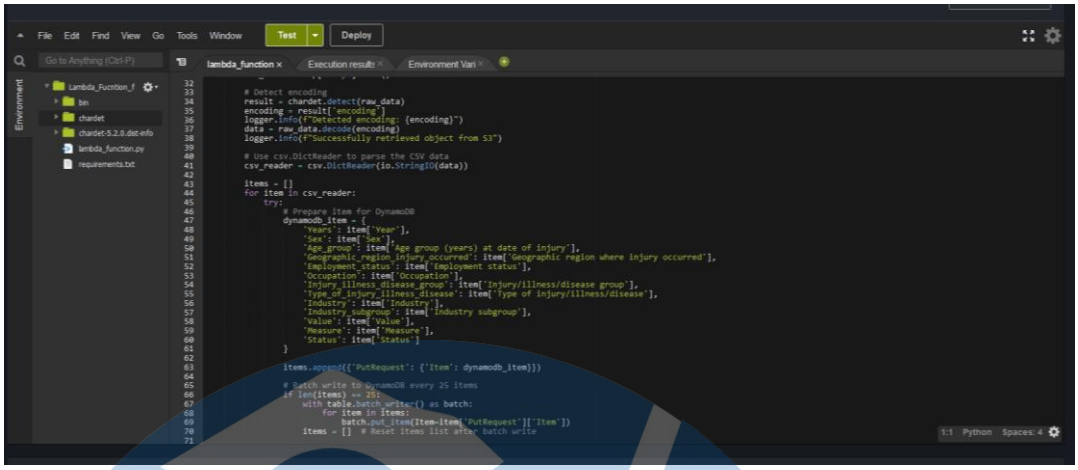


Gambar 4. 4 AWS Lambda Function

Adapun *Source code* yang ada pada *AWS Lambda Fucntion* adalah sebagai berikut:



Gambar 4. 5 Potongan Source code dari AWS Lambda Function

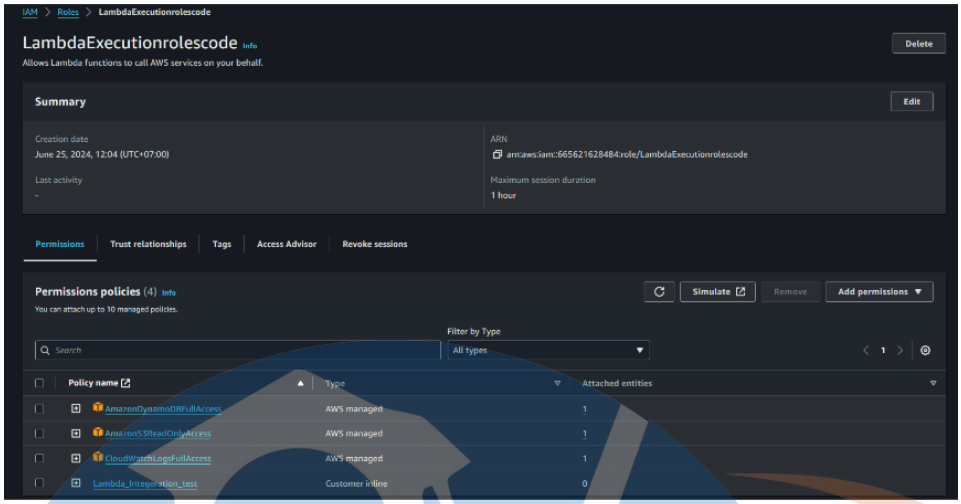


Gambar 4. 6 Lanjutan Source Code dari AWS Lambda Function

Pada *Gambar 4.5* dan *Gambar 4.6* yang merupakan sebuah kode yang peneliti buat pada *AWS Lambda Function* akan digunakan pada penelitian ini yang bertujuan untuk melaksanakan command pengimputan data yang ada di *Amazon S3 Buckets* menuju *DynamoDB Table* yang bertujuan *Amazon S3 Buckets* dapat mengimput *data file .csv* yang sudah tersimpan di *Amazon S3 Buckets* ke *DynamoDB Table* untuk dilakukan pengujian pada *DyanamoDB Table* dengan menggunakan integerasi *source code* yang sudah peneliti rancang pada perancangan ini.

4.2.4 Perancangan *AWS IAM*

Pada tahap ini ketika peneliti sudah membikin rancangan *AWS Lambda* yang dimana akan digunakan sebagai perintah untuk melakukan pengimputan data dalam bentuk `.csv` dari *Amazon S3 Buckets* kepada *DynamoDB*, adapun kegunaan *AWS IAM* pada ada perancangan ini iala memberikan akess kepada *AWS lambda* untuk melakukan konfigurasi yang akan dilaksanakan.

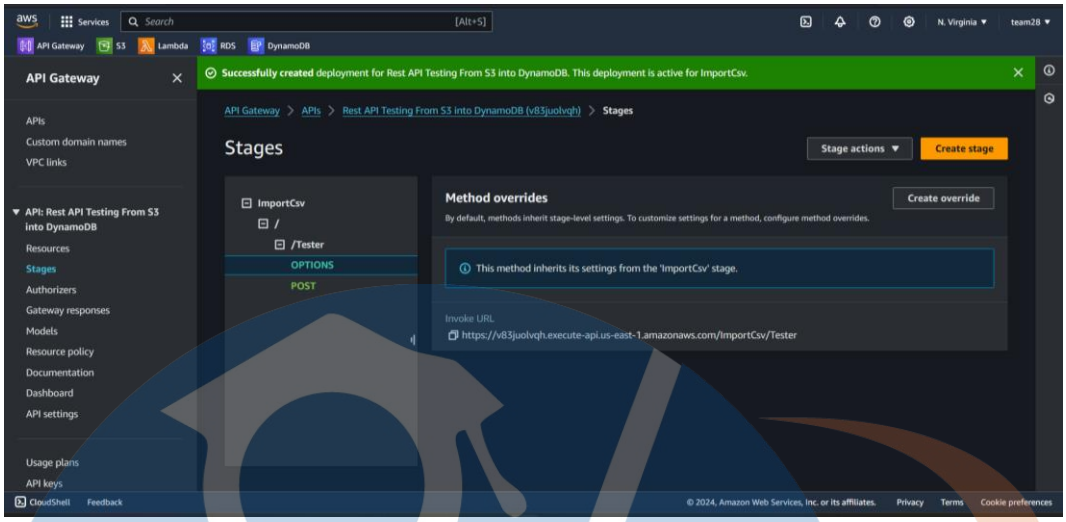


Gambar 4. 7 Pemberian aksess dari AWS IAM kepada AWS Lamba untuk melakukan konfigurasi

Adapun peran terpenting yang dilakukan *AWS IAM* yang tertera pada *Gambar 4.7* ini ialah *AWS IAM* memberikan sebuah aksess kepada *AWS Lambda Function* dengan memasukan hasil konfigurasi tersebut kedalam *AWS Lambda Function* yang dituju dalam bentuk *AWS IAM Roles* yang bertujuan dapat melakukan eksekusi dari *AWS Lambda Function code* yang sudah peneliti rancang pada penelitian ini. Terdapat akes "*AmazonDynamoDBFullAccess"* yang dimana *AWS lambda* diberikan akess full terhadap integerasi terhadap *DynamoDB*, "*AmazonS3ReadOnlyAccess*", dan "*CloudWatchLogsFullAccess"* yang dimana *AWS Lambda* hanya diberikan aksess untuk membaca data yang masuk.

4.2.5 Perancangan Amazon API Gateway

Di tahap ini , peneliti merancang sebuah *API Gateway* pada *AWS API Gateway* yang akan di *deploy* menggunakan metode *Simple REST API Gateway* dimana *Amazon API Gateway* tersebut hanya memiliki satu *trigger endpoint* untuk pelaksanaan penelitian ini yaitu hanya menggunakan *trigger* "*POST*" dan pada *AWS API Gateway* yang bertujuan untuk membantu pengiriman *data file* dalam bentuk *.csv* dari *Amazon S3 Buckets* menuju ke *Amazon DynamoDB* dengan proses konfigurasi *command* (integrasi *source* code) dari *AWS Lambda Function* yang peneliti rancang. Adapun rancangan *Amazon API Gateway* tertera pada *Gambar 4.8* seperti berikut :

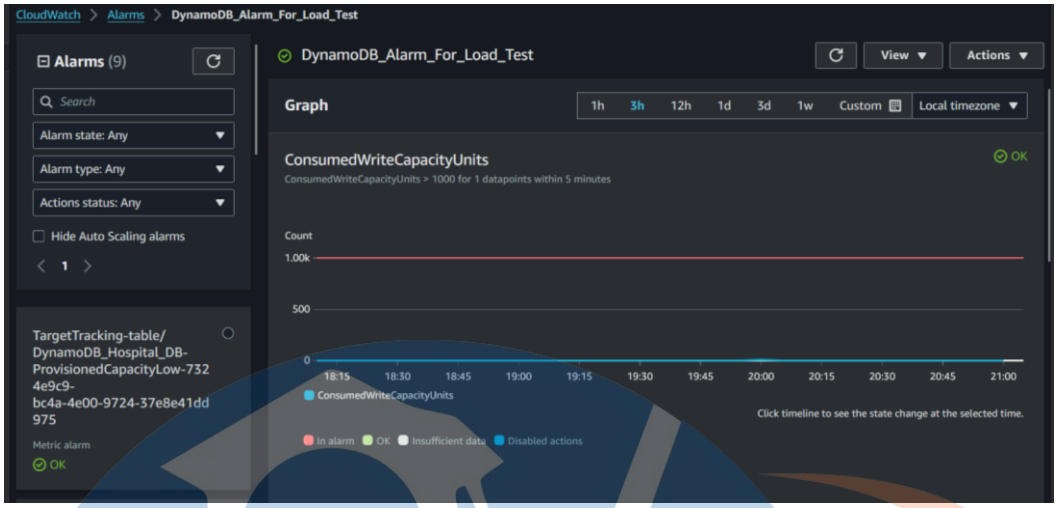


Gambar 4. 8 AWS API Gateway yang sudah diaktifkan deploy API

Pada rancangan ini *Amazon API Gateway* dengan menggunakann metode *Simple REST API Gateway* yang memiliki satu *trigger endpoint* yaitu *trigger "POST* berkerja sebagi *support*(pendunkung) untuk *AWS Lambda Funtion* akan melakukan proses konfigurasi *command* (integerasi *source code*) yang akan dijalankan

4.2.6 Perancangan Amazon Cloudwatch

Pada tahapan ini, setelah peneliti merancang *tools* yang akan digunakan pada penelitian ini tiba pada tahapan merancang *Amazon Cloudwatch* sebagi memonitoring *database*, terutama *DynamoDB* dengan tujuan *Amazon CloudWatch* dapat memantau dan memberikan *Alert Trigger* apabila *DynamoDB* tersebut telah terjadi kerentanan baik dari segi performa yang menurun atau sudah melebihi kapasitas, maka *Amazon CloudWatch* akan memberikan peringatan tersebut yang dikirimkan melalui *E-Mail* yang telah terdaftar.



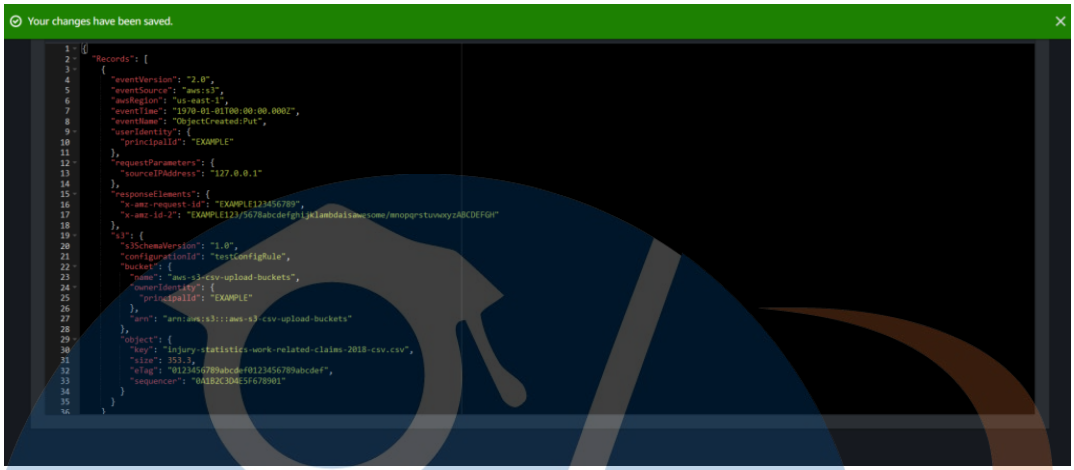
Gambar 4. 9 Amazon CloudWatch

Pada *Gambar 4.9 Amazon CloudWatch* berperan sebagai monitor apabila *DynamoDB* yang sudah di buat oleh peneliti terjadi kelonjakan yang dimana garis merah menunjukan batas akhir dari pengimputan data yang akan dilaksanakan. Apabila terjadi kelonjakan selama pengimputan dan telah melebihi batas yang sudah di tentukan oleh peneliti, makan *Amazon CloudWatch* akan memberikan peringatan (*Alert*) yang akan di kirimkan melalui Email yang sudah dicantumkan oleh peneliti

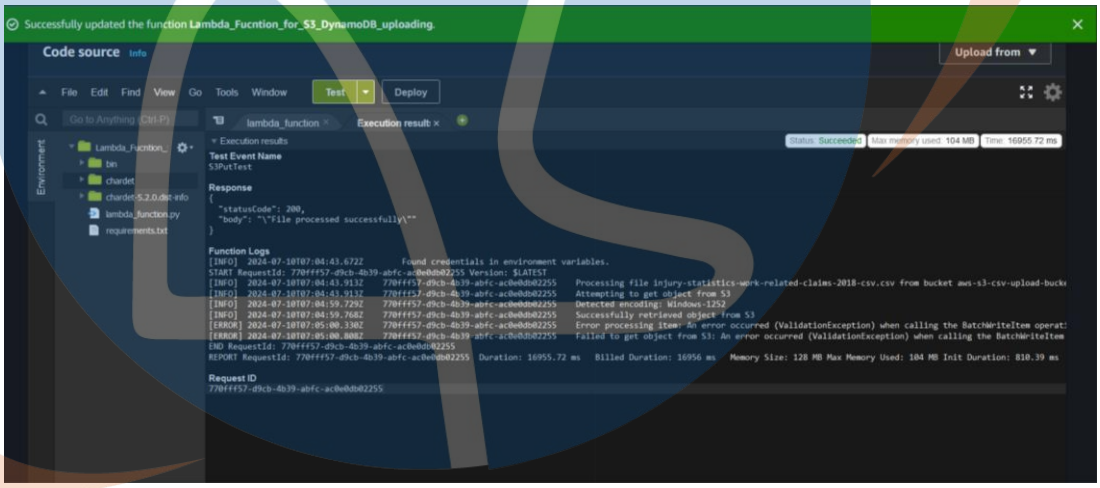
4.2.7 Pengujian dengan menggunakan metode Load Testing

Pada tahap ini, ketika peneliti sudah merancang semua *Tools* yang akan dilakukan pada penelitan tersebut, peneliti melanjutkan tahapan ini dengan membikin sebuah *Python Script* yang digunakan untuk melakukan simulasi pengimputan suatu data yang memili beban yang berlebih dengan menggunakan metode *Load Testing* yang bertujuan apakah *Database* tersebut masih melaksanakan kinerja nya dengan baik walaupun *Database* tersebut telah dimasukan *data* yang memiliki bobot yang berlebih atau tidak. Setelah dari kita melakukan pengimputan, peneliti melanjutkan untuk mengecek *Amazon CloudWatch* untuk memantau apakah *Database* tersebut terjadi kelonjakan yang seknifikan atau tidak. Dari hal yang telah disampaikan , peneliti dapat memberikan evaluasi berdasarkan implementasi yang telah dilaksanakan oleh peneliti pada penelitan ini. Sebelum penelitian tersebut dimulai peneliti melakukan sebuah test kepada *source code* yang sudah di buat pada *AWS*

Lambda Function untuk memastikan apakah *source code* tersebut berjalan lancar sebelum di masukan *scirpt*. Adapun *test event* dengan menggunakan *source code* seperti berikut :



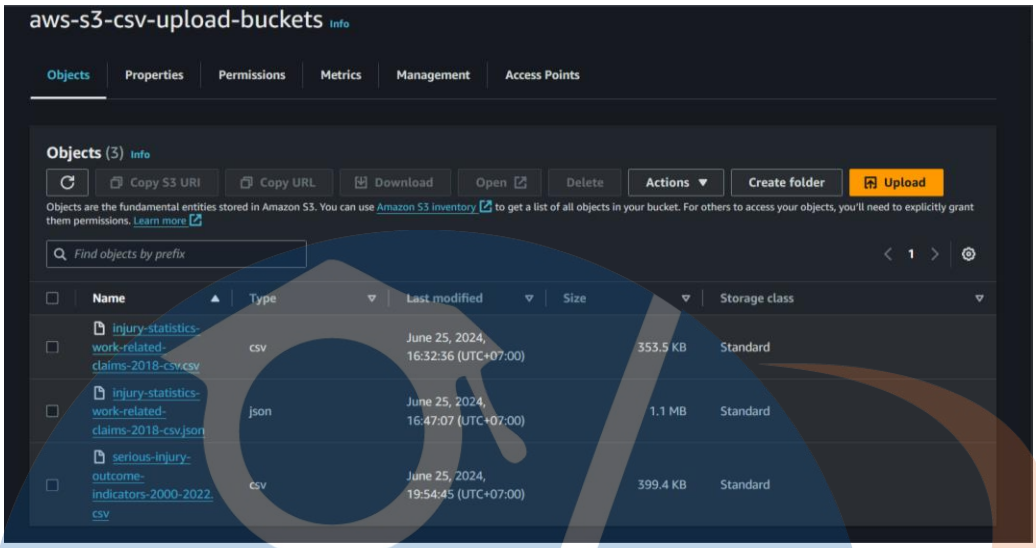
Gambar 4. 10 Lambda Test Event Code



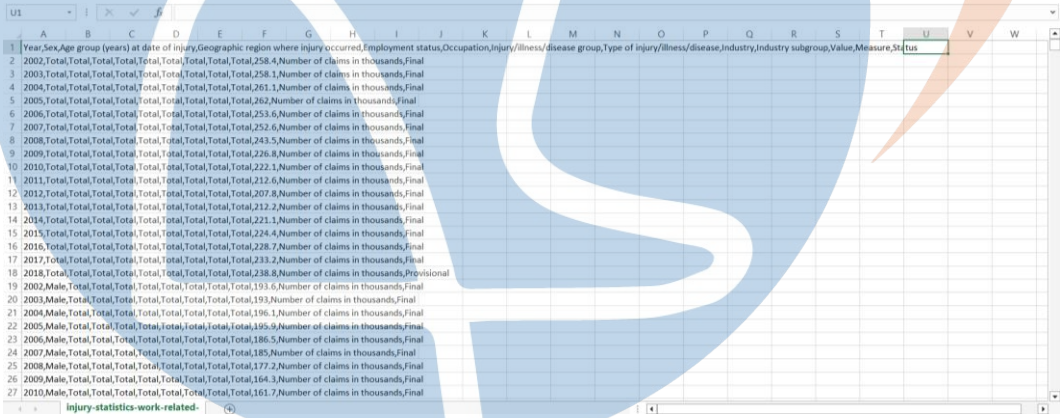
Gambar 4. 11 Hasil Dari Lambda Test Event Code

Berdasarkan dari *Gambar 4.10 dan Gambar 4.11* merupakan pengujian *source code* dari *AWS Lambda Funtion* menggunakan metode *Test Even* pada *AWS Lambda Funtion* untuk memastikan apakah *source code* tersebut telah berjalan sebelum dilakukan pengujian menggunakan metode *Load Testing*. Setelah tahapan tersebut sudah di laksanakan oleh peneliti, langkah selanjutnya yaitu percobaan sitem yang telah dirancang dengan menggunakan *AWS Tools* dalam bentuk *Serverless*, yang dimulai dengan mengupload *data file* yang berbentuk *.csv* yang peneliti beri nama "*injury-statistics-work-related-claims-2018-csv*.csv" sebagai penginputan pertama kedalam *Amazon S3 Buckets* kemudian

dilanjutkan ke *DynamoDB Tables*, adapun bentuk data file berbentuk .csv yang akan di jelaskan pada *Gambar 4.11* dan *Gambar 4.12* seperti berikut ini:



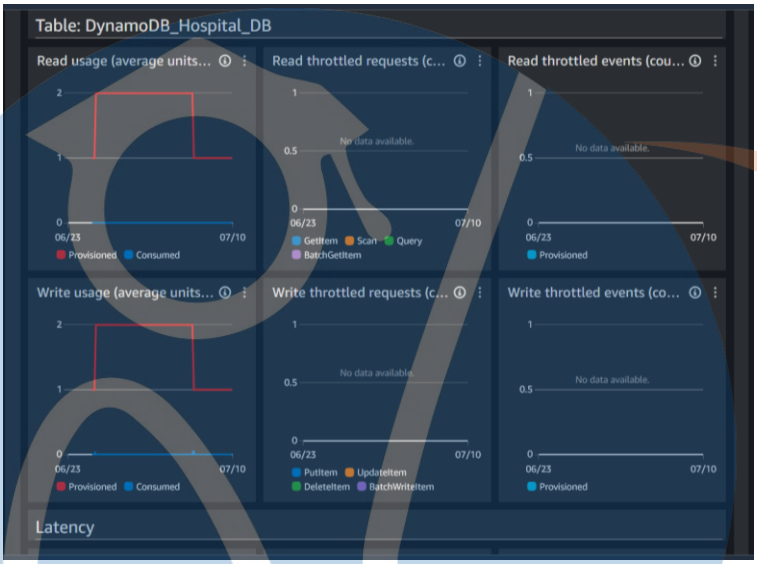
Gambar 4. 12 Data .csv yang sudah diupload kedalam Amazaon S3 Buckets



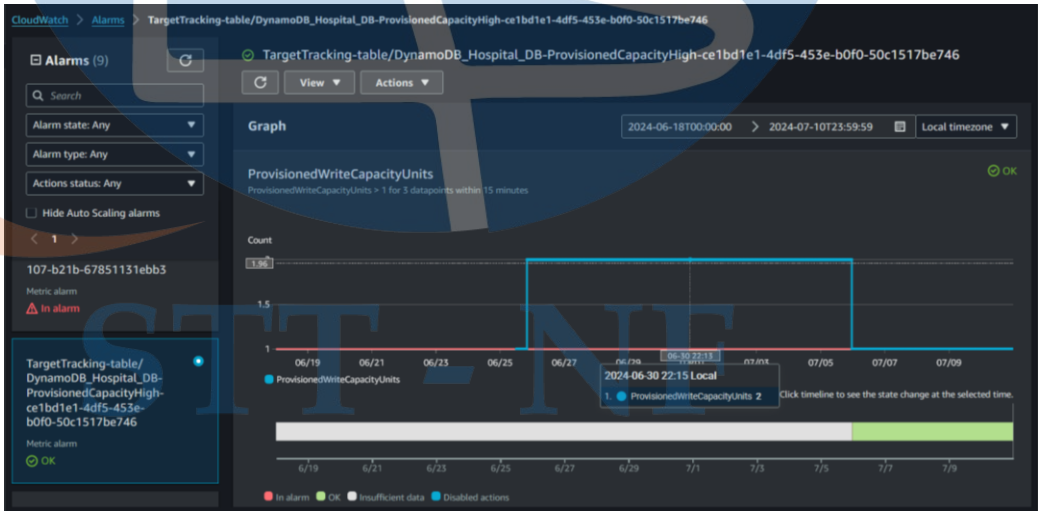
Gambar 4. 13 Data dalam bentuk .csv yang akan di ujikan

Dari *Gambar 4.12* dan *Gambar 4.13* di jelaskan bahwa data file yang berbentuk *.csv* akan peneliti input kedalam Amazon S3 Buckets tanpa mengganti isi dari *data file .csv* tersebut guna untuk penelitian ini, setelah *data file .csv* sudah terinput kedalam *Amazon S3 Buckets*, langka selanjutnya ialah *data file .csv* tersebut akan dilanjutkan proses penginputannya kedalam *Amazon DynamoDB Table* dengan menggunakan integrasi *AWS lambda* yang sudah di rancang sebelum nya . Tahap selanjutnya peneliti melakukan proses eksekusi pada sistem implementasi yang sudah dirancang dengan menjalakan *source code* yang sudah dirancang rancang pada *AWS Lambda Function* secara berkala dengan *time out*

yang sudah di tetapkan yaitu 30 detik setiap proses eksekusi *source code* tesebut. Setelah dilakukan proses pengetesan kepada sistem yang sudah di rancang oleh peneliti secara bekala dengan *time out* yang sudah ditetapkan oleh peneliti terbentuklah sebuah pembuktian dari pengetesan ini yang dimana sistem database yang rancang oleh peneliti telah terjadi peningkatan yang tersedia dalam bentuk diagram seperti berikut ini.



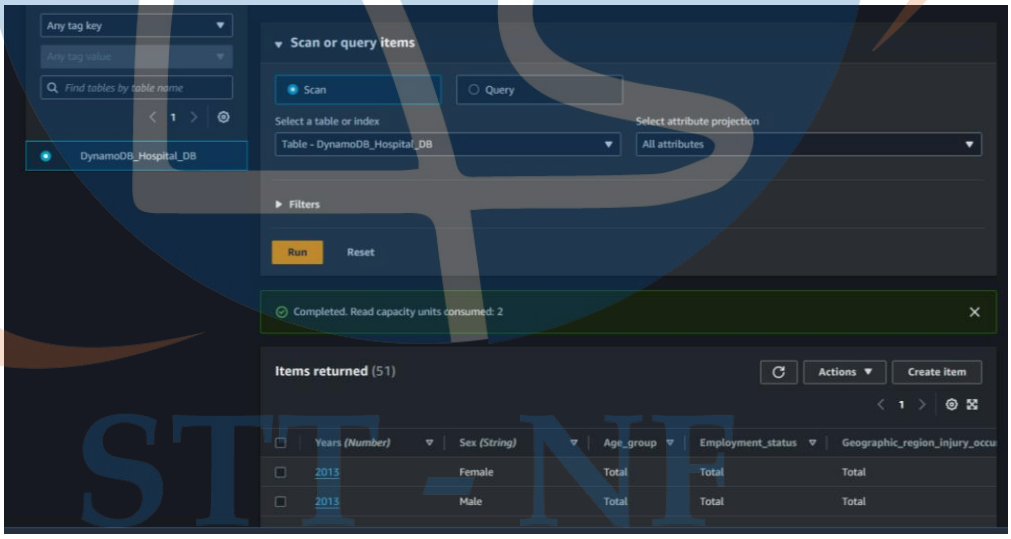
Gambar 4. 14 Diagram Hasil Pengujian Dari DynamoDB Table Setelah Dilakukan Pengetesan Secara Berkala



Gambar 4. 15 Diagram Hasil Monitoring Dari Amazon CloudWatch Setelah Dilakukan pengetesan kepada DynamoDB Secara Berkala

Terlihat pada pada *Gambar 4.14* dan *Gambar 4.15* merupakan diagram grafik hasil dari pengetesan yang dilakukan oleh peneliti secara bertahap dengan *time out* yang sudah di tentukan, dimana hasil dari pengetesan tersebut berupa

sebuah diagram grafik pada *Amazon CloudWatch* dan *Amazon DynamoDB* menunjukan bahwa *Amazon DynamoDB Table* yang telah dirancang oleh peneliti telah terjadi peningkatan setelah dilakukan pengetesan dengan menggunakan metode *load testing* dan *time out* yang sudah ditentukan oleh peneliti yang dilaksanakan secara berkala untuk memastikan bahwa sistem berbasis *serverless AWS* yang telah dirancang oleh peneliti telah terjadi peninggakan. Dimana terlihat pada diagram grafik *Amazon CloudWatch* diatas garis berwarna merah merupakan batas data yang terbaca dan garis berwarna biru merupakan jumlah data yang terbaca ketika proses pengimputan tersebut belangsung. Di sisi lain diagram grafik yang ada di *Amazon DynamoDB* merupakan hasil rekam pengimputan yang dilakukan secara bertahap diamana garis berwarna merah ialah hasil penginputan yang telah di lakukan sebelumnya dan garis berwana biru merupakan data yang baru terinput dari *data file .csv* yang ada di *Amazon S3 Buckets* ke *Amazon DynamoDB* dengan menggunakan integrasi *AWS Lambda Function* dan *Amazon API Gateway* sebagai *support* (pendukung).



Gambar 4. 16 Hasil data yang sudah terinput kedalam Amazon DynamoDB.

Pada *Gambar 4.16* ialah hasil dari pengetesan yang sudah terinput dari *Amazon DynamoDB* dengan menggunakan metode load testing dan time out yang sudah ditentukan berjalan dengan lancar , terdapat data yang masuk dengan total 51 data yang telah terinput dari *data file .csv* yang berada didalam *Amazon S3 Buckets*. Namun hasil dari pengetesan ini masih jauh dari harapan peneliti,

dimana peneliti berharap data yang terinput kedalam *Amazon DynamoDB Table* ketika pelaksanaan nya data bisa terinput dengan total minimal 150 data dari *data file .csv* yang berada di dalam *Amazon S3 Buckets* ke *Amazon DynamoDB Table* dengan menggunakan metode *load testing* dan *time out* yang sudah di lakukan. Adapun sistem yang sudah di rancang oleh peneliti dengan menggunakan terknologi *serverless* di bawah fondasi *AWS* menggunakan metode *load testing* sebagai pengetesan sistem tersebut telah berjalan sesuai dengan harapan peneliti yang dimana kinerja dari sistem tersebut setelah terinput sebuah *data file .csv* dengan memiliki beban yang berlebih bisa melakukan kinerja yang maksimal .

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang bisa diambil oleh peneliti pada penelitan yang berjudul "IMPLEMENTASI SISTEM LOGISTIK YANG DAPAT MENAMPUNG LONJAKAN PADA INSTANSI KESEHATAN DENGAN MEMANFAATKAN TEKNOLOGI *SERVERLESS* PADA LAYANAN *AWS SERVICES*" adalah sebagai berikut:

1. Perancangan sistem *database* menggunakan teknologi *Serverless* dibawah layanan *AWS Services* melibatkan berbagai *tools* yang disediakan dalam layanan *AWS Console* ketika penelitan berlangsung. Yaitu melibatkan *Amazon S3 Buckets* sebagai wadah untuk penginputan data yang berbentuk *.csv* yang akan dilanjutkan proses penginputan ke *Amazon DynamoDB* untuk dilakukan pengetesan dengan menggunakan integrasi dari *AWS Lambda Funtion* yang telah mendapatkan izin untuk melaksanakn integrasi dari *AWS IAM* , *Amazon API Gateway* berperan sebagai pendukung dari integrasi *AWS Lambda Function*, serta *Amazon CloudWatch* berperan sebagai *monitoring* atau memantau data ketika telah terbaca kedalam *Amazon DynamoDB* setelah pelaksanaan implementsi sistem dengan menggunakan metode *Load Testing* dengan memasukan *data file* kedalam *database* dimana memiliki skala beban yang sangat besar.

2. Dari hasil penelitian implementasi sistem berbasis *Serverless* berbasis *AWS Services* menggunakan metode *Load Testing* sebagai pengetesan pada implementasi sistem yang dilakukan pada penelitian ini menunjukan hasil persentase dalam bentuk diagram grafik bahwasanya sistem yang dirancang mampu berjalan ketika telah terinput *data file .csv* yang mempunyai skala beban yang besar dan sistem berjalan sesuai dengan harapan, akan tetapi hasil data yang telah terinput kedalam sistem ketika implementasi dengan menggunakan metode pengujian *Load Testing* berlangsung sebesar 51 data yang terinput, hal ini masih jauh dari target data yang terinput yang

diharapkan oleh peneliti yaitu minimal 150 data yang terinput kedalam sistem. Dapat disimpulkan bahwasanya persentase keberhasilan implementasi sistem dengan menggunakan metode *Load Testing* sebagai metode pengujian kepada sistem tersebut ialah 75%, hal tersebut perlu adanya penelitian lebih lanjut terkait sistem agar sistem tersebut berjalan 100% dengan baik

5.2 Saran

Berdasarkan dari hasi penelitian yang sudah dilaksanakan oleh peneliti maka peneliti akan memberikan saran kepada peneliti selanjutnya adalah sebagai berikut:

1. Database merupakan sebuah sistem yang penting pada suatu instansi, terutama instansi kesehatan, karena database pada instansi kesehatan memiliki jenis data yang beragam seperti data pasien, data alat-alat kesehatan(logistik kesehatan), data obat-obatan dan masih banyak lagi. Apabila terjadi kelonjakan bagaimana solusi untuk mengatasi hal tersebut. Saran yang akan diberikan kepada peneliti selanjutnya terhadap "implementasi sistem logistik yang dapat menampung kelonjakan pada instansi kesehatan dengan memanfaatkan teknologi *Serverless* pada layanan *AWS Services"* agar hasil yang sesuai yang diharapkan sebaiknya peneliti selanjutnya untuk mempelajari dan memahami secara mendalam dari berbagai referensi dalam bentuk artikel atau jurnal yang berkaitan dengan implementasi sistem logistik yang dapat menampung kelonjakan baik pada instansi-instansi lainnya juga instansi kesehatan dengan memanfaatkan teknologi *Serverless* pada layanan *AWS Services* supaya memberikan hasil penelitian yang lebih baik dan terperinci.

2. Diharapkan juga kepada peneliti selanjutnya untuk mempersiapkan untuk lebih teliti dan tekun dalam proses pencarian dan pengumpulan data agar penelitian ini sesuai dengan rancangan rencana dapat berjalan dengan lancar dan tanpa hambatan. Disaran juga peneliti selanjutnya untuk melakukan wawancara kepada narasumber dari suatu instansi kesehatan untuk membahas implementasi sistem logistik yang dapat menampung kelonjakan dengan memanfaatkan

teknologi *Serverless* pada layanan *AWS Services* sehingga dapat memberikan gambaran dan arahan yang memadai untuk melaksanakan penelitian.

3. Saran untuk peneliti selanjutnya yang tertarik dengan penelitian ini ialah untuk mengkaji lebih dalam terkait penelitan yang berjudul " implementasi sistem logistik yang dapat menampung kelonjakan pada instansi kesehatan dengan memanfaatkan teknologi *Serverless* pada layanan *AWS Services*" untuk melakukan pengembangan lebih lanjut , dikarenakan hasil dari penelitian ini belum sesuai harapan dari peneliti sebelum nya yang dimana sistem yang dirancang berjalan dengan lancar, namun hasil yang di dapatkan belum sesuai dengan target yang di harapkan, dimana harapan dari peneliti sebelum nya sistem yang sudah di rancang, dimana bisa membaca data diatas 150 data dalam *file .csv* dalam satu kali imputan namun, ketika pelaksanaan sistem nya hanya terbaca sekitar 51 data. Dapat ditekankan untuk peneliti selanjut nya untuk melakukan pengembangan lebih lanjut terkait penelitian ini

DAFTAR PUSTAKA

[1] E. Raza, L. O. Sabaruddin, and A. L. Komala, "Manfaat dan Dampak Digitalisasi Logistik di Era Industri 4.0," *Jurnal Logistik Indonesia*, vol. 4, no. 1, pp. 49–63, Oct. 2020, doi: 10.31334/logistik.v4i1.873.

[2] Wiryany, D., Natasha, S., & Kurniawan, R. (2022). Perkembangan Teknologi Informasi dan Komunikasi terhadap Perubahan Sistem Komunikasi Indonesia. *Jurnal Nomosleca*, *8*(2), 242-252.

[3] R. A. PAMUNGKAS, S. RAHMAYATI, and R. FIRMANSYAH, "Analisis Sistem Informasi Manajemen dalam Penggunaan Aplikasi Blibli," *JEMSI (Jurnal Ekonomi, Manajemen, dan Akuntansi)*, vol. 7, no. 1, pp. 33–41, Feb. 2021, doi: 10.35870/jemsi.v7i1.525.

[4] S. Dwiyatno, E. Rakhmat, and S. Christina, "PERANCANGAN PRIVATE CLOUD BERBASIS INFRASTRUCTURE AS A SERVICE," vol. 8, no. 2, 2021.

[5] H. Shafiei, A. Khonsari, and P. Mousavi, "Serverless Computing: A Survey of Opportunities, Challenges, and Applications," *ACM Comput Surv*, vol. 54, no. 11s, Nov. 2022, doi: 10.1145/3510611.

[6] J. Elektronika and D. Komputer, "PENERAPAN SERVERLESS COMPUTING DALAM MENDETEKSI PENYAKIT MULUT DENGAN METODE CNN," vol. 16, no. 2, pp. 230–238, 2023, doi: 10.51903/elkom.v16i2.1006.

[7] What Is Amazon S3, diakses pada 9 Juni 2024 (online) dari : https://docs.aws.amazon.com/id_id/AmazonS3/latest/userguide/Welcome.html

[8] What is AWS Lambda, diakses pada 30 maret 2024 dari: https://docs.aws.amazon.com/id_id/lambda/latest/dg/welcome.html

[9] What Is Amazon API Gateway, diakses pada 30 Maret 2024(online) dari: https://docs.aws.amazon.com/id_id/apigateway/latest/developerguide/welcome.html

[10] What Is Amazon Cloudwatch,diakses pada 30 Maret 2024 (online) dari: https://docs.aws.amazon.com/id_id/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html

[11] What Is DynamoDB, diakses pada 30 Maret 2024 (online) dari: https://aws.amazon.com/dynamodb/

[12] What is IAM ?, diakses pada 23 juni 2024 (online) dari : https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html

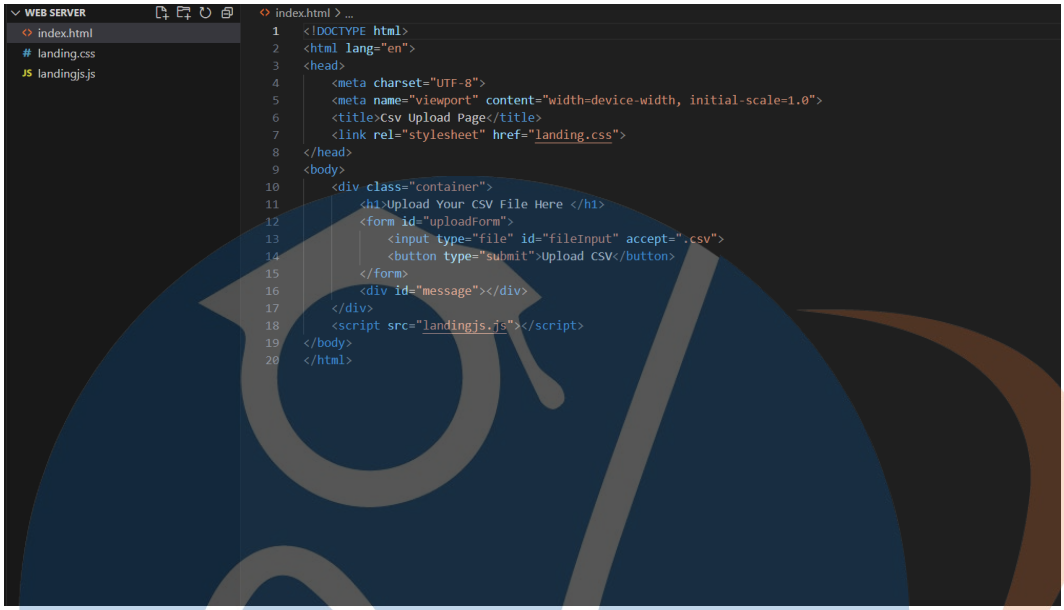
[13] Maydiantoro, Albet. "Model-model penelitian pengembangan (research and development)," *Jurnal pengembangan profesi pendidik indonesia (JPPPI)*, vol. 1, no. 2, 2021

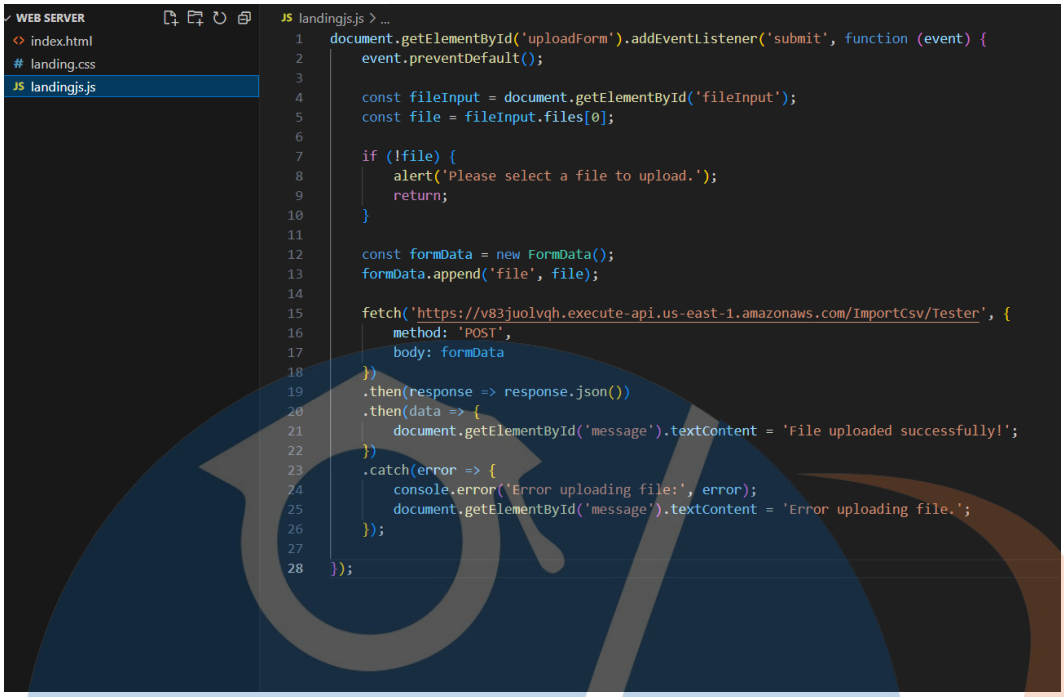
[14] D. I. Permatasari, "Pengujian Aplikasi menggunakan metode Load Testing dengan Apache JMeter pada Sistem Informasi Pertanian," *Jurnal Sistem dan Teknologi Informasi (JUSTIN)*, vol. 8, no. 1, p. 135, Jan. 2020, doi: 10.26418/justin.v8i1.34452.

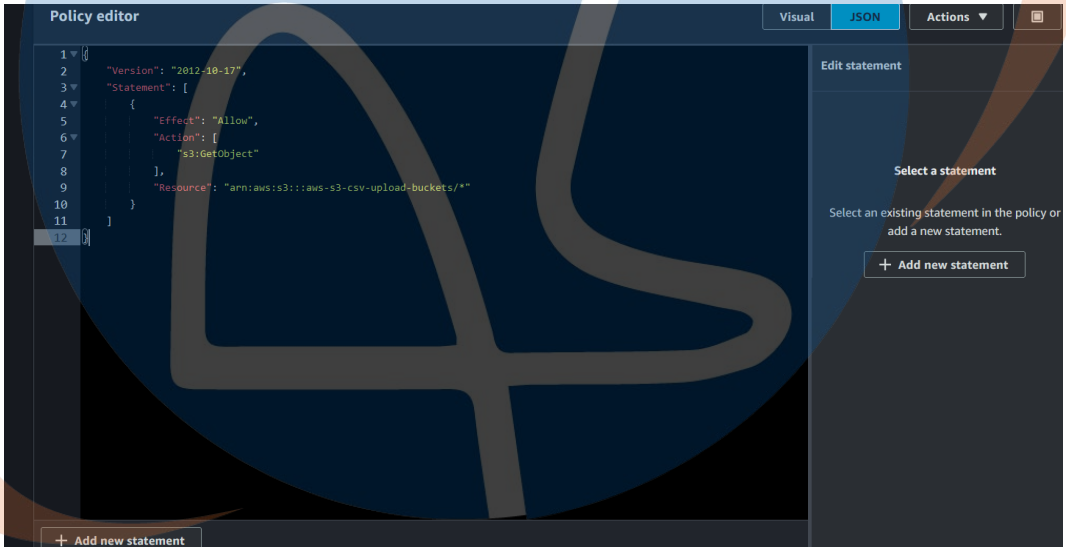
[15] Sudrajat, Asep, Dede Hertina, and Wien Dyahrini. "SISTEM LOGISTIK DI INDONESIA: TINJAUAN KELEMBAGAAN DAN SISTEM INFORMASI." *SCIENTIFIC JOURNAL OF REFLECTION: Economic, Accounting, Management and Business* 7.2 (2024): 283-297.

[16] Sinaga, Feliza Paramitha, et al. "Analisis penggunaan metode mengajar (metode demonstrasi, metode eksperimen, metode inquiry, dan metode discovery di SMA Negeri 11 Kota Jambi)." *Relativitas: Jurnal Riset Inovasi Pembelajaran Fisika* 5.2 (2023): 103-110.

LAMPIRAN




WEB SERVER
index.html
landing.css
landingjs.js
landingjs.js > ...
document.getElementById('uploadForm').addEventListener('submit', function (event) {
event.preventDefault();
const fileInput = document.getElementById('fileInput');
const file = fileInput.files[0];
if (!file) {
alert('Please select a file to upload.');
return;
}
const formData = new FormData();
formData.append('file', file);
fetch('https://v83juolvqh.execute-api.us-east-1.amazonaws.com/ImportCsv/Tester', {
method: 'POST',
body: formData
})
.then(response => response.json())
.then(data => {
document.getElementById('message').textContent = 'File uploaded successfully!';
})
.catch(error => {
console.error('Error uploading file:', error);
document.getElementById('message').textContent = 'Error uploading file.';
});
});


Policy editor
Visual
JSON
Actions
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": [
"s3:GetObject"
],
"Resource": "arn:aws:s3:::aws-s3-csv-upload-buckets/*"
}
]
}
Edit statement
Select a statement
Select an existing statement in the policy or add a new statement.
Add new statement
Add new statement