BAB II
KAJIAN LITERATUR

2.1 Tinjauan Pustaka

Tinjauan Pustaka ini menjelaskan teori yang dijadikan sebagai acuan dari penelitian ini meliputi *Monitoring* Jaringan, JSON, ELK, Logstash, Elasticsearch, Kibana.

2.1.1 Internet Protocol

Transmission Control Protocol (TCP) dan Internet Protocol (IP) merupakan standar dari komunikasi data yang dipakai oleh komunitas internet. Standar ini mengatur dalam proses tukar-menukar data atau informasi dari satu komputer ke komputer lain di dalam jaringan internet. IP address adalah alamat identifikasi komputer/host yang berada didalam jaringan. Dengan adanya IP address maka data yang dikirimkan oleh host/komputer pengirim dapat dikirimkan lewat protokol TCP/IP hingga sampai ke host/komputer yang dituju. Setiap komputer/host memiliki IP address yang unik sehingga dua komputer/host yang berbeda tidak boleh memiliki IP address yang sama dalam satu jaringan.

Terdapat 2 Jenis IP Address :

IP Adresss Private merupakan alamat-alamat IP yang disediakan untuk digunakan pada jaringan local (LAN) (Madcoms, 2009). IP Private dapat terhubung ke dalam jaringan internet melalui NAT. Network Address Translation (NAT) sendiri merupakan metode untuk menghubungkan lebih dari satu komputer ke jaringan internet dengan menggunakan satu alamat IP (Zamzami, N. F, 2013). Sehingga pada jaringan internet nantinya alamat IP private tersebut tidak terekspos karena hanya dapat terbaca satu alamat IP saja.

IP Private di gunakan untuk jaringan local yang di mana IP Private tidak terekspos ke internet dan hanya dapat di akses oleh perangkat yang sesamanya terhubung di jaringan local. Penggunaan IP private tidak perlu didaftarkan ke pihak otoritas sebelum digunakan karena penggunaan IP

private telah diatur, dialokasian dan distandarkan oleh IANA. Dalam dokumen RFC 1918

Terdapat 3 Blok kelas IP Private:

1. Kelas A

   Dengan blok kelas 10.0.0.0 to 10.255.255.255 (10.0.0.0 /8) dengan host yang berjumlah 16,777,216

2. Kelas B

   Dengan blok kelas 172.16.0.0 to 172.31.255.255 (172.16.0.0 /12) dengan host yang berjumlah 1,048,576

3. Kelas C

   Dengan blok kelas 192.168.0.0 to 192.168.255.255 (192.168.0.0 /16) dengan host yang berjumlah 65,536

IP *Address Public* merupakan alamat-alamat IP yang disediakan untuk digunakan pada jaringan internet. Karena kelas IP address ini digunakan di dalam jaringan internet maka IP ini bisa diakses melalui jaringan internet secara langsung (Madcoms, 2009). IP Publik merupakan IP yang mengidentifikasi sebuah perangkat di jaringan Publik yang dapat di akses oleh siapa saja. IP public adalah alamat IP yang digunakan dalam jaringan global Internet serta penggunaan dan alokasinya diatur oleh InterNIC untuk menjamin penggunan IP address ini secara unik.

Terdapat empat kelas dalam IP Publik:

1. Kelas A dimulai dari 0000 0000 (0) range IP 0 – 127 memiliki host maksimum sebanyak 16.777.214.
2. Kelas B dimulai dari 1000 0000 (128) range IP 128 – 191 dan memiliki host maksimum sebanyak 65.534.
3. Kelas C dimulai dari 1100 0000 (192) range IP 192 – 223 dan memiliki host maksimum sebanyak 254.
4. Kelas D dimulai dari 1110 0000 (224) range IP 224 – 239 dan digunakan untuk alamat IP multicast.

2.1.2 Port Address

Port adalah mekanisme yang memungkinkan komputer terhubung dengan beberapa sesi koneksi dengan komputer dan program lainnya dalam

jaringan. Setiap port dikaitkan dengan proses atau layanan tertentu. Port mengidentifikasi aplikasi dan service yang menggunakan koneksi di dalam protokol TCP/IP. Port memungkinkan computer. Dengan Port setiap trafik menjadi lebih spesifik dan di bedakan berdasarkan aplikasi yang di gunakan. Port number yang dapat digunakan mulai dari 0 – 65353 dan di bagi menjadi 3 kategori (J. Reynolds, J. Postel. 1992), yaitu:

1. *Well-know Ports* : Merupakan port yang telah di pesan oleh Internet Assigned Number Authority (IANA) untuk sistem dalam komputer dan jaringan dapat berjalan yang memiliki rentang 0 – 1023 dan di butuhkan akses root untuk dapat memanipulasi port tersebut.
2. *Registered ports* : Merupakan port yang di pakai oleh vendor jaringan computer untuk mendukung aplikasi dan sistem oprasi yang di buat. Rentangnya berkisar dari 1024 hingga 49151.
3. Dynamically Assigned Port : Merupakan port-port yang ditetapkan oleh sistem operasi atau aplikasi yang digunakan untuk melayani request dari pengguna sesuai dengan kebutuhan rentangnya berkisar dari 1024 hingga 65536 .

2.1.3 Log

Log merupakan pencatatan dari aktifitas yang di lakukan oleh aplikasi atau suatu perangkat. Log membantu dalam melakukan analisa terhadap suatu sistem dengan log dan dapat dengan mudah melakukan debugging terhadap suatu masalah di dalam aplikasi ataupun perangkat (Gerhards, Rainer. 2009).

Log level atau Log *severity* adalah informasi yang memberi tahu betapa pentingnya pesan log yang diberikan. Ini adalah cara sederhana, namun sangat ampuh untuk membedakan peristiwa log satu sama lain. Jika level log digunakan dengan benar di maka administrator hanya perlu melihat tingkat keparahannya terlebih dahulu.

Setiap *Entry* log berisi bidang tingkat yang menunjukan perkiraan tingkat keparahan peristiwa yang menyebabkan entri log. Di dalam Cisco memungkinkan ketika pencatatanlog di aktifkan maka Cisco mencatat

seluruh aktifitas dari router termasuk konsol, monitor, dan system secara default. Ada delapan tingkat logging (Gerhards, Rainer. 2009), yaitu :

- 0 - Emergency : Kondisi di mana sistem tidak stabil sehingga dapat mengakibatkan terhentinya sebuah sistem
- 1 - *Alert*: Kondisi dimana sistem tidak stabil dan harus segera di perbaiki.
- 2 - *Critical*: Kondisi dimana berhentinya sebuah service penting di dalam sistem dan harus segera di perbaiki.
- 3 - *Error*: Kondisi terjadinya kegagalan suatu proses service.
- 4 - *Warning*: Kondisi peringatan ketika suatu service menyentu parameter yang di tetapkan service sebelum terjadi eror.
- 5 - *Notifications*: Informasi yang umumnya berguna untuk dicatat seperti mulai atau stopnya sebuah service.
- 6 - *Informational*: Informasi terkait aktifitas service.
- 7 - *Debug*: Pesan debug dari service.

2.1.4 Monitoring Jaringan dan Sistem

Monitoring Jaringan menggunakan sistem untuk memantau secara konstan data-data dari perangkat-perangkat yang terdapat di dalam jaringan jika terjadi masalah atau perangkat yang tiba-tiba mati dan memberikan notifikasi kepada *administrator* lewat Email, SMS, ataupun alarm.

Sistem *Monitoring* Jaringan merupakan perangkat lunak yang secara otomatis melakukan pemantauan insiden yang terjadi dalam sistem jaringan serta menganalisa adanya masalah terhadap jaringan (Panjaitan & Syafari, 2019). Sistem *Monitoring* Jarigang memiliki 3 tahapan yaitu:

- Pengumpulan Data :

Pada tahapan ini dilakukan proses mengumpulkan data *monitoring* yang terdapat dalam jaringan seperti *Network Trafic, Hardware Information, Population, economy.* Pada tahapan ini data-data di kumpulkan yang nantinya di analisa.

- Analisis Data :

Pada tahapan ini akan di lakukan *selecting, filtering, updateting* pada data-data yang telah di kumpulkan untuk nantinya data dapat di analisa dan di pantau jika terdapat masalah di dalam jaringan.

- Menampilkan Data :

Pada tahapan ini data yang telah dikumpulkan dan di analisa di tampilkan dalam bentuk table, animasi, ataupun curva sehingga lebih mudah di pahami untuk mendapatkan *insight* dalam pengambilan keputusan.

Sistem *Monitoring* Jaringan di bantu oleh *network protocol* dalam mengumpulkan data dari perangkat jaringan yang di monitor. n*etwork protocol* merupakan sekumpulan peraturan yang memungkinkan perangkat jaringan di dalam jaringan bias saling berkomunikasi. Salah satu contoh *network protocol* dasar yang biasa digunakan Network *Monitoring* Sistem untuk membantu mengambil data, yaitu:

Simple Network Management Protocol

Simple Network Management Protocol atau biasa di singkat SNMP di gunakan untuk mengumpulkan data dan mengorganisir informasi dari perangkat yang terhubung di dalam jaringan, beberapa perangkat yang support dengan SNMP seperti modem, router, switch, server, printer, dan lain-lain (Prayogi, Orisa, & Ariwibisono, 2020). SNMP merupakan salah satu protocol yang digunakan untuk membantu sistem monitoring jaringan dalam melakukan pengumpulan data.

SNMP menampilkan data dari managemen sistem sebagai *variable*, *variable* yang di tampilkan oleh SNMP berbentuk hierarki. SNMP sendiri tidak menentukan hirarkinya sendiri terkait variabel mana yang harus di tampilkan oleh sistem. sebaliknya, SNMP menggunakan desain yang dapat diperluas sehingga memungkinkan aplikasi untuk menentukan hierarki sistem mereka sendiri. Hirarki ini biasa di sebut sebagai MIB (Management Information Base). Hirarki atau MIB mendeskripsikan OIDs (*object identifiers*). OID merupakan mekanisme untuk standarisasi yang di buat oleh International Telecommunication Union (ITU) dan ISO/IEC untuk memberi

penamaan pada objek, konsep, ataupun benda dengan nama yang pasti dan tidak ambigu secara global.

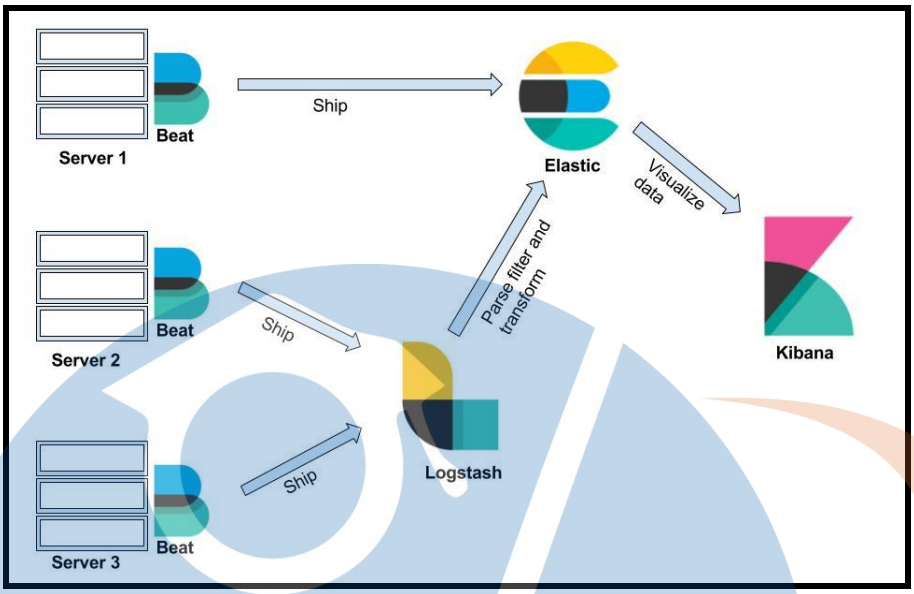
SNMP menggunakan transport protokol UDP untuk saling mengirim query data dan menerima respon (Roquero & Aracil, 2021). UDP port 161 digunakan oleh SNMP untuk mengirim dan menerima request, sedangkan port 162 di gunakan untuk menerima *traps* dari perangkat yang di *monitoring*.

2.15 JSON

JSON (JavaScript Object Notation) adalah *data representation format* untuk pertukaran data yang ringan, sangat mudah bagi manusia untuk membaca dan menulis file JSON. JSON merupakan text format yang *independent* karena menggunakan gaya bahasa umum yang biasa di gunakan oleh Programer C-*family*, seperti C, C++, C#, Java, JavaScript, Perl, Python, dan lain-lain, Sehingga membuat JSON ideal untuk pertukaran data antar bahasa pemrograman (ECMA-404, 2017). JSON terbuat dari dua strukur yaitu Key dan Value, Value dalam JSON dapat berupa String, Object, Array, True, False, null.

2.1.6 ELK

ELK *stack* merupakan ekosistem yang kaya dengan komponen *open source* yang berfungsi penuh untuk membantu dalam melakukan analisa. ELK *stack* terdiri dari 3 komponen inti yaitu Logstash, Elasticsearch, Kibana. Nantinya data dikirim dan dibaca oleh Logstash, kemudian ditransfer ke dalam Elasticsearch sebagai tempat menyimpan, mencari, dan memproses data log dan menghasilkan output dalam bentuk JSON untuk bisa ditampilkan secara visual di browser dengan Kibana (Romadhon, 2021).



Gambar 2.1 Arsitektur dan Komponen ELK *Stack*

Gambar 2.1 menjelaskan tentang arsitektur dan proses dari ELK stack di mana data di kumpulkan oleh Beat dan juga Logstas untuk di simpan di simpan di dalam Elasticsearch yang nantinya data-data tersebut di visualisasikan oleh Kibana.

Di dalam penerapannya ELK *stack* dapat menunjang kebutuhan bisnis yang efisien karena setiap komponen saling terintegrasi dan dirancang untuk memproses data yang besar secara *real-time* dan menampilkannya dalam visualisasi yang baik supaya dapat memberikan wawasan yang dapat ditindaklanjuti. Komunitas ELK *stack* juga sangat aktif dalam melakukan pengembangan sehinga memberikan dampak positif untuk ELK *stak* ke depannya. Berikut ini merupakan poin penting ke keungulan ELK *stack* :

1. ELK *stack* dapat mengambil data, menganalisis, serta melakukan visualisasi secara *real-time* dari berbagai macam sumber dengan aman.
2. ELK *stack* di siapkan untuk mengatasi *Big data* dan melakukan *filtering* untuk memberikan *business insights*.
3. ELK *stack* dapat menerapkan sistem *clustering* sehingga dapat beradaptasi dengan cukup baik terhadap pertumbuhan data.
4. ELK *stack* merupakan *open source* sehingga dapat lebih hemat biaya.

2.1.7 Elasticsearch

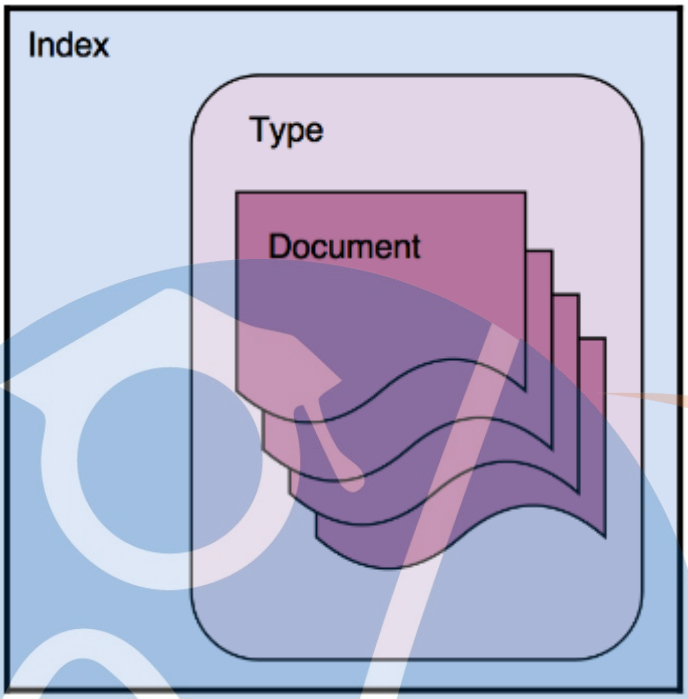
Elasticsearch Merupakan Sebuah sistem terdistribusi, gratis, *Open search*, dan dapat melakukan analisa terhadap semua jenis data, termasuk data teks, angka, *geospatial*, data terstruktur, dan yang tidak terstruktur. Elasticsearch di buat berdasarkan *library Apache Lucene* yang di rilis pada tahun 2010 oleh Elasticsearch N.V (elastic, 2021). Elasticsearch merupakan komponen utama dalam ELK *stack* dengan ciri khas REST APIs yang sederhana, cepat, dan memiliki skalabilitas yang baik. Sebagai komponen pusat dari ELK *stack* Elasticsearch memiliki kumpulan *tools* gratis untuk data *ingestion*, menyimpan, menganalisis dan memvisualisasikan data. Dalam penerapannya Elasticsearch biasa di gunakan untuk (elastic, 2021) :

1. Metrik infrastruktur dan pemantauan container
2. Pencarian aplikasi
3. Pencarian situs *web*
4. Pencarian perusahaan
5. Log dan analitik log
6. Pemantauan kinerja aplikasi
7. Analisis dan visualisasi data geospasial
8. Analisis keamanan
9. Analisis bisnis

Terdapat beberapa bagian penting dalam struktur penyimpanan yang ada di Elasticsearch, antara lain yaitu:

- *Indexes*

Index merupakan sebuah container di dalam elasticsearch, index berfungsi untuk menyimpan dan memanage document dari satu jenis *type* di Elasticsearch (Shukla & Kumar M N, 2019). Gambar 2.2 menjelaskan tentang struktur dari Elasticsearch yang di mana terdapat sebuah *index* menyimpan satu buat *type* yang berisi beberapa *documents*. Index bertugas



Gambar 2.2 Struktur dari Index, Type dan Documents

- *Types*

Types berfungsi untuk mengelompokan dan mengatur D*ocuments* yang sama. Elasticsearch bersifat *schemaless* yang artinya dapat menyimpan *Documents* JSON dalam bentuk apapun sehingga perlu menghindari penyimpanan dengan entitas yang berbeda (Handika, 2020).

- *Documents*

Documents merupakan entitas terkecil dalam struktur data Elasticsearch yang di mana terdiri dari beberapa *field* dan merupakan bentuk dasar informasi yang berada di dalam Elasticsearch. Yang melambangkan sebuah kolom dalam *database relasional*.

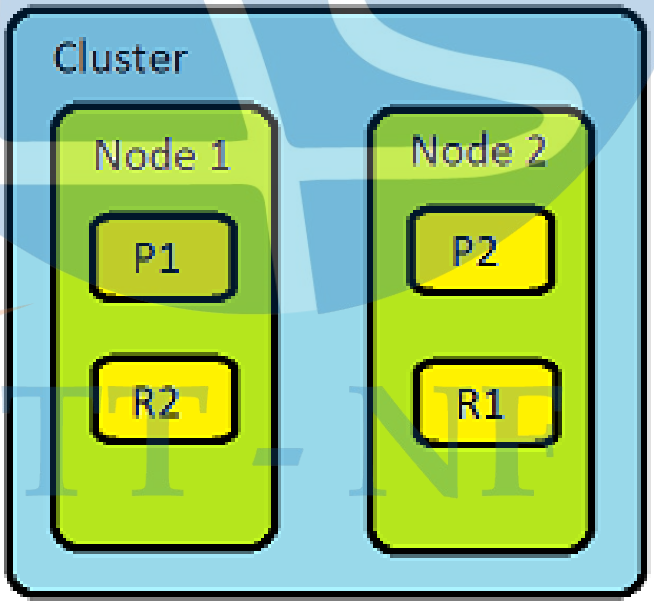
Elasticsearch dapat dibuat menjadi sebuah *cluster* yang terdiri dari satu atau lebih *Node*. *Node* di sini merupakan sebuah Elasticsearch server yang tergabung sehingga membentuk sebuah *cluster* Elasticsearch. Setiap *node* dalam *cluster* Elasticsearch memiliki tanggung jawab untuk menyediakan oprerasi seperti *searching*, *indexing*, and *aggregations* (Shukla & Kumar M N, 2019). *Cluster* di dalam Elasticsearch menerapkan *Shards and Replicas* untuk *High-Avaibility Cluster*.

- *Shards*

Sebuah *Index* berisi satu atau beberapa *Documents*. *Shards* membantu dalam memisahkan beberapa *Documents* dalam satu buah *index* kedalam beberapa *node*. *Shards* merupakan karakter dari elasticsearch dan *Shards* merupakan cara untuk *scaling* dan *parallelizing* yang dapat membantu memanfaatkan penyimpanan secara maksimal dan memanfaatkan kekuatan dari setiap node secara maksimal dalam memproses data (Shukla & Kumar M N, 2019). Secara default sebuah *Index* di pecah menjadi 5 *Shards* dan tidak dapat di rubah jika *Index* tersebut telah terbentuk. Jika ingin merubah sebuah index maka harus di tetapkan saat pertama membentuk Index tersebut.

- *Replicas*

Replicas merupakan fiture yang di sediakan Elasticsearch saat terjadi kegagalan dalam sebuah *cluster*. Setiap *shards* yang berada dalam sebuah *Index* dapat memiliki 1 buah salinan atau lebih dan setiap salinan ini memiliki fungsi yang sama persis terhadap aslinya (Shukla & Kumar M N, 2019).



Gambar 2.3 Contoh Replicas dan Shards di Cluster Elasticsearch

Gambar 2.3 menjelaskan tentang Cluster Elasticsearch dengan dua buah Node, yang di mana sebuah Index di pecah menjadi dua buah Shards dan setiap Shards memiliki satu Replica.

2.1.8 Logstash

Logstash adalah *data pipeline* yang meruapakan suatu tahapan pemrosesan data, di logstash istilah *data pipeline* di bagi menjadi 3 buah proses yaitu *input, filter, output*. Logstash merupakan kunci di dalam ELK *stack* karena Logstash memusatkan data peristiwa seperti log, metrik atau data lainnya dalam format apapun dan melakukan transformasi data sesuai kebutuhan sebelum menyimpan ke dalam penyimpanan yang di inginkan. Logstash memiliki lebih dari 200+ total *plugin* yang dapat di pakai dalam mendukung *input, filter, output* (Shukla & Kumar M N, 2019). Logstash sangat bagus di jadikan mesin untuk mengolah data yang dapat bekerja secara *real-time*, dan *scalable*.

2.1.9 *Dashboard*

Dashboard dapat diartikan sebagai *information dashboard* yaitu tampilan visual dari informasi penting yang diperlukan untuk mencapai satu atau beberapa tujuan dengan mengkonsolidasikan dan mengatru informasi dalam satu layar (*single screen*), sehingga kinerja organisasi dapat di monitor dengan sekilas. *Dashboard* dapat juga di gunakan untuk memvisualisasikan sebuah data kejadian sehingga bisa mendapatkan *insight* yang tersembunyi di dalam data tersebut. Di dalam penggunaanya Dashboard di kategorikan ke dalam tiga tipe, (Ilhamsyah & Rahmayudha, 2017) yaitu :

- Dashboard strategis

  Dashboard strategis digunakan untuk mendukung manajemen level strategis memberikan informasi dalam membuat keputusan bisnis, memprediksi peluang, dan memberikan arahan pencapaian tujuan strategis.

- Dashboard taktis

Dashboard tipe ini berfokus pada proses analisis untuk menentukan penyebab dari suatu kondisi atau kejadian tertentu.

- Dashboard operasional

Dashboard operasional yang berfungsi sebagai pendukung monitoring dari aktifitas proses bisnis yang spesifik.

Berdasarkan ke 3 jenis dashboard yang di sebutkan di atas peneliti menerapkan Dashboard Operasional yang di gunakan sebagai pendukung untuk melakukan monitoring dari aktifitas proses bisnis yaitu sistem dan jaringan. Nantinya peneliti akan membuat Dashboard sistem monitoring dan jaringan dengan menggunakan Kibana.

2.1.10 Kibana

Kibana merupakan *tool open source* untuk memvisualisasikan data dari Elastic *stack* berbasis *web*. Kibana menawarkan jenis visualisasi yang beragam seperti *histogram*, *time series*, *chart*, *graphs*, *maps*, *table*, dan lain-lain sehingga dapat memberikan insight tentang data yang telah di kumpulkan. Kibana juga dapat mengkombinasikan beberapa jenis visualisasi seperi yang telah di sebutkan sehingga dapat menciptakan sebuah *Dashboard* yang interaktif dan berkualitas (Shukla & Kumar M N, 2019).

2.2 Penelitian Terkait

Dalam penelitian ini menjelaskan tentang penelitian terdahulu bertujuan sebagai bahan perbandingan dan acuan. Selain itu, peneliti menghindari yang beranggapan kesamaan dengan penelitian sebelumnya. Daftar penelitian terkait yang peneliti temukan bisa dilihat ditabel dibawah ini.

Tabel 2.1 Penelitian Terkait

No	Nama dan Tahun	Judul	Topik	Hasil
1	(WAHYU DWI ROMADHON, 2021)	*IMPLEMENTASI SURICATA IDPS UNTUK MONITORING JARINGAN*	*Intrusion Detection System*	Sistem dengan *Suricata* yang dapat mendeteksi adanya serangan, ELK yang dapat memvisualisasikan log *Suricata*

		DENGAN VISUALISASI ELK (ELASTICSEARCH, LOGTASH, KIBANA) DAN NOTIFIKASI MELALUI BOT TELEGRAM.		sehingga mudah di mengerti dan terintegrasi dengan bot *Telegram* secara *realtime* untuk mengirimkan *Alret*.
2	(Prayogi, Orisa, & Ariwibisono, 2020)	RANCANG BANGUN SISTEM MONITORING JARINGAN ACCESS POINT MENGGUNAKAN SIMPLE NETWORK MANAGEMENT PROTOCOL (SNMP) BERBASIS WEB	*Network Monitoring System*	Sistem *monitoring* jaringan untuk Pemantauan *AccessPoint* dengan menggunakan protocol SNMP berbasis *web* yang dapat memonitor resource, identitas, jumlah *client*, SSID dan juga dapat mengirim notifikasi kepada administrator jaringan jika terjadi masalah dalam *AccessPoint*
3	(Sulasno Sulasno, Rakhmat Saleh, 2020)	Desain dan Implementasi Sistem Monitoring Sumber Daya Server Menggunakan Zabbix 4.0	*Network Monitoring System*	Server Monitoring yang dapat memonitor dan menampilkan dalam bentuk grafik hasil monitoring meliputi informasi penggunaan sumber daya hard disk, utilitas CPU,

				penggunaan memori, dan besaran lalu lintas paket data ethernet baik paket data yang masuk ke sistem (*incoming*), maupun paket data yang keluar dari sistem (*outgoing*).