# MITIGASI SERANGAN SYBIL DENGAN METODE *BEHAVIOUR DETECTION* PADA JARINGAN *FOG COMPUTING*

**Yusuf Agung Purnomo[1], Dr. Vera Suryani, S.T., M.T.[2], Aulia Arif Wardana, S.Kom., M.T.[3]**

[1,2,3]Fakultas Informatika, Universitas Telkom, Bandung
[1]yusufap@student.telkomuniversity.ac.id, [2]verasuryani@telkomuniversity.ac.id,
[3]auliawardan@telkomuniversity.ac.id

**Abstrak**

***Internet of Things* saat ini berkembang dengan pesat, pengaplikasian teknologi ini sudah banyak dilakukan di berbagai macam lini. Tetapi, tidak berarti bahwa teknologi ini aman dari serangan *cyber*, salah satu jenis serangan yang bisa menyerang jaringan ini ialah *Sybil* yaitu sebuah jenis serangan di mana penyerang akan meniru identitas *node* lokal yang ada di jaringan tersebut. Sehingga diperlukan metode pendeteksian yang bisa menyaring dan memisahkan penyerang ini sehingga performansi jaringan tetap terjaga.**

**Salah satu metode yang bisa dilakukan untuk mendeteksi aktifitas *Sybil* adalah dengan metode *Behaviour Detection*, di mana metode ini akan memeriksa setiap komunikasi keluar masuk jaringan dan memisahkan *sybil node* keluar jaringan jika terdeteksi. Kriteria sebuah *node* dikatakan *sybil* adalah ketika ia melewati sebuah *threshold* yang telah ditentukan, komponen *threshold* itu di antara nya adalah cepat nya perubahan *ID*, dan lokasi dari suatu *node*.**

**Hasil yang didapatkan adalah sistem ini dapat mendeteksi *sybil node* dengan tingkat keberhasilan hingga 90%. Hasil ini didapatkan dengan mengatur sensitivitas *threshold* yang digunakan untuk proses pendeteksian.**

**Kata Kunci**: ***internet of things, sybil, fog computing, honeypot, behavioral detection, iot.***

**Abstract**

**Nowadays, Internet of Things are fastly being develop, the application of this technology had already been done on several fields. But, it doesn't mean that this technology is safe from cyber attacks, one of cyber attacks that could attack this network is Sybil where the attackers will impersonating other user on the network. So. a detection method is needed to filter and separate these attackers in order to maintain the performance of the network.**

**One of the methods that can be use to detect Sybil activities is Behaviour Detection method which it will check every communications in and out of the network and separated the sybil node out of the network if detected. The criteria of a node can be called as a sybil is when it pass certain thresholds that has been already set, one of the threshold components are how fast the interval of an ID and locations of a node changed.**

**The end result is this system can detect sybil node with success rate up to 90%. The result came from arranging the threshold sensitivity that being used to do detection.**

**Key Words : *internet of things, sybil, fog computing, honeypot, behavioral detection, iot.***

# 1. PENDAHULUAN

## 1.1. Latar Belakang

Perkembangan teknologi *Intenet of Things* banyak membawa perubahan. Salah satunya menghubungkan objek yang terhubung dengan jaringan *internet* ke dunia fisik sehingga bisa memudahkan interaksi antara manusia dengan objek. Objek – objek ini bisa mengambil data dari lingkungan sekitar nya berkat sensor – sensor yang tertanam dalam objek tersebut[1]. Jika digabungkan dengan kemampuan untuk melakukan *sensing*, komunikasi dan komputasi [2],[3], *Internet of Things* bisa memberikan berbagai layanan pintar[4] untuk membentuk jaringan *smart home* [5], *smart grid* [6],[7],[8] dan lain lain. Karenanya perkembangan *Internet of Things* sangat berpengaruh terhadap keseharian kita dalam beraktifitas.

Tetapi, jaringan *Internet of Things* sendiri bukanlah sebuah jaringan yang dijamin aman tanpa serangan, objek – objek yang saling berhubungan dalam jaringan ini rentan terhadap serangan baik dari luar atau dari dalam. Jika salah satu *node* mengalami *down*, mempengaruhi kinerja seluruh jaringan. Jenis serangan yang akan digunakan dalam penelitian ini adalah *Sybil attack* dimana sebuah *node* memiliki berbagai identitas sehingga penyerang bisa berada di berbagai tempat dalam satu waktu penyerangan. Tujuan dari *Sybil attack* adalah untuk mengurangi integritas dari keamanan data dan penggunaan sumber daya pada jaringan. *Sybil attack* diambil sebagai kasus dikarenakan tingkat kerusakan yang bisa ditimbulkan oleh serangan ini, sebagai contoh dalam sebuah sistem *online voting*, *node Sybil* bisa secara illegal membuat banyak identitas sebagai *node* biasa sehingga hasil *vote* akan condong ke salah satu pihak , contoh lain adalah pada media jejaring sosial, pada sebuah laporan di tahun 2012[1], terdapat akun – akun dalam jumlah substansial yang terkonfirmasi sebagai akun palsu atau akun *Sybil* dalam jaringan sosial daring, 76 juta akun di *Facebook* dan 20 juta akun palsu dibuat di dalam *Twitter* per minggu nya. Akun – akun palsu ini selain menyebarkan iklan – iklan menyesatkan dan *spam*, tetapi juga menyebarkan *malware* dan website *phising* untuk mencuri informasi pribadi akun – akun yang terinfeksi *malware* dan/atau masuk ke website *phising* tersebut. Oleh karena itu, diperlukan suatu mekanisme pertahanan dari serangan *Sybil* ini agar sistem tidak mengalami *down*.

Mekanisme pertahanan yang diusulkan oleh penulis adalah mekanisme pertahanan dengan menggunakan *behavioural detection* yang dikombinasikan dengan *honeypot*. Jenis jaringan yang akan digunakan dalam implementasi mekanisme ini adalah jaringan *fog computing*. Dengan mekanisme pertahanan secara ringkas sebagai berikut, di dalam jaringan setiap komunikasi data akan melewati *intrusion detection system* (IDS), ciri dari sebuah *node Sybil* menurut Zhang dkk. [1] adalah ia melakukan suatu kegiatan spesifik dengan frekuensi yang lebih tinggi daripada *node* normal. IDS mendeteksi dan melacak *node* mana yang terindikasi sebagai *Sybil* dan diarahkan ke *honeypot*, data terkait penyerang akan tersimpan di dalam sebuah *log*.

## 1.2. Perumusan Masalah

Bagaimana memisahkan *Honest nodes* dan *Sybil nodes* yang terdapat dalam sebuah jaringan *Fog Computing* dengan menggunakan metode deteksi perilaku.

## 1.3. Batasan Masalah

Batasan masalah yang ditentukan adalah :

1. Parameter hasil pengujian yang digunakan adalah berapa persen tingkat deteksi sistem,
2. Simulasi dijalankan dengan bahasa pemrograman python,
3. Jenis serangan Sybil yang diteliti adalah SA-2,
4. Protokol yang digunakan dalam komunikasi adalah socket protokol dalam satu perangkat komputer.

## 1.4. Tujuan

Tujuan dari penelitian ini adalah :

1. Melakukan simulasi jaringan yang memiliki *honest nodes* dan *Sybil nodes*,
2. Mengimplementasikan deteksi *Sybil nodes* pada *fog computing*,
3. Merancang sistem yang dapat membedakan *honest nodes* dan *Sybil nodes*,
4. Mengukur akurasi deteksi *Sybil nodes* yang telah diimplementasikan dalam bentuk persentase jumlah serangan yang terdeteksi.

## 1.5 Organisasi Tulisan

2. Kajian Pustaka : berisi teori/studi literatur yang mendukung topik TA yang dikerjakan.
3. Sistem yang Dibangun : penjelasan sistem atau produk yang dihasilkan.
4. Evaluasi : berisi hasil dan analisis pengujian yang dilakukan.
5. Kesimpulan : berisi kesimpulan dan saran untuk pengerjaan yang akan datang.
6. Daftar Pustaka : berisi kumpulan sumber yang digunakan dalam menyusun TA ini.

## 2. KAJIAN PUSTAKA

### 2.1. Sybil Attack

*Sybil Attack* adalah salah satu jenis ancaman yang menyerang *platform Internet of Things*. Prinsip kerja nya secara sederhana adalah penyerang akan mengambil identitas dari *node – node* lain yang terhubung dalam satu jaringan dan menggunakan identitas tersebut untuk membuat kerusakan pada jaringan, seperti menyebarkan *spam* atau *malware*. Kuan Zhang dkk.[1] mendefinisikan jenis serangan *Sybil* menjadi tiga kategori berdasarkan model grafis sosial :

1. SA-1 *Sybil Attack*

   Dalam jenis serangan ini, *node sybil* memiliki kemampuan yang tidak terlalu kuat sehingga jumlah keterhubungan antara *sybil nodes* dengan *honest nodes* masih terbatas. Pelaku SA-1 biasanya berada pada *sensing* dan *social domain* seperti sistem *voting* atau pada sistem *mobile sensing*. Tujuan utama dari serangan ini adalah memanipulasi keseluruhan pilihan atau popularitas dalam sistem. Contohnya pada sebuah sistem pemilihan, *sybil nodes* dengan identitas ganda nya akan sanggup mempengaruhi hasil pemilihan dengan menggunakan identitas curian tersebut. Dalam ranah *mobile sensing*, *sybil nodes* akan memberikan data pembacaan sensor yang kurang tepat sehingga mempengaruhi keseluruhan pembacaan. Sehingga oleh karena itu dalam beberapa kasus, pelaku sulit dibedakan dengan pengguna normal.

2. SA-2 *Sybil Attack*

   Pelaku penyerangan SA-2 biasanya berada pada *social domain* dan mampu membentuk sebuah hubungan dengan *honest nodes* tidak seperti SA-1. Sehingga kemampuannya dalam meniru *nodes* normal dalam perspektif grafik sosial adalah kuat. Tujuan SA-2 adalah menyebarkan *spam*, iklan, dan *malware*, mencuri dan melanggar privasi pengguna lain, dan memanipulasi reputasi sistem. SA-2 juga mampu dalam konteks *social domain* membuat berbagai komentar tentang sistem tersebut, komentar positif untuk melebih – lebihkan sistem dan komentar negatif untuk menjatuhkan reputasi dari sistem. Sehingga untuk SA-2, bisa diketahui dengan aktifitasnya yang berfokus ke suatu kegiatan spesifik dengan frekuensi pengulangan yang tinggi.

3. SA-3 *Sybil Attack*

   Pelaku SA-3 biasanya berada pada *domain mobile sensing*, tujuan dari SA-3 sama dengan SA-2 hanya bedanya adalah akibat dan durasi yang dihasilkan dari serangan ini, yaitu hanya berdampak secara lokal atau pada rentang waktu yang pendek. Ini dikarenakan oleh pengguna jaringan *mobile* yang selalu bergerak sehingga terkadang mereka tidak terhubung dengan jaringan. Otoritas terpusat tidak dapat selalu ada pada jaringan *mobile*, sehingga tidak seperti sistem daring, ada beberapa aspek yang tidak dapat diperoleh oleh sistem pertahanan terhadapa SA-3. Mobilitas dan kurangnya informasi global berakibat pada sulitnya pertahanan SA-3 daripada SA-1 dan SA-2.

### 2.2. Fog Computing

Konsep *fog computing* pertama kali diperkenalkan oleh Cisco di tahun 2012, yaitu perpanjangan dari paradigma *cloud computing* yang menyediakan layanan komputasi, penyimpanan, dan jaringan diantara *end devices* dengan *cloud server* tradisional[9]. Sehingga *fog computing* bukan pengganti dari *cloud* tapi hanya sebagai pelengkap dari sistem *cloud*. *Nodes* pada *fog computing* di *deploy* secara heterogenik pada ujung jaringan yang berdekatan dengan perangkat *IoT*. Lalu definisi *fog computing* ini dikembangkan lebih lanjut menjadi, "sebuah skenario dimana sejumlah besar perangkat yang heterogenik, dimana – mana, dan desentralistik berkomunikasi dan secara potensial bekerja sama diantara sesama perangkat dan jaringan untuk melakukan penyimpanan dan pemrosesan tugas tanpa campur tangan pihak ketiga"[10]. Dengan definisi baru tersebut, maka *fog computing* tidak hanya sebagai perpanjangan dari *cloud*, tetapi juga sebagai paradigma itu sendiri. Dengan semakin pesatnya perkembangan *Internet of Things*, *fog computing* semakin dibutuhkan dikarenakan *cloud computing* sendiri tidak mampu menangani semakin banyaknya kebutuhan dalam bidang *Internet of Things*.

*2.3.1.* Arsitektur *Fog Computing*

Jianbing Ni dkk.[11] membagi *Fog Computing* menjadi dua kategori, yaitu *Cloud-Fog-Device* dan *Fog-Device* kerangka kerja. Kerangka kerja pertama memiliki tiga lapisan, lapisan *cloud*, lapisan *fog* dan lapisan *device*, sedangkan kerangka kerja kedua hanya memiliki dua lapisan, lapisan *device* dan lapisan *fog*. Kerangka – kerangka kerja ini disusun dengan memperhatikan urutan kemampuan penyimpanan dan pemrosesan dimana yang kemampuannya paling tinggi maka ia ada di puncak. Setiap layer bersifat fleksibel dan bisa ditambah komponennya, sehingga bisa memenuhi apabila diperlukan penambahan kemampuan pada layer tersebut.

Lapisan *Device* memiliki dua jenis perangkat, perangkat *IoT* bergerak dengan yang tidak atau *fixed*. Contoh perangkat *IoT* bergerak adalah perangkat – perangkat yang bisa dibawa oleh penggunanya, misal seperti jam tangan pintar. Sedangkan perangkat *IoT* yang tidak bergerak atau *fixed* contohnya seperti sensor pendeteksi kebakaran hutan atau sensor pemeriksa kualitas

udara, perangkat – perangkat ini sudah di *deploy* di suatu tempat spesifik untuk memenuhi suatu tugas tertentu yang spesifik untuk perangkat itu saja. Baik perangkat bergerak atau tidak, keduanya memiliki kemampuan komputasi dan penyimpanan yang rendah, karena tugas mereka hanyalan mengumpulkan data mentah lalu meneruskannya ke lapisan selanjutnya untuk diproses.

Lapisan *Fog* berisi peralatan jaringan, seperti *routers, bridges, switches, local servers*. *Nodes* disebarkan secara hirarki diantara *cloud* dan perangkat *IoT* pada kerangka kerja *Cloud – Fog – Device* atau diatas perangkat *IoT* pada kerangka kerja *Fog – Device*. Lapisan ini cenderung memanjangkan komputasi awan ke ujung jaringan. Lapisan ini memiliki beberapa kemampuan untuk melakukan komputasi dan pemrosesan untuk meringankan beban pada perangkat *IoT*. Lapisan *Cloud* pada kerangka kerja *Cloud – Fog – Device* adalah sekumpulan *platform* komputasi dan penyimpanan yang menyediakan berbagai layanan aplikasi *IoT* dari perspektif global. *Cloud* memiliki kemampuan komputasi dan penyimpanan yang cukup signifikan dan bisa diakses oleh pengguna nya dari mana saja dan kapan saja selama mereka terkoneksi dengan *internet*. *Cloud* menerima ringkasan data dari berbagai *node fog* dan melakukan analisis secara global dari berbagai data yang diberikan untuk meningkatkan wawasan bisnis pada aplikasi *IoT*.

*2.3.2.* Fitur dari *Fog Computing*

Fitur utama dari *Fog Computing* adalah ia mampu menangani data *IoT* secara lokal dengan memanfaatkan *node – node fog* yang diletakkan di dekat pengguna sehingga bisa membuat penyimpanan, komputasi, transmisi, kontrol dan manajemen data lebih nyaman. Shropshire[12] mendefinisikan lima fitur *fog computing* sebagai berikut :

1. *Location Awareness*, lokasi dari sebuah *node fog* bisa dilacak secara aktif atau pasif untuk mendukung perangkat dengan layanan yang ramai di ujung jaringan.
2. *Geographic Distribution*, *node fog* diletakkan di posisi tertentu, seperti di sepanjang jalan raya, di stasiun selular, di lantai musium atau pada tempat – tempat tertentu lainnya. Alasannya adalah agar *node* bisa memperoleh aliran data dengan kualitas yang tinggi dari perangkat *IoT*.
3. *Low Latency*, *node fog* mampu melakukan komputasi dan penyimpanan tanpa *cloud* dengan relatif cepat dikarenakan posisinya yang dekat dengan perangkat sehingga memiliki latensi yang lebih rendah daripada *cloud*.
4. *Large – Scale IoT Applications Support*, dalam aplikasi *IoT* skala besar seperti pemantauan pergantian iklim, *fog computing* memiliki kemampuan dan otonomi untuk mengatur miliaran perangkat *IoT*.
5. *Decentralization*, *fog computing* adalah arsitektur desentralistik yang tidak memiliki *server* pusat untuk mengatur sumber daya dan layanan. *Node fog* bekerja dengan sendirinya untuk bekerja sama menyediakan layanan secara *real-time* dan aplikasi *IoT* untuk pengguna nya.

## 2.3. Metode Pertahanan

Kumar, Ch Niranjan[13] dalam penelitiannya memberikan beberapa jenis metode pertahanan untuk menghadapi serangan *Sybil*. Diantaranya adalah :

1. *Trusted Devices*
   Metode ini berfungsi dengan mengikat satu perangkat keras ke satu entitas di dalam jaringan. Meskipun cara ini cukup efektif tetapi tidak ada cara efisien untuk mencegah satu entitas mengambil beberapa perangkat keras selain dengan intervensi manual.
2. *Trusted Certification*
   Metode ini memiliki potensi untuk mengeliminasi serangan *Sybil* secara menyeluruh. Tetapi, metode ini bergantung dengan otoritas pusat yang harus menyediakan tiap *node* dengan identitas unik yang diindikasikan dengan kepemilikan sertifikat. Metode ini harus dilakukan secara manual sehingga menyebabkan terjadinya *bottleneck*.

Metode yang diajukan oleh penulis dalam penelitian ini memberikan keleluasaan kepada *node – node* untuk melakukan komunikasi ke *cloud*. Namun, teknik ini pun juga memiliki kelemahan yaitu suatu *node* bisa saja terdeteksi sebagai *sybil* ketika aktifitas nya mirip seperti serangan sebuah *node* sybil meskipun *node* tersebut adalah *honest node*.

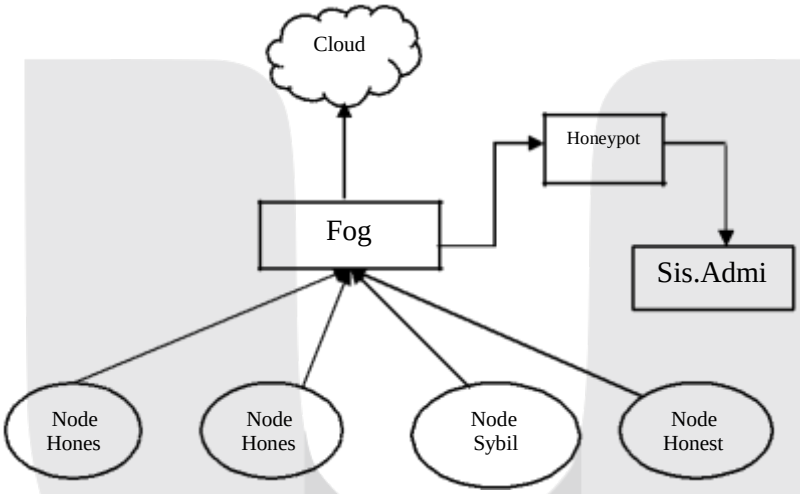
## 3. SISTEM YANG DIBANGUN

### 3.1. Gambaran Umum Sistem

Sistem yang dibangun berfungsi untuk mendeteksi *node Sybil* dengan cara melihat aktifitas dari si *node* tersebut, yaitu satu buah *node* yang melakukan aktivitas dengan frekuensi lebih tinggi daripada *node* di sekitarnya. Aliran data akan diterima oleh IDS, apabila dideteksi aktifitas tidak wajar dari sumber aliran data maka aliran data dari *node* sumber akan di alihkan ke *honeypot* jika tidak aliran data akan diteruskan ke *cloud server*. Sistem pada jaringan ini terdiri dari node client yang memiliki dua jenis yaitu jenis node honest dan jenis node sybil. Node honest adalah node yang memiliki sifat tidak melakukan serangan juga memiliki kebiasaan sebagai node yang normal, sedangkan node sybil bersifat melakukan aktivitas seperti node normal namun memiliki tujuan tersendiri yaitu melakukan perubahan identitas dan juga melakukan serangan berupa koneksi secara cepat dan terus-menerus kepada targetnya.

### 3.2. Perancangan Sistem

Mekanisme kerja sistem yang dibangun, mulai dari menerima masukan berupa aliran data, memilah dan melakukan pengalihan akses terhadap *node Sybil*, direpresentasikan secara umum dalam topologi sebagai berikut :




Gambar 1 Topologi sistem yang dirancang

Serangan *Sybil* yang akan diujicobakan dengan sistem ini adalah jenis serangan *Sybil* dengan perilaku *node* tersebut akan selalu berganti – ganti identitas dan lokasi dengan identitas yang diambil merupakan identitas *honest node* yang ada di jaringan tersebut dan *sybil node* akan melakukan serangan berupa *spamming* atau *high frequency requesting*. Suatu *node* akan dikatakan sebagai *sybil* apabila sistem mendeteksi *node* tersebut melakukan perubahan ID secara acak atau terprogram, posisi nya berubah (dinyatakan dalam sumbu x dan y), dan melakukan serangan berupa salah satunya *spamming* dengan tujuan untuk mengganggu *traffic* dari jaringan tersebut.

*Honeypot* dirancang untuk melakukan penerimaan dan pembalasan data layaknya *cloud* tetapi *honeypot* hanya menangani akses *node* yang telah dinyatakan sebagai *sybil* oleh IDS. IDS dibangun di dalam *fog* dan bertujuan untuk melakukan *listening* dari *node* yang telah terkoneksi ke jaringan, jika *node* tersebut

```
            print "FINISH!"
        elif(message =="post"):
            print "sending to server"
            HoneypotSocket.send(data.encode()) #jika terdeteksi sybil, kirimkan ke honeypot
            data_Honeypot = HoneypotSocket.recv(1024) #tunggu dan baca balasan dari honeypot
            Honeypot_response = '\nFull data: ' + format(data_Honeypot) #rubah format balasan
            print "Received from server: {}".format(data) #cetak balasan dari honeypot
            connection.send(Honeypot_response.encode()) #kirimkan balasan dari honeypot ke node (sybil)
    else:
        if(message == "quit"):
            print "FINISH!"
        elif(message =="post"):
            print "sending to server"
            serversocket.send(data.encode()) #dikirimkan dulu pesan dari node ke cloud
            data_server = serversocket.recv(1024) #tunggu dan baca balasan dari cloud
            server_response = '\nFull data: ' + format(data_server) #baca balasan dari cloud
            print "Received from server: {}".format(data) #cetak ke asan cloud
            connection.send(server_response.encode()) #kirimkan balasan dari cloud ke node
    dataout.write(tanggal.strftime("%y-%b-%d %H:%M:%S")+ "\t"  + str(addr2) + "\t" + str(x) + "\t" + str(y)
     + "\n")
```

Gambar 2 Mekanisme *redirecting node* ke *cloud* atau *honeypot*

dinyatakan aman maka *fog* akan meneruskan pesan ke *cloud* dan *cloud* akan mengirimkan balasan ke *node* tersebut melalui *fog*. Namun, jika *node* tersebut berdasarkan *behavior* nya teridentifikasi sebagai *Sybil* maka ia akan diarahkan menuju *Honeypot*.

## 4. EVALUASI

### 4.1. Hasil Pengujian

Pengujian dilakukan dengan menguji salah satu dari tiga parameter, yaitu *threshold*, *interval threshold*, dan *sybil threshold*. *Threshold* berfungsi sebagai sensitivitas deteksi berdasarkan perubahan ID, lokasi node dan *interval* sebuah koneksi node, *interval threshold* menunjukkan batas apakah suatu *node* merupakan *node* yang memiliki *fast interval*, sedangkan *sybil threshold* berfungsi sebagai batas untuk menentukan apakah suatu *node* merupakan *sybil* apa tidak berdasarkan dari perhitungan rata – rata perubahan pada ID, lokasi node dan *interval* koneksi *node* tersebut. Percobaan dilakukan sebanyak 5 kali dengan menggunakan 5 buah *honest node* dan *sybil node* yang dibuat semakin banyak di tiap pengujiannya dan melakukan penyerangan sebanyak 100 kali. Pemilihan besaran nilai *threshold* disesuaikan dengan paper refrensi dengan sampel setiap *node* melakukan *faking* ID sebanyak 3 hingga 5 ID, sehingga besaran nilai *threshold* yang diuji berada di angka 1 hingga 10. Berikut adalah hasil pengujian dengan syarat pengujian apabila salah satu parameter dijadikan sebagai objek pengujian maka parameter lain dibuat konstan :

Pengujian dengan 5 *honest nodes* dan 1 *sybil node*

| *Threshold* | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| *Interval Threshold* | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| *Sybil threshold* | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 |
| | | | | | | | | | | |
| Banyaknya Serangan | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| *Sybil nodes* terdeteksi | 87 | 88 | 87 | 91 | 93 | 91 | 94 | 96 | 97 | 97 |
| Persentase | 87% | 88% | 87% | 91% | 93% | 91% | 94% | 96% | 97% | 97% |

Tabel 1 pengujian dengan 5 honest nodes dan 1 sybil node

Pengujian dengan 5 *honest nodes* dan 2 *sybil nodes*

| *Threshold* | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| *Interval Threshold* | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| *Sybil threshold* | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 |
| | | | | | | | | | | |
| Banyaknya Serangan | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| *Sybil nodes* terdeteksi | 65 | 72 | 76 | 78 | 84 | 80 | 86 | 89 | 90 | 94 |
| Persentase | 65% | 72% | 76% | 78% | 84% | 80% | 86% | 89% | 90% | 94% |

Tabel 2 pengujian dengan 5 honest nodes dan 2 sybil nodes

Pengujian dengan 5 *honest nodes* dan 3 *sybil nodes*

| *Threshold* | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| *Interval Threshold* | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| *Sybil threshold* | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 |
| | | | | | | | | | | |
| Banyaknya Serangan | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| *Sybil nodes* terdeteksi | 58 | 60 | 63 | 67 | 72 | 72 | 76 | 80 | 87 | 91 |
| Persentase | 58% | 60% | 63% | 67% | 72% | 72% | 76% | 80% | 87% | 91% |

Tabel 3 pengujian dengan 5 honest nodes dan 3 sybil nodes

Pengujian dengan 5 *honest nodes* dan 4 *sybil nodes*

| *Threshold* | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| *Interval Threshold* | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| *Sybil threshold* | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 |
| | | | | | | | | | | |
| Banyaknya Serangan | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| *Sybil nodes* terdeteksi | 43 | 44 | 49 | 58 | 59 | 67 | 72 | 75 | 79 | 85 |
| Persentase | 43% | 44% | 49% | 58% | 59% | 67% | 72% | 75% | 79% | 85% |

Tabel 4 pengujian dengan 5 honest nodes dan 4 sybil nodes

Pengujian dengan 5 *honest nodes* dan 5 *sybil nodes*

| *Threshold* | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| *Interval Threshold* | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| *Sybil threshold* | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 |
| | | | | | | | | | | |
| Banyaknya Serangan | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| *Sybil nodes* terdeteksi | 25 | 30 | 41 | 43 | 51 | 54 | 63 | 68 | 78 | 85 |
| Persentase | 25% | 30% | 41% | 43% | 51% | 54% | 63% | 68% | 78% | 85% |

Tabel 5 pengujian dengan 5 honest nodes dan 5 sybil nodes

**4.2.** Analisis Hasil Pengujian

Berdasarkan pengujian pada parameter *threshold* dengan pengujian sebanyak 10 kali dengan jumlah *node Sybil* yang berbeda, diperoleh data yang menunjukkan semakin kecil besaran *threshold* maka pendeteksian *node Sybil* akan semakin sensitif dan tingkat keberhasilan sistem mendeteksi bergantung dengan berapa *node Sybil* yang melakukan penyerangan. Dengan persentase terbaik adalah sebesar 97.00% dengan besaran *threshold* masing – masing adalah 2 dan 1, dan jumlah *node Sybil* yang melakukan penyerangan berjumlah satu *node*.

## 5. KESIMPULAN

**5.1.**Kesimpulan

- Telah dilakukan simulasi jaringan yang memiliki *honest node* dan *Sybil node*,
- Telah diimplementasikan deteksi *Sybil node* pada *fog computing* dan dapat membedakan *honest node* dan *Sybil node*,
- Deteksi *Sybil node* yang telah diimplementasikan memiliki akurasi hingga 97.00%.

**5.2.** Saran

Untuk penelitian selanjutnya, dapat dilakukan percobaan dengan protokol lainnya seperti MQTT dan dilakukan penelitian tentang jenis serangan Sybil lainnya seperti SA-1 dan SA-3

## REFERENCE


[1] K. Zhang, X. Liang, R. Lu, and X. Shen, "Sybil attacks and their defenses in the internet of things," *IEEE Internet Things J.*, vol. 1, no. 5, pp. 372–383, 2014.

[2] D. Perera, C., Zaslavsky, A., Christen, P., Georgakopoulos, "Context aware computing for the internet of things: A survey. Communications Surveys Tutorials," *Ieee*, vol. 16, no. 1, pp. 414 – 454, 2014.

[3] C. Lai, H. Li, X. Liang, R. Lu, K. Zhang, and X. Shen, "CPAL: A conditional privacy-preserving authentication with access linkability for roaming service," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 46–57, 2014.

[4] A. Huertas Celdran, F. J. Garcia Clemente, M. Gil Perez, and G. Martinez Perez, "SeCoMan: A Semantic-Aware Policy Framework for Developing Privacy-Preserving and Context-Aware Smart Applications," *IEEE Syst. J.*, vol. 10, no. 3, pp. 1111–1124, 2016.

[5] J. Huang, Y. Meng, X. Gong, Y. Liu, and Q. Duan, "A novel deployment scheme for green internet of things," *IEEE Internet Things J.*, vol. 1, no. 2, pp. 196–205, 2014.

[6] A. A. Aziz, Y. A. Şekercioğlu, P. Fitzpatrick, and M. Ivanovich, "A survey on distributed topology control techniques for extending the lifetime of battery powered wireless sensor networks," *IEEE Commun. Surv. Tutorials*, vol. 15, no. 1, pp. 121–144, 2013.

[7] K. Zhang, R. Lu, X. Liang, J. Qiao, and X. S. Shen, "PARK: A privacy-preserving aggregation scheme with adaptive key management for smart grid," *2013 IEEE/CIC Int. Conf. Commun. China, ICCC 2013*, no. Iccc, pp. 236–241, 2013.

[8] M. Wen, R. Lu, K. Zhang, J. Lei, X. Liang, and X. Shen, "PaRQ: A privacy-preserving range query scheme over encrypted metering data for smart grid," *IEEE Trans. Emerg. Top. Comput.*, vol. 1, no. 1, pp. 178–191, 2013.

[9] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," *MCC'12 - Proc. 1st ACM Mob. Cloud Comput. Work.*, pp. 13–15, 2012.

[10] L. M. Vaquero and L. Rodero-Merino, "Finding your Way in the Fog," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 27–32, 2014.

[11] J. Ni, K. Zhang, X. Lin, and X. S. Shen, "Securing Fog Computing for Internet of Things Applications: Challenges and Solutions," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 1, pp. 601–628, 2018.

[12] J. Shropshire, "Extending the cloud with fog: Security challenges & opportunities," *20th Am. Conf. Inf. Syst. AMCIS 2014*, pp. 1–10, 2014.

[13] C. N. Kumar and N. Satyanarayana, "International Journal of Modern Trends in Engineering and DETECTION OF SYBIL ATTACK USING POSITION VERIFICATION METHOD IN MANETS," 2014.