

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada saat ini teknologi mengalami perkembangan yang sangat pesat. Berbagai ilmu pengetahuan telah dikembangkan guna membantu manusia dalam menyelesaikan berbagai macam masalah yang dialami sehari-hari. Salah satunya yang sedang banyak dibicarakan yaitu *Internet of Things* (IoT). Perkembangan IoT sendiri mengalami kemajuan yang sangat luar biasa dan akan menjadi cikal bakal modernisasi dimasa yang akan datang. IoT merupakan penemuan terbaru yang dikembangkan karena memiliki kelebihan dari segi fungsionalitas dan mendukung kinerja manusia tanpa menggunakan bantuan kabel atau biasa disebut dengan *wireless*[1]. IoT memiliki konsep dimana sebuah objek atau alat memiliki kemampuan konektivitas untuk mentransmisikan atau mengirimkan data melalui jaringan internet tanpa menggunakan bantuan perangkat komputer yang tinggi ataupun manusia. IoT atau sering disebut juga dengan *Internet of Everything* saat ini mengalami banyak perkembangan. Penerapannya dilakukan pada berbagai bidang seperti kesehatan, transportasi, energi dan lingkungan umum. Salah satu contoh penerapan IoT pada bidang lalu lintas yaitu kamera pengawas pada lampu lalu lintas yang dapat diakses melalui *internet* bahkan dapat mendeteksi pengemudi yang melanggar lalu lintas[2].

Internet of Things juga dapat diterapkan pada *backup power supply*, sebagai contoh *monitoring* dan kendali *generator set*, hal ini memudahkan untuk dilakukannya perawatan sehingga meringankan pekerjaan manusia. Kebutuhan daya cadangan mengalami peningkatan yang cukup besar, baik pada perindustrian besar maupun kebutuhan rumah. *Generator* menjadi peranan dan fungsi yang sangat penting dalam penyaluran tenaga listrik pada suatu sistem tenaga listrik. *Generator set* biasanya mengalami masalah pada *starter* nya jika kapasitas pada baterai aki turun, maupun jika bahan bakar habis sampai kosong, ada juga masalah ketika genset lama tidak digunakan, oli pelumas yang melumasi mesin akan kembali kebawah (tangkai oli) karena gaya gravitasi, menyebabkan komponen – komponen mesin bisa berkarat dan tersangkut satu sama lain. Oleh sebab itu, genset harus dilakukan pemanasan mesin seminggu sekali agar sirkulasi oli tetap berjalan. Mesin pada *genset* yang sulit dihidupkan merupakan masalah yang terbilang sangat fatal karena mesin tidak bisa digunakan pada saat dibutuhkan. Meski mesin telah dinyalakan berulang kali namun tetap

saja *generator* tidak bisa menyala. Masalah ini disebabkan karena genset sudah lama tidak digunakan mati mengakibatkan *genset* sangat sulit untuk dihidupkan, masalah ini juga biasa terjadi dikarenakan baterai *genset* yang sudah melemah, karena itu pemantauan yang rutin dan pengisian ulang pada *baterai* genset yang sudah lemah diperlukan. Hal ini tentu membutuhkan pemantauan yang rutin atau secara berkala, dan tentunya membutuhkan tenaga manusia untuk selalu memantau keadaan genset.

Ada banyak peralatan yang perlu dihubungkan dengan sebuah *Smartphone* ataupun Komputer, salah satunya adalah *Generator Set (Genset)*. *Generator* bekerja dengan cara merubah energi gerak (kinetik) menjadi energi listrik, dengan cara melakukan sinkronasi berdasarkan induksi elektro magnetik, setelah rotor diputar oleh penggerak (motor) dengan demikian kutub-kutub pada rotor akan berputar dan dapat menghasilkan arus listrik. Akan tetapi, *Internet of Things(IoT)* biasanya menggunakan *server cloud* atau jaringan awan dimana semua *bandwidth* jaringan bergantung dari *traffic* pada *Internet Service Provider* dan kepadatan jaringan sangat mempengaruhi kelancaran sistem dalam menjalankan suatu fungsi, oleh karena itu komputasi kabut dibuat demi mengatasi masalah yang ada pada jaringan *cloud*[3].

Fog computing memberikan kelebihan berupa kecepatan dan lebih aman karena operasi berlangsung di jaringan ujung yang terdistribusi daripada lokasi yang terpusat, hal ini menyebabkan lebih mudahnya untuk mengidentifikasi potensi ancaman sebelum mempengaruhi seluruh jaringan. Satu lagi kelebihan komputasi kabut adalah kelancaran suatu sistem, dikarenakan server berada dekat dengan pengguna sehingga proses pengambilan keputusan dan pengiriman dilakukan pada server lokal. Tentunya sistem ini juga memiliki kelemahan dibandingkan dengan komputasi awan yaitu biaya yang tinggi serta pemasangan yang cukup sulit maupun rumit. Kelebihan dari komputasi kabut lebih banyak dan dibutuhkan pada era IoT dan industri 4.0 ini dimana semua membutuhkan jaringan *Internet*, sehingga jika pengguna *Internet* semakin banyak maka *bandwidth* yang tersedia semakin sedikit dan rawan mengakibatkan *server* untuk *overload*, sedangkan *Fog Computing* hanya membutuhkan sedikit *bandwidth* untuk banyak data[4].

Quality of service (QoS) mengacu pada teknologi apa pun yang mengelola lalu lintas data untuk mengurangi *packet loss* (kehilangan paket), *latency*, dan *jitter* pada jaringan. QoS mengontrol dan mengelola sumber daya jaringan dengan menetapkan prioritas untuk tipe data tertentu pada jaringan. Jaringan yang baik perlu menyediakan layanan yang dapat

diprediksi dan terukur sebagai aplikasi (seperti suara, video, dan data yang sensitif terhadap keterlambatan) untuk melintasi jaringan. Sebuah kualitas jaringan dapat diukur menggunakan QoS untuk memenuhi persyaratan lalu lintas dari aplikasi sensitif, seperti suara dan video *real-time*, dan untuk mencegah penurunan kualitas yang disebabkan oleh *packet loss*, penundaan dan *jitter*. Suatu jaringan dapat mencapai QoS dengan menggunakan alat dan teknik tertentu, seperti *jitter buffer* dan *traffic shaping*. Bagi banyak organisasi jaringan, QoS termasuk dalam *service-level agreement* (SLA) dengan penyedia layanan jaringan untuk menjamin tingkat kinerja tertentu. Pada penelitian ini, penulis akan membandingkan kualitas layanan pada jaringan ujung menggunakan konsep *fog computing* dengan *cloud computing*.

Berdasarkan latar belakang diatas penulis membuat alat berupa sebuah sistem *Internet of Things* yang dihubungkan pada *generator*. Daya *generator* disesuaikan dengan kebutuhan prodi Teknik elektro universitas trunojoyo madura yaitu sebesar 5kVA (5000 Volt Ampere). *Generator* dapat dikontrol dari jarak jauh dengan menggunakan *Smartphone*. Genset dapat dimatikan dari jarak jauh dan dapat menghemat bahan bakar pada Genset. Selain itu, *IoT* juga digunakan untuk me *monitoring* daya *Output* pada genset. *Monitoring* daya *output* sangat berguna karena jika daya *output* tidak sesuai atau naik turun maka sistem akan memberi peringatan sehingga tidak dapat merusak perangkat yang disambungkan dengan Genset, sekaligus untuk penjadwalan *generator* untuk hidup yang bertujuan untuk *warm-up* mesin sehingga genset tidak akan macet atau tidak berfungsi jika dibutuhkan[5].

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang dijelaskan di atas, maka perumusan masalah yang dapat diambil yaitu:

1. Bagaimana desain sistem *Fog Computing* untuk mengontrol dan memonitoring *Generator Set*?
2. Bagaimana performa *Quality of Service* dari metode *Fog Computing* dibanding *cloud computing*?
3. Bagaimana membuat sistem kontrol *warming up* genset?

1.3 Tujuan dan Manfaat

Tujuan dan manfaat yang diharapkan dengan dilaksanakannya penelitian ini adalah sebagai berikut:

- a. Untuk Membuat sistem fog computing sebagai kontrol dan monitoring generator set

- b. Untuk meningkatkan kualitas *Internet Of Things*, sehingga sistem akan menjadi lebih cepat dan meminimalisasi kegagalan dengan menggunakan *fog computing*.
- c. Untuk membuat sistem *warm-up* generator set.

1.4 Batasan Masalah

Berikut merupakan hal-hal yang menjadi Batasan masalah penulis pada penelitian Skripsi adalah:

- a. Pada penelitian yang akan dibuat menggunakan *Generator Set* (Genset) dengan daya 5 kVa
- b. *Generator Set* (Genset) yang digunakan dibeli dalam bentuk jadi bukan membuat *Generator set*.
- c. Fokus penelitian berfokus pada kualitas layanan dari *Fog Computing* dan otomasi *generator set* terhadap saat yang dibutuhkan.
- d. Maintenance *generator* harus dilakukan manual seperti mengganti oli dan mengisi bahan bakar

1.5 Metodologi Penulisan Skripsi

Metodologi pengerjaan skripsi yang akan dikerjakan memiliki 5 tahap utama. Kelima tahap tersebut antara lain: tahap persiapan, tahap perancangan, pengumpulan dan analisa data, tahap evaluasi dan pembuatan laporan akhir.

- a. Tahap Persiapan
- b. Tahap Perancangan
- c. Pengumpulan dan Analisis Data
- d. Uji Coba dan Training Sistem
- e. Tahap Evaluasi dan Pembuatan Laporan

1.6 Sistematika Penulisan

Berikut adalah sistematika penulisan dalam penelitian ini:

1. Bab I Pendahuluan

Bab ini membahas tentang latar belakang dari masalah dalam pembuatan tugas akhir ini, perumusan masalah, batasan masalah, tujuan yang dicapai, dan sistematika penulisan. Pada bab ini menjelaskan latar belakang penulis memilih judul yang diajukan berdasarkan masalah yang dihadapi.

2. Bab II Teori Penunjang

Bab ini membahas tentang teori-teori pendukung selama melakukan penelitian. Selain itu pada bab ini juga membahas penelitian-penelitian yang dibuat sebelumnya yang berhubungan dengan judul yang diajukan.

3. Bab III Perancangan Sistem

Bab ini membahas tentang metodologi penelitian yang digunakan, tahap-tahap penelitian, blok diagram system dan peralatan yang dibutuhkan berupa hardware maupun software.

4. Bab IV Implementasi dan Uji Coba Sistem

Bab ini berisi tentang hasil pengujian dan analisa yang dilakukan terhadap alat secara keseluruhan. Hal ini dilakukan untuk mengetahui performansi alat, sehingga dapat dilakukan perbaikan dan perkembangan pada masa mendatang.

5. Bab V Penutup

Bab ini merupakan bab terakhir yang berisi kesimpulan yang diperoleh dari hasil pembuatan skripsi serta saran-saran untuk pengembangan lebih lanjut dari alat yang direalisasikan.

BAB II

TEORI PENUNJANG

2.1 Penelitian Terdahulu

Dwiki Nuridhuha, Mochammad Hannats Hanafi Ichsan, Rizal Maulana (2020). Pada Penelitian ini berfokus pada pengukuran kinerja dari nRF24l01 pada lingkungan *Fog Computing*, parameter yang diukur berupa *delay*, *throughput*, *packet loss*, dan *RTT (round trip-time)* dan variasi waktu pengiriman ke *cloud*. Dari pengujian yang telah dilakukan di dapat hasil jarak pengiriman terjauh untuk ada penghalang yaitu 28 meter, sedangkan tanpa halangan 44 meter. Untuk kinerja *delay*, *throughput*, *RTT (round trip-time)* baik tanpa atau ada penghalang dan pada jarak apapun nilai kinerjanya tidak terpengaruh, namun kinerja dari *packet loss* semakin banyak penghalang dan jauhnya pengiriman data maka akan meningkat juga nilai dari *packet loss*. Untuk variasi waktu pengiriman data ke *cloud* baik tanpa atau ada penghalang dan pada jarak apapun tidak mengalami banyak perubahan[6].

Muhammad Al Arifin (2019). Sistem ini ada 5 parameter *equipment* yang dapat di monitoring dalam sistem ini antara lain *volume* solar menggunakan sensor SRF05, tegangan accu genset menggunakan sensor tegangan DC, suhu ruangan menggunakan sensor dht 11, Arus accu menggunakan sensor arus acs712 dan waktu pergantian oli genset menggunakan *relay* omron mk2p 220 VAC. Hasil dari 5 parameter ini mempunyai indikasi *Alarm* masing-masing, untuk mengolah datanya menggunakan Arduino Nano dan *system IoT* menggunakan aplikasi android *Blynk*. Setelah di ujicobakan dapat disimpulkan hasil dari sistem *monitoring* ini keberhasilan mencapai 100 %[7].

Dwiki Aditya Wibowo, Suhanto, Darmadji (2020). Pada penelitian ini, Sistem alat ini memanfaatkan aplikasi android *mobile* sebagai media *visual monitoring* untuk memberikan informasi dari jarak jauh serta mengontrol menghidupkan dan mematikan *generator* set via jaringan *Internet* dalam keadaan *auto* maupun *manual* dimana sudah teintegrasi dengan ESP 32 untuk mengkonfirmasi penghidupan dan pemadaman *generator* set serta informasi data monitoring pada *generator* tersebut. Dengan monitoring dan kontrol jarak jauh *generator* set maka akan lebih efisien dan efektif dibandingkan tanpa monitoring jarak[8].

Y. Chen, E. Sun, dan Y. Zhang (2017). Menyatakan bahwa *fog computing* terdidasar dari kelemahan sistem *Internet of things* yang berbasis *cloud computing*. komputasi

kabut memperkenalkan sistem baru yaitu *layer* baru diantara perangkat dan *cloud*, yang menghasilkan optimisasi dan mempercepat proses komputasi. mereka membuktikan bahwa *transmission delay* dan *processing delay* pada *fog computing* terbukti meningkatkan performa pada proses transmisi juga pemrosesan data nya dengan menggunakan 3-layer model pada *fog computing*[9].

H. J. Hong (2017). Mereka meneliti tentang pengaruh ekosistem *fog computing* dengan menggunakan *real testbed* dan mengevaluasi hasilnya menggunakan skenario berbeda, ketiga skenario tersebut adalah pendistribusian konten pada jaringan yang diuji, *fog computing* yang berasal dari banyak orang(*crowd-sourced*) dan *programmable IoT analytic*. hasil yang didapatkan dengan menggunakan algoritma optimisasi mereka sendiri menghasilkan peningkatan performa masing masing yaitu 30.3%, 20.2% dan 89.4% dibandingkan dengan algoritma dasar. disini membuktikan bahwa meskipun komputasi kabut sudah dapat memberi peningkatan performa, akan tetapi masih dapat lebih di optimalkan lebih baik lagi[10].

N. Mani, A. Singh, and S. L. Nimmagadda (2020). Menerapkan sistem komputasi kabut pada sistem pemantauan kesehatan terpandu untuk manajemen notifikasi *real-time*. Pada paper ini disebutkan bahwa *platform* komputasi baru yang disebut komputasi kabut dapat mengecilkan *latency* saat mengirimkan data atau berkomunikasi pada *server* pusat, yang dapat mempercepat layanan kesehatan. karena urgensi dari apapun yang berhubungan dengan healthcare, *fog computing* dipilih karena mereduksi latensi dan mempercepat proses *monitoring* secara *realtime* dan juga mereduksi biaya penggunaan pada *server cloud*[11].

A. A. Mutlag, dkk. (2019) banyak pelajaran yang berkaitan dengan komputasi kabut. Komputasi kabut dapat dipastikan menurunkan latensi, berbeda dengan komputasi awan. Penelitian ini menunjukkan bahwa simulasi dan eksperimen memastikan pengurangan latensi yang signifikan. Penurunan latensi sangat penting untuk sistem *Internet of Things* pada pelayanan kesehatan karena harus secara *real-time* respon sistemnya[12].

R. K. Naha dkk.(2018) Peneliti ini menyakan bahwa penggunaan cloud computing saja merupakan solusi yang kurang tepat dikarenakan beberapa pengaplikasian IoT membutuhkan *response-time* secara *real-time*, dikarenakan hal ini *fog computing* yang bersifat diantara *cloud* dan perangkat *Internet of Things* diusungkan. *fog computing* bertanggung jawab atas respon cepat untuk komputasi dan penyimpanan data[13].

K. Gasmi, S. Dilek, S. Tosun, and S. Ozdemir. (2022) Pada penilitan yang dilakukan oleh K. Gasmi dkk. disebutkan bahwa *offloading* atau pemindahan beban komputasi yang masif dari *cloud* menuju komputasi kabut akan menguntungkan beberapa

perangkat yang membutuhkan waktu *real-time* atau dapat mempersingkat waktu dikarenakan proses komputasinya yang berada dekat dengan pengguna sekaligus mengatasi kendala penggunaan sumber daya perangkat *Internet of Things* serta mengurangi penundaan total[14].

2.2 Dasar Teori

2.2.1 *Internet Of Things*

Internet Of Things (IoT) adalah sistem komputasi yang saling terhubung melalui jaringan Internet, perangkat IoT saling berkomunikasi dengan perangkat lain dan saling bertukar informasi melalui perangkat sensor, data dari sensor ini yang kemudian diproses untuk mendapatkan hasil seperti *smart recognitions*, *positioning*, monitoring serta *decision making*. *Internet Of Things* bertujuan untuk memperluas manfaat dari konektivitas Internet yang tersambung secara terus menerus yang memungkinkan kita untuk menghubungkan mesin, peralatan, dan benda fisik lainnya dengan sensor jaringan dan aktuator untuk memperoleh data dan mengelola kinerjanya sendiri, sehingga memungkinkan mesin untuk berkolaborasi dan bahkan bertindak berdasarkan informasi baru yang diperoleh secara independen. *Internet Of Things* atau sering disebut IoT adalah sebuah gagasan dimana semua benda di dunia nyata dapat berkomunikasi satu dengan yang lain sebagai bagian dari satu kesatuan sistem terpadu menggunakan jaringan Internet sebagai penghubung. Misalnya CCTV yang terpasang di sepanjang jalan dihubungkan dengan koneksi *Internet* dan disatukan di ruang kontrol yang jaraknya mungkin puluhan kilometer, atau sebuah rumah cerdas yang dapat dikontrol melalui smartphone dengan bantuan koneksi Internet. Pada dasarnya perangkat IoT terdiri dari sensor sebagai media pengumpul data, sambungan Internet sebagai media komunikasi dan server sebagai pengumpul informasi yang diterima sensor dan untuk analisa. Ide awal *Internet Of Things* pertama kali dimunculkan oleh Kevin Ashton pada tahun 1999 di salah satu presentasinya. Kini banyak perusahaan besar mulai mendalami *Internet Of Things* sebut saja *Intel*, *Microsoft*, *Oracle*, dan banyak lainnya. Banyak yang memprediksi bahwa pengaruh *Internet Of Things* adalah “*the next big thing*” di dunia teknologi informasi, hal ini karena IoT menawarkan banyak potensi yang bisa digali. Contoh sederhana manfaat dan implementasi dari *Internet Of Things* misalnya adalah kulkas yang dapat memberitahukan kepada pemiliknya via SMS atau email tentang makanan dan minuman apa saja yang sudah habis dan harus di isi ulang lagi.

2.2.2 Infrastruktur *Internet of Things*

Infrastruktur pada *Internet of Things*(IoT) berhubungan erat dengan pengembangan dan pembangunan infrastruktur cerdas yang menggabungkan teknologi IoT untuk meningkatkan efisiensi, pemeliharaan dan penghematan biaya. Ini mencakup kemampuan untuk memantau serta mengendalikan operasi infrastruktur perkotaan dan juga pedesaan, seperti jembatan, rel kereta api, dan ladang angin lepas pantai. berikut adalah lima jenis dari infrastruktur *Internet of Things* :

a) *Consumer Internet of Things*

Consumer Internet of Things (CIoT) berdasar pada penggunaan IoT pada perangkat dan aplikasi yang digunakan konsumen atau perorangan. Produk CIoT yang umum diantaranya adalah asisten pintar dan peralatan rumah tangga pintar serta perangkat perangkat pintar yang membutuhkan Internet.

Biasanya, solusi CIoT memanfaatkan *Bluetooth* dan *Wi-Fi* untuk konektivitasnya. Teknologi ini menawarkan komunikasi jarak pendek yang cocok untuk penerapan di tempat yang lebih kecil, seperti rumah dan kantor

b) *Commercial Internet of Things*

IoT Komersial merupakan skala infrastruktur yang lebih besar dari *Consumer Internet of Things*, IoT komersial menjangkau ke tempat yang lebih besar seperti supermarket, hotel, fasilitas kesehatan, tempat hiburan, jalan atau taman.

Ada banyak kasus penggunaan untuk IoT komersial, termasuk memantau kondisi lingkungan, mengelola akses ke fasilitas perusahaan, dan menghemat utilitas dan konsumsi di hotel dan tempat besar lainnya. Banyak solusi IoT Komersial diarahkan untuk meningkatkan pengalaman pelanggan dan kondisi bisnis.

c) *Industrial Internet of Things* (IIoT)

Industrial IoT (IIoT) adalah penerapan dari *Internet of Things* pada skala industri yang berfokus untuk mengembangkan sistem industri agar lebih efisien dan produktif. Penerapan IIoT sering ditemukan pada pabrik skala besar dan sering berhubungan erat dengan industri farmasi, pertanian,

logistik dan otomotif. Internet Industri mungkin adalah contoh IIoT yang paling umum dan terkenal.

d) *Internet of Military Things* (IoMT)

Internet of Military Things (IoMT), juga disebut sebagai *Battlefield IoT*, *Internet of Battlefield Things*, atau hanya IoBT. IoMT merupakan penerapan dari IoT pada keperluan militer dan situasi medan perang. IoMT ditujukan untuk meningkatkan kesadaran situasional, memperkuat management resiko, dan meningkatkan waktu respons.

Aplikasi IoMT pada umumnya termasuk menghubungkan senjata, kendaraan militer tentara, *drone*, dan bahkan Pangkalan Operasi Maju melalui sistem yang saling berhubungan dengan jaringan militer yang lebih besar. Selain itu, IoMT menghasilkan data yang dapat dimanfaatkan untuk meningkatkan praktik, waktu respon, peralatan, dan strategi militer.

2.2.3 Arsitektur *Internet of Things*

Internet of Things terdiri dari beberapa bagian yang menjadi arsitektur penyusun. Adapun arsitektur dari *Internet of Things* adalah sebagai berikut:

a. *Things*

Things merupakan objek yang dilengkapi dengan sensor yang mengumpulkan data yang akan ditransfer melalui jaringan dan aktuator yang memungkinkan sesuatu untuk bertindak (misalnya, untuk menghidupkan atau mematikan lampu, membuka atau menutup pintu, menambah atau mengurangi kecepatan putaran mesin dan banyak lagi). Konsep ini termasuk lemari es, lampu jalan, bangunan, kendaraan, mesin produksi, peralatan rehabilitasi, dan segala sesuatu yang dapat dibayangkan. Sensor dalam semua kasus tidak melekat secara fisik pada benda-benda. Sensor mungkin perlu memantau, misalnya, apa yang terjadi di lingkungan terdekat dengan suatu benda.

b. *Gateway*

Gateways adalah sarana yang menyediakan konektivitas antara hal-hal dan bagian cloud dari solusi IoT, memungkinkan *preprocessing* dan pemfilteran data sebelum memindahkannya ke *cloud* (untuk mengurangi volume data untuk pemrosesan dan penyimpanan terperinci) dan mentransmisikan perintah kontrol dari *cloud* ke berbagai hal. Hal-hal kemudian menjalankan perintah menggunakan aktuatornya.

c. *Cloud*

Cloud gateway memfasilitasi kompresi data dan transmisi data yang aman antara gateway bidang dan server *cloud* IoT. Ini juga memastikan kompatibilitas dengan berbagai protokol dan berkomunikasi dengan gateway lapangan menggunakan protokol yang berbeda tergantung pada protokol apa yang didukung oleh gateway.

d. *Streaming data processor*

Streaming data processor berfungsi untuk memastikan transisi input data yang efektif ke danau data dan aplikasi kontrol. Tidak ada data yang sesekali dapat hilang atau rusak.

e. *Data lake*

Data lake merupakan sebuah wadah yang digunakan untuk menyimpan data yang dihasilkan oleh perangkat yang terhubung dalam format alami. Data besar datang dalam kumpulan atau aliran. Ketika data diperlukan untuk wawasan yang bermakna, data itu diekstraksi dari danau data dan dimuat ke gudang data besar.

f. *Big data warehouse*

Big data warehouse merupakan data yang difilter dan diproses yang diperlukan untuk wawasan yang berarti diekstraksi dari danau data ke gudang data besar. Gudang data besar hanya berisi data yang dibersihkan, terstruktur, dan cocok (dibandingkan dengan danau data yang berisi semua jenis data yang dihasilkan oleh sensor). Juga, gudang data menyimpan informasi konteks tentang hal-hal dan sensor (misalnya, di mana sensor dipasang) dan aplikasi kontrol perintah kirim.

g. *Data analytics*

Data analytics dapat menggunakan data dari gudang data besar untuk menemukan tren dan mendapatkan wawasan yang dapat ditindak lanjuti. Ketika dianalisis (dan dalam banyak kasus – divisualisasikan dalam skema, diagram, infografis) data besar menunjukkan, misalnya, kinerja perangkat, membantu mengidentifikasi inefisiensi dan mencari cara untuk meningkatkan sistem IoT (membuatnya lebih dapat diandalkan, lebih banyak pelanggan- berorientasi). Selain itu, korelasi dan pola yang ditemukan secara manual dapat berkontribusi lebih lanjut untuk membuat algoritma untuk aplikasi kontrol.

h. *Machine Learnings*

Dengan pembelajaran mesin, ada peluang untuk membuat model yang lebih tepat dan lebih efisien untuk aplikasi kontrol. Model diperbarui secara berkala (misalnya, seminggu sekali atau sebulan sekali) berdasarkan data historis yang

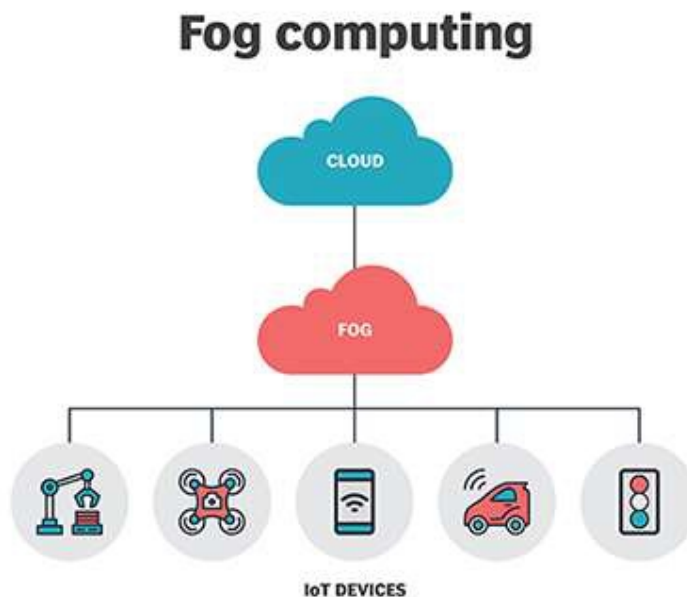
terakumulasi di gudang data besar. Ketika penerapan dan efisiensi model baru diuji dan disetujui oleh analisis data, model baru digunakan oleh aplikasi kontrol.

i. *User application*

User applications adalah komponen perangkat lunak dari sistem IoT yang memungkinkan koneksi pengguna ke sistem IoT dan memberikan opsi untuk memantau dan mengontrol hal-hal cerdas mereka (sementara mereka terhubung ke jaringan hal-hal serupa, misalnya, rumah atau mobil dan dikendalikan oleh sistem pusat). Dengan aplikasi seluler atau web, pengguna dapat memantau keadaan barang-barang mereka, mengirim perintah untuk mengontrol aplikasi, mengatur opsi perilaku otomatis (pemberitahuan dan tindakan otomatis ketika data tertentu berasal dari sensor).

2.2.4 *Fog Computing*

Komputasi kabut, juga dikenal sebagai *fogging*, adalah model di mana data, pemrosesan, dan aplikasi serta perangkat terkonsentrasi pada jaringan ujung atau dekat dengan pengguna. Berkebalikan dengan *cloud* computing dimana semua terfokus pada jaringan *cloud*[15].



Gambar 2. 1 Komputasi kabut

Komputasi kabut adalah salah satu pendekatan untuk menghadapi tuntutan meningkatnya jumlah perangkat yang terhubung ke Internet yang terkadang terjadi *overload*

atau *bandwidth* yang kecil menyebabkan terlambatnya data yang dikirim maupun gagalnya data yang dikirim. Dengan demikian, *Fog* mengurangi latensi layanan, dan meningkatkan *QoS*, menghasilkan pengalaman pengguna yang lebih unggul. *Fog Computing* mendukung aplikasi *Internet of Everything* (IoE) yang muncul yang menuntut latensi *real-time* / dapat diprediksi (otomasi industri, transportasi, jaringan sensor dan aktuator). Berkat distribusi geografisnya yang luas, *Fog Computing* memiliki posisi yang baik untuk data besar secara *real time* dan kecepatan yang tinggi[11].

Selain itu *fog computing* juga dapat memperkecil penggunaan daya dikarenakan tidak terlalu bergantung pada komputasi *server cloud* yang membutuhkan banyak daya. *Fog Computing* berfokus pada pengangkatan sebagian beban kerja dari layanan *cloud* reguler dengan menggunakan sumber daya yang terpusat pada jaringan lokal untuk memberikan pengalaman yang lebih cepat, lebih lancar, dan lebih praktis bagi pengguna[16]. Berikut ini adalah beberapa keunggulan dan kekurangan dari sistem *Fog Computing*.

Keunggulan:

- a. Keamanan yang lebih baik. Dikarenakan sifatnya yang lokal atau dekat dengan pengguna serta proses komputasi dilakukan pada jaringan lokal, sehingga mengurangi potensi ancaman keamanan. Seperti contoh *packet sniffing*, hal ini akan sia-sia dilakukan jika *database* atau semua data penting hanya disimpan di *server* lokal sehingga data yang dikirim pada *cloud* hanyalah data yang akan ditampilkan di web atau bersifat *non-sensitive*.
- b. Semua data dan pemrosesan dilakukan di jaringan lokal atau ujung sehingga mengurangi *bandwidth* yang dibutuhkan serta mengurangi kebutuhan komputasi yang tinggi pada *cloud*
- c. Mengurangi latensi sehingga pengambilan keputusan dilakukan dengan cepat
- d. Lebih mudah untuk dibuat daripada *cloud computing* yang membutuhkan *server* yang besar
- e. Pada dasarnya, *fog node* bersifat *mobile* atau dapat dengan mudah dipindahkan dari suatu tempat ke tempat lainnya
- f. *Fog nodes* dapat ditempatkan pada lingkungan manapun seperti area terpencil

Kelemahan :

- a. Rawan terjadi *IP Spoofing* sehingga dapat terjadi suatu kebocoran data pada jaringan lokal
- b. Konsumsi daya yang lebih dibandingkan dengan *cloud computing* yang sifatnya *decentralized* atau terpusat.

- c. Relatif lebih rumit untuk dibuat akan tetapi lebih murah secara biaya

2.2.5 Basis Data

Basis data (*database*) adalah suatu kumpulan data yang disusun dalam bentuk tabel-tabel yang saling berkaitan maupun berdiri sendiri dan disimpan secara bersama-sama pada suatu media. Basis data dapat digunakan oleh satu atau lebih program aplikasi secara optimal, data disimpan tanpa mengalami ketergantungan pada program yang akan menggunakannya. Basis data adalah kumpulan file yang saling berkaitan. Pada model data relasional hubungan antar *file* direlasikan dengan kunci relasi (*relation key*), yang merupakan kunci utama dari masing-masing *file*, sehingga data tidak akan berubah atau tertimpa karena kesalahan dari pengguna.

Perancangan basis data yang tepat akan menyebabkan paket program relasional akan bekerja secara optimal. Pada sistem *IoT database* sering digunakan sebagai penyimpanan data dari sensor sebagai data logger dan monitoring data. Tiap waktu data akan dikirim ke *server* basis data dan data kemudian dapat diolah menjadi suatu grafik atau data terstruktur serta dapat digunakan sebagai data untuk pemilihan kondisi.

2.2.6 Automatic Transfer Switch

ATS merupakan singkatan dari kata *Automatic Transfer Swicth*, jika dipahami berdasarkan arti kata tersebut maka ATS adalah sakelar yang bekerja otomatis, namun kerja otomatisnya jika sumber listrik dari PLN terputus atau mengalami pemadaman maka sakelar akan berpindah ke sumber listrik yang lainnya misalnya adalah *generator*[17].

Automatic Transfer switch merupakan rangkaian kontrol sakelar power inverter dengan PLN yang sudah *full automatic*. Alat ini berguna untuk menghidupkan dan menghubungkan power inverter ke beban secara otomatis pada saat PLN padam. Pada saat PLN hidup kembali, alat ini akan Memindahkan sumber daya ke beban dari power inverter ke PLN. Dalam perkembangan teknologi dunia elektrikal akhirnya merekayasa hal tersebut kemudian di jalankan secara *Automatic* yang di singkat ATS (*Auto Transfer Swicth*) yang di fungsikan secara otomatis untuk memindahkan daya sesuai dengan kebutuhan tanpa menggunakan tenaga manusia untuk mengoperasikannya. Beberapa jenis ATS di bedakan menurut kapasitas daya yang di butuhkan atau berdasar Phasa dan Ampere yang melalui panel tersebut, namun untuk prinsip kerjanya sama. Pada dasarnya pembuatan ATS adalah memainkan penalaran

logika matematika dengan merangkaikan beberapa alat seperti *Relay*, *Timer*, Kontaktor, dan MCB. Alat – alat tersebut pada prinsipnya adalah sebagai sakelar ataupun pemutus hubungan. Pemakaian panel ATS ini di bedakan pada besar kecilnya pemakaian listrik. Semakin tinggi pemakaian daya listrik, tentunya akan semakin besar pula spesifikasi komponen komponennya terutama *Breaker* dan kontaktornya dan juga ukuran kabelnya[18].

2.2.7 Generator

Genset atau kepanjangan dari *Generator set* adalah sebuah perangkat yang dapat menghasilkan daya listrik. *Genset* terdiri dari pembangkit listrik (*generator*) dengan mesin penggerak yang disusun menjadi satu kesatuan untuk menghasilkan suatu tenaga listrik dengan besaran tertentu. Pada *Genset*, Mesin berfungsi sebagai perangkat pemutar dan generator atau alternator sebagai alat untuk menghasilkan listrik. Kedua perangkat tersebut digunakan untuk menghasilkan arus atau daya listrik, dimana pada perangkat mesin (*motor*) perangkat ini memutar sebuah *generator* yang terbuat dari sekumpulan tembaga. Sehingga pada saat terjadi perputaran terus menerus dengan kecepatan yang stabil kemudian dihubungkan dengan *generator* hingga dapat menghasilkan arus listrik.

2.2.8 Segitiga Daya

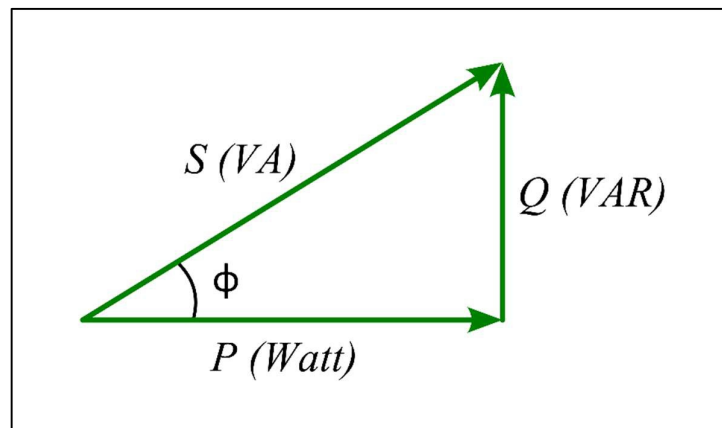
Daya adalah sejumlah energi listrik yang dikeluarkan untuk melakukan suatu kerja atau usaha. Misalkan pada lampu 6 *Watt* dimana memerlukan daya 6 watt untuk meng-konversikannya dari energi listrik ke energi cahaya. Daya listrik biasanya dinyatakan dalam satuan watt atau *Horsepower* (HP) dimana 1 Hp ini setara dengan 746 Watt. Pada listrik searah (DC) daya listrik ini dirumuskan dengan rumus $P = V \times I$. Dimana P adalah daya , V adalah tegangan dan I adalah arus. Pada dasarnya daya listrik dibagi menjadi tiga yaitu:

1. Daya Aktif atau Daya Nyata (*Real Power*) Daya aktif adalah suatu daya yang sesungguhnya terpakai untuk melakukan kerja terhadap beban atau merupakan daya yang sesungguhnya dibutuhkan beban. Daya ini digunakan untuk mengubah suatu energi listrik menjadi bentuk energi lain. Misalkan pada sebuah lampu dimana ada konversi energi listrik menjadi energi cahaya. Satuan dari daya aktif adalah Watt dan daya aktif ini bisa terjadi pada beban induktif maupun beban resistif.

2. Daya Semu Merupakan keseluruhan kapasitas daya yang belum terpakai. Kapasitas daya ini yang disediakan oleh PLN dengan satuan VA (Volt Ampere). Pada persamaan segitiga daya bisa dilihat bahwa daya semu ini tidak terdapat $\cos \phi$.
3. Daya reaktif adalah sebuah daya yang terserap untuk pembentukan medan magnet. Daya ini ditimbulkan oleh beban induktif seperti transformator, motor, dan lain lain. Beban induktif disebabkan oleh lilitan kawat atau kumparan yang digunakan untuk membangkitkan medan magnet agar peralatan listrik dapat bekerja dengan baik. Satuan dari Daya Reaktif ini adalah VAR (*Volt Ampere Reactive*)

Sedangkan Faktor daya atau *power factor* ($\cos \phi$) merupakan suatu nilai yang disebabkan oleh adanya kerugian daya. Dalam perhitungan matematis. $\cos \phi$ ini merupakan perbandingan dari daya aktif (daya yang sebenarnya terpakai) dengan daya semu (total keseluruhan daya). $\cos \phi$ pada instalasi satu fasa biasanya ditetapkan 0,8 karena tidak akan jauh dari nilai tersebut.

Berikut ini merupakan rumus segitiga daya menurut Von Meier Alexander[19] :



Gambar 2. 2 Segitiga daya

$$P = V \times I \times \cos \phi$$

$$S = V \times I$$

$$Q = V \times I \times \sin \phi \dots \dots \dots (1)$$

Dimana :

P : Daya Aktif

V : Tegangan

I : Arus

$\cos \phi$: Faktor Daya

2.2.9 Quality of Service

Quality of Service (QoS) Merupakan kemampuan suatu jaringan untuk menyediakan layanan yang baik dengan menyediakan kapasitas jaringan, mengatasi *jitter* dan *delay* (waktu tunda). QoS dirancang untuk membantu pengguna menjadi lebih produktif dengan memastikan bahwa pengguna mendapatkan kinerja yang handal dari aplikasi-aplikasi berbasis jaringan. QoS merupakan suatu tantangan yang besar dalam jaringan berbasis IP dan internet secara keseluruhan. Terdapat banyak hal bisa terjadi pada paket ketika melakukan perjalanan dari asal ke tujuan, yang mengakibatkan masalah-masalah dan sering disebut sebagai parameter-parameter QoS, Parameter dibawah ini berdasarkan standarisasi dari ETSI TR 101 329-7, berikut adalah parameter dari QoS:

- a. *Throughput, Throughput* yaitu kecepatan (*rate*) *transfer* data efektif, yang diukur dalam bps (*bit per second*). *Throughput* adalah jumlah total kedatangan paket yang sukses yang diamati pada tujuan selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut.

Tabel 2. 1 Tabel *Throughput*

Kategori Throughput	Throughput (bps)	Index
Sangat Bagus	100	4
Bagus	75	3
Sedang	50	2
Jelek	<25	1

Persamaan perhitungan *Throughput* :

$$Throughput = \frac{\text{Paket Data Diterima}}{\text{Lama Pengamatan}} \dots\dots\dots(2)$$

- b. *Packet Loss* merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang dapat terjadi karena *collision* dan *congestion* pada jaringan.

Tabel 2. 2 Tabel *Packet Loss*

Kategori Degradasi	Packet Loss (%)	Index
Sangat Bagus	0	4
Bagus	3	3
Sedang	15	2
Jelek	25	1

Persamaan perhitungan *Packet Loss*:

$$Packet\ Loss = \frac{Paket\ Dikirim - Paket\ Diterima \times 100}{Paket\ Dikirim} \dots\dots\dots (3)$$

- c. *Delay (Latency)* merupakan waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. Delay dapat dipengaruhi oleh jarak, media fisik atau juga waktu proses yang lama.

Tabel 2. 3 Tabel *Delay*

Kategori Delay	Besar <i>Delay</i> (ms)	Indeks
Sangat Bagus	< 150	4
Bagus	150 – 300	3
Sedang	300 – 450	2
Jelek	> 450	1

Persamaan perhitungan *Delay* :

$$Rata - rata\ delay = \frac{Total\ delay}{Packet\ data\ diterima} \quad (4)$$

- d. *Jitter* atau variasi kedatangan paket, *Jitter* diakibatkan oleh variasi-variasi dalam panjang antrian, dalam waktu pengolahan data, dan juga dalam waktu penghimpunan ulang paket-paket diakhir perjalanan *jitter*.

Tabel 2. 4 Tabel *Jitter*

Kategori Jitter	Jitter (ms)	Index
Sangat Bagus	0	4

Bagus	0 – 75	3
Sedang	75 – 125	2
Jelek	125 – 225	1

Persamaan perhitungan *Throughput* :

$$Jitter = \frac{\text{Total variasi delay}}{\text{Packet data diterima}} \dots\dots\dots(5)$$

QoS mengacu pada kemampuan jaringan untuk menyediakan layanan yang lebih baik pada trafik jaringan tertentu melalui teknologi yang berbeda-beda. QoS menawarkan kemampuan untuk mendefinisikan atribut-atribut layanan jaringan yang disediakan, baik secara kualitatif maupun kuantitatif. Pada Tabel I diperlihatkan nilai presentase dari QoS.

Tabel 2. 5 Persentase dan Nilai QoS

Index	Persentase (%)	Nilai
Sangat Bagus	95 – 100	3,8 - 4
Bagus	75 – 94.75	3 – 3,79
Sedang	50 – 74.75	2 – 2,99
Jelek	25 – 49.75	1 – 1,99

2.2.10 Protokol HTTP

HTTP adalah protokol jaringan lapisan aplikasi (*application layer*) yang dikembangkan untuk membantu proses transfer antar komputer. Protokol HTTP menyediakan kumpulan perintah di dalam komunikasi antar jaringan. Komunikasi tersebut berlangsung antara *web server* dengan *client* atau sebaliknya. Di dalam komunikasi ini, komputer client melakukan permintaan dengan mengakses alamat *IP Address* atau domain (URL). Kemudian *web server* mengelola permintaan tersebut dan mengirimkan respon pada *client*. sebagai contoh, Klien mengirimkan permintaan HTTP ke *server*; kemudian *server* mengembalikan respons ke klien. Tanggapan berisi informasi status tentang permintaan dan mungkin juga berisi konten yang diminta. *HTTP Method* terdiri dari:

- a. *GET Method*, GET digunakan untuk meminta data dari sumber daya tertentu. GET adalah salah satu metode HTTP yang paling umum. berikut adalah contoh string query dari *GET method* dikirim dalam URL, permintaan GET: `http://127.0.0.1/test/demo_form.php?nama=bagus&kota=surabaya`
- b. *POST Method*, POST digunakan untuk mengirim data ke *server* , dan *server* membuat / memperbarui sumber daya. Data yang dikirim ke *server* dengan POST disimpan di request body permintaan HTTP.



Gambar 2. 3 Header POST Method

- c. *PUT Method*, PUT digunakan untuk mengirim data ke server untuk membuat / memperbarui sumber daya. Perbedaan antara POST dan PUT adalah bahwa permintaan PUT bersifat idempoten yang artinya, memanggil permintaan PUT yang sama beberapa kali akan selalu menghasilkan hasil yang sama. Sebaliknya, memanggil permintaan POST berulang kali memiliki efek samping berupa pembuatan sumber daya yang sama beberapa kali.
- d. *HEAD Method*, HEAD hampir identik dengan GET, tetapi tanpa body response. Dengan kata lain, jika GET / pengguna mengembalikan daftar pengguna, maka HEAD / pengguna akan membuat permintaan yang sama tetapi tidak akan mengembalikan daftar pengguna. Permintaan HEAD berguna untuk memeriksa permintaan GET apa yang akan dikembalikan sebelum benar-benar membuat permintaan GET – seperti sebelum mengunduh file besar atau isi respons.
- e. *DELETE Method*, metode DELETE digunakan untuk menghapus sumber daya yang ditentukan.
- f. *OPTIONS Method*, metode OPTIONS digunakan untuk menjelaskan opsi komunikasi untuk sumber daya target.