# BAB 2 LANDASAN PUSTAKA

## 2.1 Kajian Pustaka

Kajian pustaka yang digunakan pada penelitian ini adalah penelitian terdahulu yang membahas tentang analisis sistem basis data dan *Internet of Things*.

Soto (2017) dalam penelitiannya yang berjudul "Performance Evaluation Of Scalable And Distributed Iot Platforms For Smart Regions" menjelaskan bahhwa perkembangan era digital akan berkembang sangat cepat dibanding tahun-tahun sebelumnya. Maka dalam penelitiannya dia mensimulasikan *smart regions* untuk mencari performansi *platform* Internet of Things supaya dapat memantau dan mengolah kota dari data-data sensor yang didapat. Hasilnya didapat kesimpulan bahwa smart region dapat diterapkan pada suatu wilayah pada saat ini, dan semakin bertambahnya tahun akan terjadi peningkatan yang signifikan seperti penggunaan sinyal 5G.

Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, dan Marimuthu Palaniswami (2013). Dalam bukunya berjudul "Internet of things (iot): A vision, architectural elements, and future directions. Future generation computer systems" menjelaskan bahwa keseluruhan generasi komputer akan terus berkembang pesat melalui visi dan misi dari stakeholders dan enterprise dimana akan menciptakan dunia yang serba mendekati otomatis.

Andy Stanford-Clark, dan Hong Linh Truong (2013). Dalam penelitiannya berjudul "MQTT For Sensor Networks (MQTT-SN) Protocol Specification" menjelaskan bahwa diperlukannya suatu protokol yang bersifat lightweight supaya dapat mengirim data secara mudah dan ringan melalui jaringan internet, dimana dia menguji MQTT dan MQTT-SN sebagai protokol yang digunakan untuk pengujian dan didapatkan hasil MQTT-SN lebih baik digunakan dalam pengiriman UDP, dan MQTT lebih baik digunakan dalam pengiriman TCP.

Sekarwati (2011) dalam penelitiannya yang berjudul "Perbandingan Perangkat Lunak Database NoSQL, Studi Kasus : Cassandra dan MongoDB" menjelaskan bahwa perkembangan data juga menuntut untuk melakukan inovasi dalam teknologi, terlebih lagi untuk mengatasi banyaknya data, banyaknya node, kebutuhan data center sebagai pusat data, dan masalah fail over. Sehingga dibutuhkan suatu database yang dapat menangani inovasi tersebut. Tujuan dan penelitian ini adaIah menguraikan kelebihan dan kelemahan basis data Cassandra dan MongoDB. Dan didapatkan hasil bahwa MongoDB cenderung unggul dalam kebanyakan hal yang ditelitinya.

Berdasarkan penelitian-penelitian diatas maka penulis dapat menyimpulkan bahwa diperlukan adanya analisis performansi sistem basis data untuk

penyimpanan data sensor IoT yang dapat memberikan informasi secara kuantitas.

## 2.2 Dasar Teori

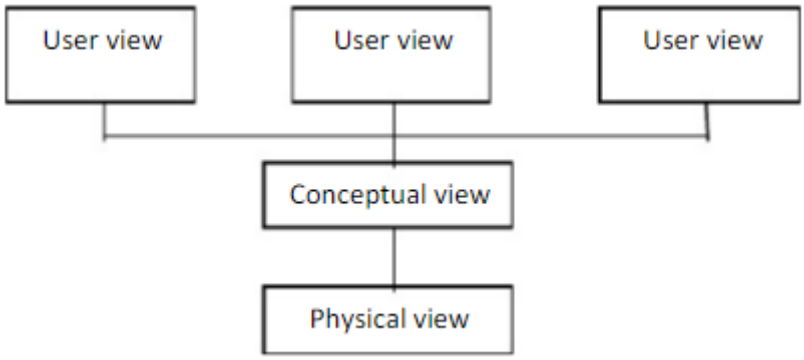
### 2.2.1 Sistem Basis Data

Sistem basis data merupakan sekumpulan basis data dengan para pemakai yang menggunakan basis data secara bersama-sama, personil yang merancang dan mengelola basis data, teknik-teknik untuk merancang dan mengelola basis data, serta sistem komputer yang mendukungnya.

Komponen-komponen utama penyusun basis data adalah perangkat keras, sistem operasi, basis data, sistem pengelola basis data (DBMS),dan pengguna database

Sistem basis data biasanya menyembunyikan detil tentang bagaimana data disimpan dan dipelihara. Data yang terlihat oleh pemakai berbeda dengan yang tersimpan secara fisik.

Abstraksi data merupakan level dalam bagaimana melihat data dalam sebuah sistem basis data. (Arsas Kafri. 2017)




**Gambar 2. 1 Komponen Sistem Basis Data**

### 2.2.2 DBMS (Database Management System)

DBMS (Database Management System) merupakan sekumpulan program yang mengatur ataupun mengkoordinasikan semua kegiatan yang berhubungan dengan basis data. Dengan adanya berbagai tingkatan pandangan dalam suatu basis data maka untuk mengakomodasikan masing-masing pengguna dalam piranti lunak manajemen database biasanya terdapat bahasa-bahasa tertentu yang disebut Data Sub Language.

Sedangkan menurut Kadir (2003) pengertian DBMS itu adalah suatu program komputer yang digunakan untuk memasukkan, mengubah, menghapus, memanipulasi dan memperoleh data/informasi dengan praktis dan efisien.

Hampir sama dituliskan oleh Connolly yang mengartikan DBMS adalah sebuah perangkat lunak yang memungkinkan pengguna untuk mendefinisikan, membuat, mengambil data, dan mengontrol akses kepada database. (Connolly, and Begg, 2010)

### 2.2.3 Not only SQL (NoSQL)

Pada tahun 1998 pertama kalinya dikembangkan sebuah konsep penyimpanan basis data yaitu NoSQL oleh Carlo Strozzi, yang kemudian pada tahun 2009 Eric Evans memperkenalkan kembali teknologi NoSQL. Kehadiran NoSQL bukan berarti untuk menggantikan model RDBMS yang sudah ada. Awal kemunculannya dilatarbelakangi oleh beberapa masalah yang muncul dari RDBMS. NoSQL dan RDBMS memiliki kelebihan dan tempat masing-masing sehingga diharapkan dapat saling melengkapi teknologi penyimpanan basis data. (Mufid Itsnaini Zain dan M. Rudyanto Arief, 2014)

Secara bahasa NoSQL merupakan singkatan dari Not Only SQL yang berarti sistem manajemen basis data tersebut berbeda dengan basis data relasional dalam beberapa aspek. Dalam RDBMS dikenal adanya konsep ACID *(Atomicity, Consistency, Isolation, Durability)*. Sedangkan dalam NoSQL menerapkan konsep BASE (Basically Available, Soft state, Eventually Consistent).

*Basically Available* memastikan bahwa sistem bekerja sepanjang waktu. *Soft state* merupakan suatu kondisi sistem tidak harus konsisten setiap saat. *Eventually Consistent* menekankan bahwa sistem akan menjadi konsisten beberapa waktu kemudian.

### 2.2.4 MongoDB

MongoDB adalah sistem basis data berorientasi dokumen yang menggunakan bahasa pemrograman C++. Objek data disimpan dalam bentuk data BSON. Objek-objek data tersebut tidak harus mempunyai struktur tabel atau kolom yang sama, yang memudahkan dalam penyimpanan karena lebih fleksibel dimana dapat berubah struktur datanya jika pengguna inginkan. Model data disimpan dalam bentuk dokumen yang menjadikannya lebih mudah dalam memperkejakan data karena memetakan objek kedalam kode aplikasi. MongoDB juga menggunakan *ad hoc queries, indexing, and real time aggregation* yang menyediakan layanan untuk mengakses data dan menganalisa data secara akurat. MongoDB juga bersifat *open source* dan gratis untuk digunakan yang berada dibawah agensi GNU Affero General Public License. (P.Sudhakaran, 2016)

### 2.2.5 Load Testing Database

*Performance Testing* atau *Load Testing* adalah proses menjalankan aplikasi dengan mengemulasikan pengguna yang actual dengan beban yang dibuat untuk tujuan menemukan *system bottleneck.* Tujuan utama dari *load testing* ini adalah

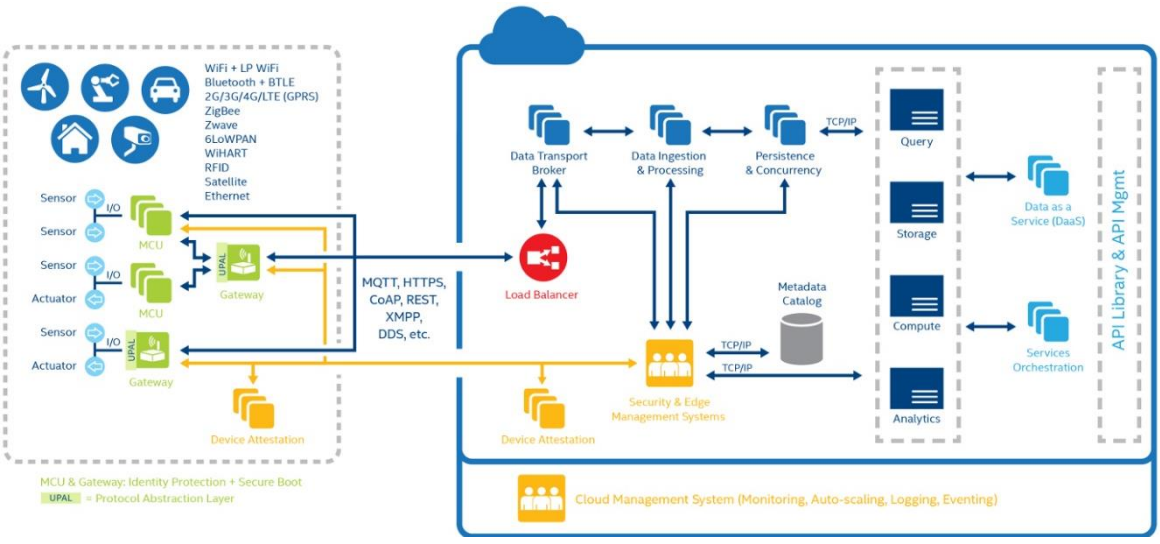
untuk melihat aspek *scalability, availability,* dan performa dari perangkat keras maupun perangkat lunak. Aspek-aspek sumber daya kerja perangkat seperti CPU *usage*, *memory usage, cache coherence, data consistency, power consumption, network banwidth usage,* dan *response time* juga termasuk informasi yang digunakan dalam *load testing.* (H. Sarojadevi. 2011)

### 2.2.6 Internet of Things (IoT)

Internet *Of Things* (IoT) merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus menerus. Suatu benda dikatakan IoT apabila terdapat pada suatu benda elektronik, atau peralatan apa saja yang tersambung ke suatu jaringan lokal dan global melalui sensor yang tertanam dan selalu aktif. Cara kerja dari IoT yaitu dengan memanfaatkan sebuah argumentasi pemrograman yang setiap perintah dari suatu argumen menghasilkan sebuah interaksi dan komunikasi antara sesama mesin yang terhubung secara otomatis dan yang menjadi media penghubung antara perangkat tersebut adalah internet.

Sebuah perangkat IoT memiliki sebuah radio yang dapat mengirim dan menerima koneksi *wireless*. Perangkat-perangkat IoT terhubung dalam sebuah jaringan mesh satu sama lain dan mengirimkan sinyal. Perangkat dalam jaringan mesh mampu untuk mentransfer sinyal sehingga memungkinkan untuk menghubungkan ribuan sensor dalam satu area yang luas, seperti sebuah kampus ataupun kota dan beroperasi dengan selaras.

Internet *of Things* merepresentasikan artitekstur yang hibrida, yang mempunyai arti bahwa IoT dapat memiliki arsitektur subsistem berbeda-beda. Pada umumnya, sistem IoT dibentuk dengan dua arsitektur managemen: Event-Driven dan Time-Based. Arsitektur sensor event-driven mengirim data ketika mereka merasakan aktivitas dari lingkungan luar. Dalam arsitektur time-based , komponen-komponennya terus-menerus melakukan pengiriman data dari berbagai interval waktu. (Vladislavs Aleksandrovičs, Eduards Filičevs, Jānis

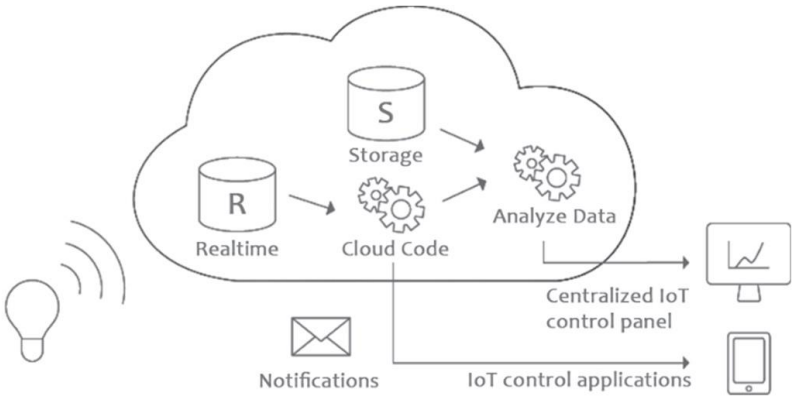



**Gambar 2. 2 Arsitektur pusat managemen sistem IoT**

Kampars, 2016)

**Sumber:** (http://files.linuxgizmos.com/intel_iot_apimgmt.jpg)

Berdasarkan dari infrastruktur IoT, perangkat-perangkat elektronik dapat menggunakan berbagai media penyimpan data dan mekanisme transmisi data. Terdapat alat-alat IoT unyuk menyimpan informasi yang didapatkan dari sensor-sensor secara langsung didalam built-in memorinya. Yang terakhir di atas semua pekerjaan yang bekerja secara otonom dan hanya mengumpulkan informasi penting untuk melakukan aktivitas real-time. Memori internal dari alat-alat tersebut biasanya sangat terbatas dan hanya diperuntukkan sebagai data yang berasal dari pembacaan sensor.




**Gambar 2. 3 Struktur pusat komputasi data dan penyimpanan**

Sekarang, penyimpanan data terpusat lebih digunakan secara ekstensif. Hal ini memungkinkan perangkat-perangkat IoT untuk mengirim data kedalam *server* terpusat dimana datanya dapat disimpan, dianalisa atau dikelola. (Charith Perera, Arkady Zaslavsky, Peter Christen, Dimitrios Georgakopoulos, 2014)

### 2.2.7 Message Queue Telemetry Transport (MQTT)

MQTT atau *Message Queue Telemetry Transport* merupakan protokol data sentris yang menerapkan metode publish-subscribe pada pengiriman datanya. Protokol MQTT memiliki kemampuan pemrosesan data yang rendah sehingga ideal untuk diterapkan pada jaringan sensor network dengan bandwith dan resource terbatas. MQTT menerapkan tipe komunikasi TCP/IP dimana TCP/IP bersifat *connection-oriented* dengan data yang *reliable*.

Metode pengiriman pesan dengan protokol MQTT-SN terbagi menjadi 3 QoS (Quality of Service). QoS 0 dengan mekanisme pengiriman pesan tanpa *acknowledge*. Mekanisme pengiriman pesan QoS 1 menggunakan *acknowledge* sehingga data lebih terjamin untuk sampai pada receiver. Sedangkan QoS 2 menawarkan tingkat keamanan yang paling tinggi dengan mekanisme pengiriman double *acknowledge*. *Acknowledge* yang pertama digunakan untuk memastikan data yang diterima sesuai dengan topik atau tidak. *Acknowledge*

yang kedua berguna sebagai pesan konfirmasi bahwa data berhasil diterima oleh *receiver*. (Stanford-Clark, 2013)

### 2.2.8 *Nodemcu ESP8266-12E Modul Wifi*

Modul *wifi* ESP8266 dikembangkan oleh tim Ai-thinker. Prosesor dari ESP8266 adalah versi lebih kecil dari Tensilica L106 yang memimpin pasar industri *ultra low power* 32-bit MCU *micro*, dengan 16-bit pada mode short, *Clock speed support* 80 MHz, 160 Mhz, mendukung RTOwS, integrasi Wi-Fi MAC/BB/RF/PA/LNA, *on-board* antenna. Modul ini mendukung standar IEEE802.11 b/g/n *agreement*, Protokol TCP/IP *stack* lengkap. Pengguna juga dapat menggunakan modul pada jaringan yang sudah ada, atau membangun controller jaringan secara terpisah. Pada penelitian ini penulis menggunakan Esp8266 12e *development kit* dari Nodemcu yang sudah dilengkapi dengan rangkaian minimum system. (*Espressif Systems et al*. 2015)

Speksifikasi dan bentuk dari Nodemcu Esp8266 12e dapat dilihat pada tabel 2.1 dan gambar 2.2.

**Tabel 2.1.Spesifikasi Nodemcu ESP8266 12E**

| | |
|---|---|
| Wifi Protocol | 802.11 b/g/n |
| Power Supply | 5 V |
| Security | WPA/WPA2 |
| Frequency Range | 2.4G-2.5G (2400M-2483.5M) |
| Network Protocols | IPv4, TCP/UDP/HTTP/FTP |

**Gambar 2. 4 Nodemcu ESP8266 12E**

**Sumber** : (http://adafruit.com/)