

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Sebelum melakukan penelitian yang akan dilakukan, hal yang harus dikerjakan adalah melihat penelitian terdahulu mengenai keterhubungan terhadap penelitian yang akan dilakukan. Tujuan melakukan tinjauan terhadap penelitian terdahulu adalah untuk menjadi landasan dalam menjalankan penelitian yang akan dilakukan dan mengetahui keterhubungan dan kebaruan antara penelitian terdahulu dengan penelitian yang akan dilakukan.

Tabel 2. 1 Penelitian Terdahulu 1 [5]

Penelitian 1	
Nama Jurnal	Simetris : Jurnal Teknik Mesin, Elektro dan Ilmu Komputer (Q3/S3)
Judul Jurnal	Perancangan dan Implementasi <i>IoT (Internet of Things)</i> pada <i>Smarthome</i> Menggunakan <i>Raspberry Pi</i> Berbasis <i>Android</i>
Nama Penulis	Lila Setiyani, Karya Suhada, Yulindawati
Hasil dan Kesimpulan	Dari hasil penelitian yang dilakukan dapat disimpulkan bahwa , board mikrokontroler <i>Raspberry Pi 3</i> model B dapat menghubungkan antara sistem operasi android dengan perangkat lampu dan pendeteksi kebakaran . Dengan pengujian – pengujian yang telah dilakukan produk <i>smart home system</i> yang dibangun peneliti terbukti dapat mengontrol lampu dan dapat memberikan informasi ketika terjadi kebakaran.

Tabel 2. 2 Penelitian Terdahulu 2 [6]

Penelitian 2	
Nama Jurnal	JATI (Jurnal Mahasiswa Teknik Informatika) (Q5/S5)
Judul Jurnal	IMPLEMENTASI <i>IoT(Internet Of Things)</i> MONITORING KUALITAS AIR DAN SISTEM ADMINISTRASI PADA PENGELOLA AIR BERSIH SKALA KECIL
Nama Penulis	Fitri Febrianti, Suryo Adi Wibowo,NurlailyVendyansyah
Hasil dan Kesimpulan	Dengan adanya implementasi <i>IoT(internet of things)</i> <i>monitoring</i> kualitas air dan sistem administrasi pada

	<p>pengelola air bersih skala kecil yang telah penulis buat dapat di ambil kesimpulan :1.Sistem yang dibuat dapat memonitoring kwalitaas <i>pH</i> air nilai 7.00 untuk air mineral, air lemon 5.9, air sabun10.4, dan dengan indikasi lakmus dalam penentuan asam basa pada air, kekeruhan air persentasi pengujian terbesar 1.45%, dan debit air yang masuk beserta tagihan air pada rumah warga. 2.Komunikasi sistem menggunakan modul wifiESP8266 untuk <i>send</i> data pada web <i>sever</i> dari jarak jauh, menggunakan konsep <i>IoT</i>, 3.Website <i>monitoring</i> kualitas air dan sistem administrasi skala kecil dapat berjalan di <i>browser Mozilla Firefox, Google Chrome, Microsoft edge</i>. Tapi pada web browser <i>Microsoft edge</i> masih tidak stabil menampilkan Gambar <i>background</i>.</p>
--	---

Tabel 2. 3 Penelitian Terdahulu 3 [7]

Penelitian 3	
Nama Jurnal	JATI (Jurnal Mahasiswa Teknik Informatika) (Q5/S5)
Judul Jurnal	IMPLEMENTASI IOT (INTERNET OF THINGS) PADA RUMAH BUDIDAYA JAMUR TIRAM PUTIH
Nama Penulis	Tareh Rozzaq Adzdziqui, Yosep Agus Pranoto, Deddy Rudhistiar
Hasil dan Kesimpulan	<p>Dari beberapa tahapan pengujian yang telah dilakukan, menghasilkan beberapa kesimpulan <i>diantaranya</i> :1.Dari hasil pengujian sensor suhu dan kelembapan udara DHT22 diketahui memiliki pembacaan kadar suhu dan kelembapan udara pada <i>baglog</i> jamur tiram putih dengan rata –rata <i>error</i> 1,28% untuk suhu dan 2,93% untuk kelembapan udara2.Dari hasil pengujian sensor kadar <i>pH</i> tanah diketahui memiliki pembacaan kadar <i>pH</i> tanah pada <i>baglog</i> jamur tiram putih dengan rata –rata <i>error</i> 3,46%.3.Dari hasil pengujian sensor ultrasonik diketahui memiliki pembacaan tinggi permukaan air pada wadah air asam / basa dengan rata –rata<i>error</i> 1,59%.4.Dapat hasil pengujian ESP8266 (<i>NodeMcu</i>) diatas bahwa pengujian komunikasi dan pengiriman 10 data menggunakan ESP8266 (<i>NodeMcu</i>) berhasil terhubung kepada <i>access point</i>, server <i>database</i>, dan sukses melakukan pengiriman data ke database.5.Dari hasil pengujian ESP8266 (<i>NodeMcu</i>) diketahui bahwa waktu pengiriman data dan waktu tampil pada <i>website</i> memiliki delay10 detik</p>

Tabel 2. 4 Penelitian Terdahulu 4 [8]

Penelitian 4	
Nama Jurnal	Jurnal Mnemonic (Q3/S3)
Judul Jurnal	IMPLEMENTASI IoT PADA REMOTE MONITORING DAN CONTROLLING GREEN HOUSE
Nama Penulis	Emmalia Adriantantri , Joseph Dedy Irawan
Hasil dan Kesimpulan	Dari hasil pengujian didapatkan bahwa eror rata-rata dari sensor suhu adalah 1,24%, kelembaban 2,18% sedangkan intensitas cahaya 1,14%, sehingga bisa disimpulkan kesalahan sensor tersebut sangat kecil. <i>Wifi</i> modul ESP8266 dapat mengirimkan data secara <i>real time</i> tanpa ada <i>delay</i> . Kemudian dengan menerapkan <i>IoT green house</i> dapat dimonitor dan dikontrol dari jarak jauh melalui jalur internet.

Tabel 2. 5 Penelitian Terdahulu 5 [9]

Penelitian 5	
Nama Jurnal	<i>International Journal of Innovative Technology and Exploring Engineering</i>
Judul Jurnal	<i>IoT implementation and management for smart farming</i>
Nama Penulis	Nagaraju, Priyanka Chawla
Hasil dan Kesimpulan	Penelitian ini terpilih dikelompokkan berdasarkan berbagai domain dan <i>subdomain</i> yang terkait dengan penggunaan sensor, aktuator, teknologi komunikasi, kontrol energi, solusi penyimpanan, analisis data untuk pengambilan keputusan, dan visualisasi kepada petani melalui aplikasi web. Pada penelitian ini melakukan digitalisasi terhadap agrikultur dengan menggunakan perangkat <i>IoT</i> dengan mengintegrasikan setiap perangkat ke sebuah sistem

Tabel 2. 6 Penelitian Terdahulu 6 [10]

Penelitian 6	
Nama Jurnal	<i>Computers, materials & continua</i>
Judul Jurnal	<i>Information-Centric IoT-Based Smart Farming with Dynamic Data Optimization</i>
Nama Penulis	Souvik Pal, Hannah VijayKumar, D. Akila, N. Z. Jhanjhi, Omar A. Darwish and Fathi Amsaad
Hasil dan Kesimpulan	Penelitian ini merancang <i>Smart Farming</i> berbasis <i>IoT</i> yang Berpusat pada Informasi dengan <i>Dynamic Data Optimization</i> (ICISF-DDO), yang meningkatkan kinerja infrastruktur <i>smart farming</i> dengan konsumsi

	energi yang minimal dan masa pakai yang lebih baik. Di sini, kerangka konseptual dari skema yang diusulkan dan statistik model desain telah didefinisikan dengan baik.
--	--

Tabel 2. 7 Penelitian Terdahulu 7 [11]

Penelitian 7	
Nama Jurnal	Sensors /MDPI
Judul Jurnal	<i>A Systematic Review of IoT Solutions for Smart Farming</i>
Nama Penulis	Emerson Navarro , Nuno Costa, António Pereira.
Hasil dan Kesimpulan	Studi ini bertujuan untuk mengidentifikasi bagaimana <i>IoT</i> digunakan dengan pertanian cerdas dengan menyajikan tinjauan sistematis tentang keadaan seni adopsi <i>IoT</i> dalam pertanian cerdas dan mengidentifikasi perangkat keras, platform, protokol jaringan yang paling umum digunakan. dan teknologi dan penerapannya pada solusi yang diusulkan. Penggunaan <i>Decision Support Sistem</i> untuk pengolahan data dengan <i>big data computing</i> kemudian menampilkan perbedaan protokol <i>network</i> yang disimulasikan untuk solusi perangkat <i>IoT</i> untuk <i>smart farming</i>

Tabel 2. 8 Penelitian Terdahulu 8 [3]

Penelitian 8	
Nama Jurnal	Sustainability/MDPI
Judul Jurnal	<i>A Low-Cost Platform for Environmental Smart Farming Monitoring System Based on IoT and UAVs</i>
Nama Penulis	Faris A. Almalki , Ben Othman Soufiene, Saeed H. Alsamhi, Hedi Sakli
Hasil dan Kesimpulan	Pertanian cerdas melibatkan integrasi teknologi canggih ke dalam praktik pertanian yang ada untuk meningkatkan efisiensi produksi dan kualitas produk pertanian. Evolusi <i>IoT</i> dan <i>UAV</i> telah memungkinkan visi pertanian cerdas berkelanjutan, di mana teknologi cerdas ini telah terbukti meningkatkan kualitas hasil panen dan mengurangi jejak lingkungan dari sektor pertanian.

Tabel 2. 9 Penelitian Terdahulu 9 [12]

Penelitian 9	
Nama Jurnal	Precision Agriculture / Springerlink
Judul Jurnal	<i>Architecture design approach for IoT-based farm management information systems</i>
Nama Penulis	Ö. Köksal, B. Tekinerdogan
Hasil dan Kesimpulan	Tujuan dari penelitian ini adalah untuk berkontribusi pada Miss mutakhir saat ini dengan meningkatkan pendekatan desain arsitektur saat ini untuk FMI berbasis <i>IoT</i> . penelitian tersebut bertujuan untuk memberikan metode perancangan arsitektur untuk merancang Miss berbasis <i>IoT</i> . Pendekatan yang disajikan mengadopsi pendekatan analisis domain berbasis fitur untuk memodelkan berbagai persyaratan pertanian cerdas.

Tabel 2. 10 Penelitian Terdahulu 10 [13]

Penelitian 10	
Nama Jurnal	Sensors /MDPI
Judul Jurnal	<i>A Framework for Malicious Traffic Detection in IoT Healthcare Environment</i>
Nama Penulis	Faisal Hussain , Syed Ghazanfar Abbas, Ghalib A. Shah , Ivan Miguel Pires , Ubaid U. Fayyaz , Farrukh Shahzad , Nuno M. Garcia, Eftim Zdravevski.
Hasil dan Kesimpulan	Hasil eksperimen menunjukkan keefektifan kerangka kerja yang diusulkan untuk mengembangkan solusi keamanan sadar konteks <i>IoT</i> yang efisien. Selain itu, kerangka kerja yang diusulkan dan kumpulan data yang dihasilkan sangat membantu para peneliti untuk mengejar metode yang diusulkan untuk mengembangkan solusi keamanan sadar konteks yang lebih kuat, terutama untuk lingkungan perawatan kesehatan <i>IoT</i> . Selanjutnya, dengan bantuan kerangka kerja yang diusulkan, para peneliti dapat dengan cepat menghasilkan lalu lintas kasus penggunaan <i>IoT</i> lainnya untuk mengembangkan solusi keamanan berbasis <i>AI</i> untuk kasus penggunaan <i>IoT</i> lainnya.

Dari 10 penelitian terdahulu yang telah disebutkan, relevansi pada penelitian yang dilakukan adalah penggunaan perangkat *IoT* terhadap topik masalah yang dirumuskan masing-masing. Kemudian terdapat beberapa penelitian terdahulu

yang menggunakan mikrokontroler ESP8266 untuk mendeteksi suhu dan kelembaban satu ruangan yang menjadi inspirasi untuk menggunakan perangkat mikrokontroler ESP walaupun dengan jenis yang berbeda yaitu ESP32 yang digunakan pada penelitian ini. Kemudian perbedaan dari 10 penelitian terdahulu jika dibandingkan dengan penelitian ini adalah belum adanya penggunaan MLX90640 untuk melakukan pendeteksi suhu objek yaitu hewan ternak kambing untuk diaplikasikan ke sebuah kandang kambing.

2.2 Teori Tentang Skripsi

2.2.1 Peternakan

Peternakan merupakan suatu usaha atau pekerjaan untuk mengembangbiakkan serta pembudidayaan hewan yang bertujuan untuk dapat dimanfaatkan hasilnya. Berdasarkan Pasal 1 ayat (1) UU Nomor 18 Tahun 2009 tentang Peternakan dan Kesehatan Hewan, Peternakan adalah segala urusan yang berkaitan dengan sumber daya fisik, benih, bibit, dan/atau bakalan, pakan, alat, dan mesin peternakan, budi daya ternak, panen, pascapanen, pengolahan, pemasaran, dan pengusahaannya.

Terdapat 2 jenis peternakan berdasarkan ukuran hewannya;

- Peternakan hewan besar: contohnya sapi, kerbau, dan kuda
- Peternakan hewan kecil: contohnya ayam, bebek, dan kelinci.

A. Peternakan Kambing

Kambing merupakan hewan mamalia yang memiliki karakteristik bertanduk, berbulu, dan memiliki susu dan daging yang bisa dikonsumsi oleh manusia. Hewan kambing merupakan hewan yang sangat dibutuhkan dalam kehidupan manusia karena banyak bagian dari tubuh kambing yang bisa dimanfaatkan. Beberapa hasil yang bisa diperoleh adalah untuk penjualan induk, anakan, daging, dan susu. Serta susu kambing mengandung nilai gizi yang tinggi di antara lain terdapat kandungan protein sebesar 3.4%, lemak 4.1%, karbohidrat 5.2%, kalsium sebesar 120

mg/ 100 gram, fosfor 135 mg/ 100 gram dan berbagai macam vitamin yang baik bagi tubuh manusia [14].

2.2.2 Penyakit Kambing

Dalam peternakan kambing, kesehatan hewan adalah aspek yang sangat penting untuk mencapai produktivitas yang optimal. Suhu lingkungan merupakan salah satu faktor lingkungan yang sangat mempengaruhi kesehatan kambing. Perubahan suhu yang ekstrim dapat menyebabkan stres pada kambing, yang pada gilirannya dapat melemahkan sistem kekebalan tubuh dan membuat kambing lebih rentan terhadap penyakit.

Suhu lingkungan yang tidak ideal dapat mempengaruhi kesehatan kambing dalam beberapa cara. Ketika suhu lingkungan terlalu tinggi, kambing dapat mengalami stres panas. Gejala stres panas pada kambing meliputi peningkatan pernapasan, nafsu makan menurun, produksi susu menurun, dan dalam kasus yang parah, dapat menyebabkan kematian. Stres panas dapat menekan sistem kekebalan tubuh, membuat kambing lebih rentan terhadap infeksi bakteri dan virus. Kemudian suhu yang sangat rendah juga dapat menyebabkan masalah kesehatan. Kambing yang terpapar suhu dingin ekstrem dapat mengalami hipotermia. Stres dingin juga dapat mengurangi nafsu makan dan produktivitas serta meningkatkan risiko penyakit pernapasan.

Terdapat beberapa penyakit kambing yang dapat dilihat gejalanya dengan melihat suhu dari tubuh kambing, salah satunya adalah *antrax* dan *blackleg*.

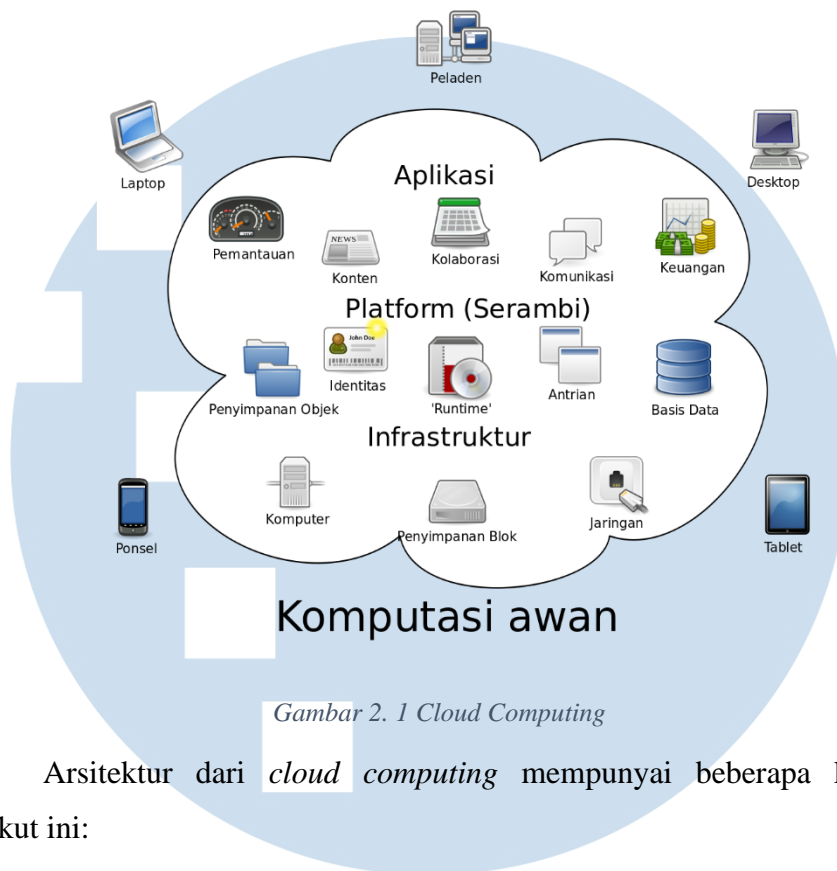
A. *Antrax* : *Antrax* adalah penyakit infeksi akut yang disebabkan oleh bakteri *Bacillus anthracis*. Penyakit ini dapat menyerang berbagai hewan termasuk kambing dan dapat ditularkan ke manusia, menjadikannya penyakit *zoonosis*. *Antrax* seringkali ditemukan di daerah dengan kondisi lingkungan yang mendukung kelangsungan hidup spora bakteri, seperti tanah yang terkontaminasi. Gejalanya adalah demam panas yang dialami oleh kambing.

B. *BlackLeg* : *Blackleg* adalah penyakit infeksi akut yang sangat mematikan pada ternak, termasuk kambing. Penyakit ini disebabkan oleh bakteri *Clostridium chauvoei*, yang merupakan bakteri anaerobik pembentuk spora. *Blackleg* terutama menyerang hewan muda yang sehat, dan sering kali berakibat fatal. Penyakit ini dikenal karena menyebabkan nekrosis otot yang parah dan pembengkakan pada anggota tubuh yang terkena. Gejalanya adalah suhu tubuh di bawah normal ($<38.5^{\circ}\text{C}$), yang dialami oleh kambing.

2.2.3 Cloud Computing

Cloud Computing merupakan sebuah proses komputasi dengan menggunakan teknologi internet dimana nantinya proses tersebut akan dilakukan oleh sebuah komputer inti (*shared computer*) dan hasil komputasi tersebut akan dikirim ke pengguna akhir sesuai dengan permintaan pengguna. Dalam istilah lain, *cloud computing* berarti melakukan akses data dan penyimpanan data melalui internet, bukan menggunakan *hard drive* komputer dari pengguna, dan *Cloud* yang dimaksud adalah sebuah analogi dari internet [15].





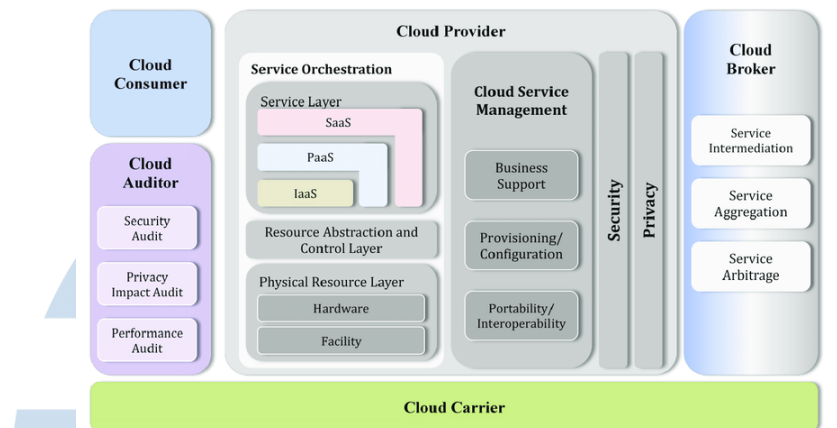
Gambar 2. 1 Cloud Computing

Arsitektur dari *cloud computing* mempunyai beberapa komponen berikut ini:

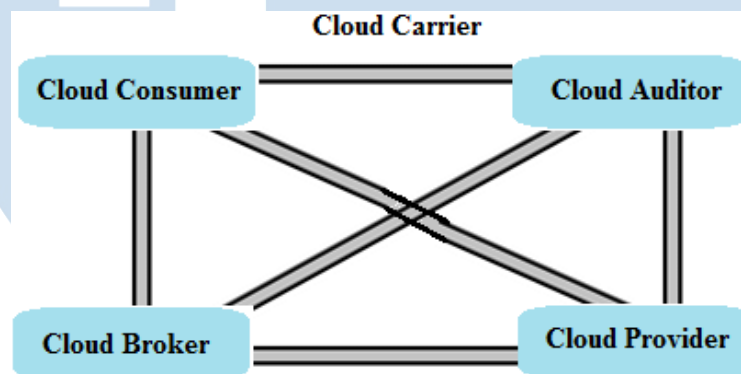
1. *Cloud Carrier*
2. *Cloud Provider*
3. *Cloud Consumer*
4. *Cloud Auditor*
5. *Cloud Broker*

Untuk penggambaran hubungan dari masing-masing komponen *cloud computing* terdapat pada gambar 2.2 dan isi dari tiap-tiap komponen *cloud computing* terdapat pada gambar 2.3.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 2. 2 Cloud Provider



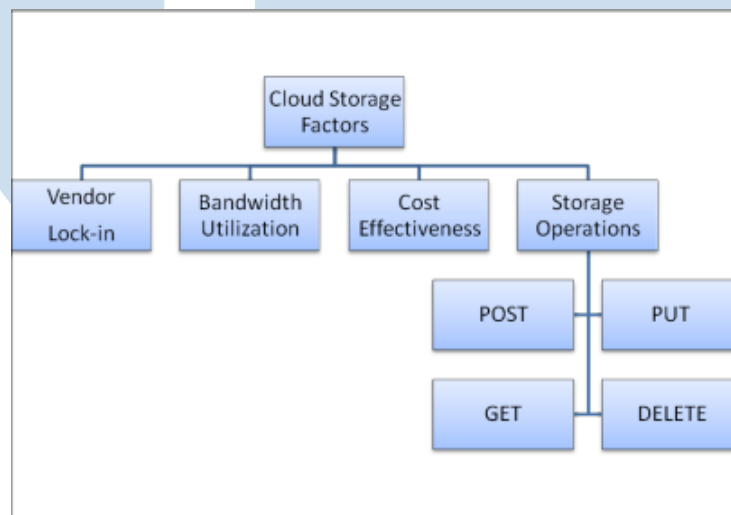
Gambar 2. 3 Cloud Carrier

Cloud Storage merupakan sebuah cabang ilmu dari *Cloud Computing*, yang menjelaskan tentang sebuah penyimpanan data atau *database* yang berbasis teknologi *Cloud*. Penggunaan *Cloud Storage* di zaman yang modern ini sangat banyak digunakan dikarenakan keuntungan yang diperoleh, seperti kemudahan dalam menyimpan sebuah data ke suatu *server database* tanpa perlu khawatir terhadap peletakan server untuk *database*.

Menurut website baktikominfo.id, *Cloud storage* merupakan media penyimpanan *file* berbasis *online* atau digital yang mengandalkan koneksi internet untuk akses data. *Cloud storage* memiliki banyak kegunaan dan kelebihan jika dibandingkan dengan media penyimpanan konvensional seperti *hardisk* dan *flashdisk*.

Proses penyimpanan data pada media penyimpanan *cloud* dapat diartikan sebagai pengiriman salinan ke penyedia layanan *cloud* atau *hosting* melalui Internet. *Server* kemudian menyediakan akses untuk mengunduh atau mengubah data jika ingin membukanya. Layanan *cloud storage* ini biasanya tidak hanya mengandalkan satu *server* saja. Sebagian besar aplikasi di penyimpanan *cloud* juga sudah terinstal di media *internet*, jadi tidak perlu *menginstalnya*.

Terdapat 4 faktor utama dari kemampuan *cloud storage* menurut [16]. :



Gambar 2. 4 Faktor pada Cloud Storage

1) Vendor Lock-In

Fitur seperti distribusi data melalui geo-lokasi, Antarmuka Program Aplikasi (API) tertentu, dll. adalah beberapa yang ditawarkan oleh sebagian besar Penyedia Penyimpanan *Cloud* saat ini. Masalah penguncian vendor ini dapat diselesaikan melalui penggunaan sistem penyimpanan *multi-cloud* di mana data pengguna tidak hanya terikat pada satu penyedia penyimpanan tetapi juga didistribusikan melalui beberapa *cloud* yang tersedia. Ini juga menguntungkan pengguna dalam

pengurangan penyimpanan data pengguna dan peningkatan ketersediaan data melalui *cloud*

2) *Bandwidth Utilization*

Transmisi data melalui jaringan melalui beberapa media dikenal sebagai *bandwidth* atau kadang-kadang disebut sebagai *throughput*. Dikarenakan faktor lain seperti *latency*, kinerja bisa lebih baik jika kapasitas *bandwidth* jaringan lebih tinggi. Permintaan *bandwidth* telah meningkat dalam beberapa tahun terakhir karena alasan bahwa baik industri dan pengguna akhir mulai lebih mengandalkan penyimpanan *cloud* daripada perangkat lunak dan perangkat keras internal mereka. Karena biaya *bandwidth* yang meningkat demikian juga sumber daya lain di *cloud* Terlepas dari beban kerja *input/output* (I/O) yang berat dan latensi akses, penyedia *cloud* harus mempertahankan tingkat keberlanjutan *bandwidth* I/O dari data besar, yang kini menjadi tantangan yang lebih besar bagi mereka. Untuk mencapai tantangan tersebut, *cloud storage* perlu menjalankan beberapa proses secara paralel yang selanjutnya membutuhkan utilisasi *bandwidth* yang lebih tinggi.

3) *Cost Effectiveness*

Munculnya penyedia *cloud* yang lebih baru dengan skema harga yang lebih rendah dan fitur yang lebih canggih tetap menjadi kejadian yang semakin umum di lingkungan *cloud*.

Ini menempatkan pengguna *cloud* dalam keadaan pengambilan keputusan, baik untuk mentransfer dan memigrasikan semua data mereka ke *cloud* baru atau tetap menggunakan yang sudah ada. Beberapa mungkin dikenakan biaya peralihan yang tinggi dan bermigrasi ke *cloud* lain yang lebih baru, sementara yang lain mungkin tetap menggunakan yang lama dan menghemat biaya peralihan yang tinggi.

Perpindahan waktu dari satu *cloud* ke *cloud* lainnya sangat memengaruhi faktor biaya karena beralih lebih awal bisa lebih buruk dalam beberapa kasus dan beralih nanti bisa kehilangan nilainya. Oleh karena itu, pengguna dapat memanfaatkan dan memanfaatkan penyedia baru dan lama dengan kinerja terbaik dengan harga lebih rendah melalui distribusi data mereka melalui beberapa penyedia *cloud*, yang hanya memerlukan sebagian kecil data yang akan dialihkan.

4) *Storage Operations*

Semua harga di penyimpanan *cloud* didasarkan pada tingkat penyimpanan tetap dan tingkat penggunaan jaringan. Oleh karena itu, operasi yang sedang dilakukan atau diminta untuk data yang disimpan melalui penyimpanan *cloud* memainkan peran penting dalam hal ini. Ada empat jenis operasi utama yang dilakukan di penyimpanan *cloud*: POST, GET, PUT, dan DELETE.

Dengan penyimpanan *cloud*, risiko korupsi atau kehilangan data sangat rendah karena data disimpan di server oleh *hosting*. Selain itu, tidak ada perangkat keras yang diperlukan untuk menggunakan penyimpanan *cloud*, karena penyimpanan *cloud* hanya bergantung pada koneksi *internet* untuk akses. Saat ini, dengan begitu banyak media penyimpanan *online* gratis. Penyimpanan *cloud* juga merupakan media penyimpanan yang ramah lingkungan karena tidak memerlukan perangkat dan tidak meninggalkan limbah.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

2.2.4 *Internet of Things*

Internet of Things merupakan sebuah konsep yang dapat menghubungkan semua perangkat dengan menggunakan teknologi *internet* dan memungkinkan untuk memiliki kemampuan seperti mengumpulkan dan mentransfer sebuah data melalui jaringan *internet* secara *cloud* (tanpa intervensi manual). Menurut [2], “*Thing*” pada konteks *IoT* dapat berupa perangkat apa saja dengan *sender* internal apapun yang memiliki kemampuan untuk mengumpulkan dan mentransfer data melalui jaringan tanpa intervensi manual. Teknologi tertanam dalam objek membantu perangkat *IoT* untuk berinteraksi dengan keadaan internal dan lingkungan eksternal, yang pada gilirannya membantu dalam proses pengambilan keputusan.

Pengembang aplikasi *IoT* mengirimkan aplikasi mereka ke penguji bersama dengan dokumentasi yang berisi standar, logika, kesalahan, dan pengecualian yang ditangani oleh pengembang itu sendiri. Sekali lagi, jika ada masalah dengan aplikasi, penguji akan menghubungi pengembang aplikasi. Ini membutuhkan beberapa iterasi dan dapat membangun aplikasi pintar dengan cara ini.

Terdapat beberapa manfaat yang diberakan dari penggunaan *IoT*, salah satunya adalah dapat mengontrol sebuah perangkat dari jarak yang cukup jauh serta dapat langsung menghubungkan dan mengintegrasikan semua perangkat ke sistem berbasis komputer. Beberapa manfaat yang ada pada *IoT* menurut Sidhiq:

- *Improved Customer Engagement* – *IoT* dapat meningkatkan pengalaman pengguna dengan mengotomatisasikan segala tindakan. Untuk contohnya, masalah apa pun di mobil akan terdeteksi secara otomatis oleh sensor. Pengemudi, serta pabrikan, akan diberitahu tentang hal tersebut. Hingga pada waktu pengemudi akan mencapai masa servis dan akan

melakukan servis, pabrikan akan dapat memastikan bahwa bagian yang kemungkinan rusak telah tersedia di bengkel.

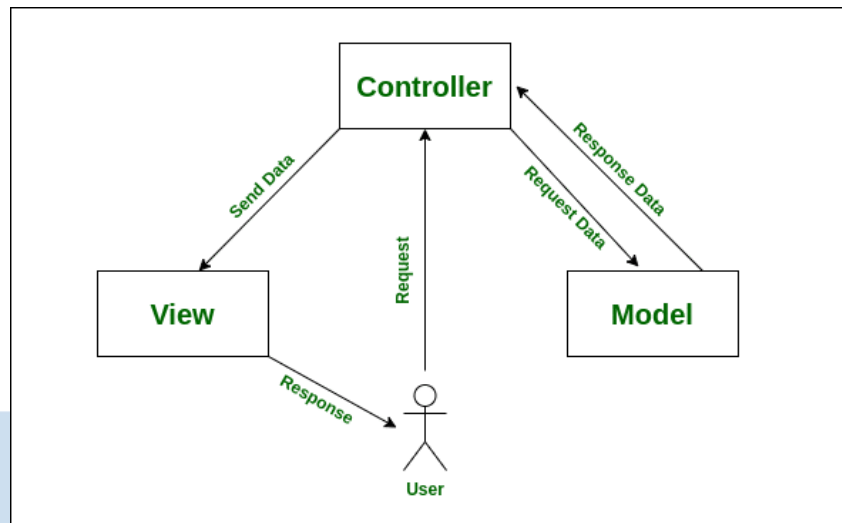
- *Technical Optimization* – IoT telah membantu banyak dalam meningkatkan kegunaan teknologi dan membuatnya menjadi lebih baik. Pabrikan dapat mendapatkan data dari sensor mobil yang berbeda dan menganalisisnya untuk meningkatkan desain dan membuatnya menjadi lebih efisien.
- *Reduce Waste* – Wawasan kita saat ini masih bisa terbilang dangkal, namun IoT menyediakan informasi *rela-time* yang mengarah ke pengambilan keputusan yang efektif dan pengelolaan sumber daya. Sebagai contohnya, jika pabrikan menemukan kesalahan pada banyak mesin, pabrikan tersebut dapat melacak pabrik pembuatan mesin tersebut dan dapat memperbaiki masalah dengan sabuk manufaktur.

2.3 Teori tentang Framework yang digunakan

2.3.1 *Laravel*

Laravel merupakan sebuah *framework* berbasis Bahasa pemrograman PHP yang paling sering digunakan dalam pengembangan sebuah aplikasi *website*. Menurut [17], *Laravel* dapat menyediakan fitur-fitur yang sangat berguna sebagai lapisan abstrak basis data ekspresif dan injeksi *depedency* dan sangat dapat diskalakan.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 2. 5 Penjelasan tentang MVC

Laravel menggunakan MVC arsitektur, yang dimana terdiri dari 3 bagian, yaitu terdapat *Model*, *View*, dan *Controller*. Komponen-komponen ini sangat mudah untuk dikonfigurasi dalam menangani aspek pengembangan khusus dari aplikasi. Pada gambar 2.5. kita dapat melihat bahwa setiap komponen terhubung untuk dapat menjalankan sebuah aplikasi. Komponen *Model* memiliki fungsi untuk menyiapkan, mengatur, memanipulasi, dan mengorganisasikan data yang ada di basis data. Komponen *view* berfungsi untuk menampilkan respon informasi aplikasi kepada pengguna dalam bentuk *Graphic User Interface* (GUI). Terakhir, Komponen *Controller* memiliki tugas untuk menghubungkan dua komponen sebelumnya yaitu *model* dan *view* agar aplikasi dapat berjalan dengan baik.

Dalam *framework laravel* terdapat sebuah komponen yang memiliki tugas untuk menyampaikan informasi ke pengguna menggunakan bahasa *HTML*, yaitu *Blade*. *Blade* merupakan *framework* untuk *front-end* aplikasi pada *laravel*. *Blade* sendiri telah tersedia pada *framework laravel* dan dapat dengan mudah digunakan karena didukung dengan konsep MVC pada *laravel*. Cara membuat *blade* cukup mudah dengan cara manual membuat *file* dengan format nama seperti “*nama_view.blade.php*” [18].

2.3.2 Tailwind

Tailwind merupakan sebuah *framework* CSS yang digunakan untuk menyediakan keperluan dasar membangun komponen tampilan *website*, seperti pengaturan margin, ukuran objek, posisi, warna, dan lain-lain [19]. *Tailwind* juga dapat dikombinasi dengan *framework* CSS lainnya seperti *Bootstrap*. Penggunaan *tailwind* juga dibidang cukup mudah karena mirip seperti *Bootstrap*, yaitu dengan menambahkan atribut pada elemen *class* di HTML.

2.3.3 PostgreSQL

Database PostgreSQL merupakan salah satu alternatif solusi bagi pengguna *database* yang mendukung banyak platform dan bebas lisensi [20]. *PostgreSQL* termasuk ke dalam *database* server yang sangat andal dengan fitur-fitur pendukungnya yang berguna dalam melakukan pengembangan aplikasi, sehingga *database* postgresql ini begitu banyak yang menggunakannya dan ideal sebagai media penyimpanan data sebuah aplikasi. *PostgreSQL* dikembangkan oleh *University of California* di *Berkeley Computer Science Department* [21].

2.4 Teori tentang Tools/Software yang digunakan

2.4.1 Visual Studio Code

Visual Studio Code (VS Code) adalah aplikasi kode editor yang dibuat oleh *Microsoft*. Dengan fitur lengkap dan ekstensi yang luas, *VS Code* menjadi pilihan utama para pengembang. Berikut beberapa hal tentang *Visual Studio Code*:

- 1) **Fitur Lengkap dan Gratis:** VS Code memiliki banyak fitur yang memudahkan pengembangan, dan yang terbaik adalah semuanya gratis.
- 2) **Ringan:** Aplikasi ini dirancang agar se-ringan dan se-nyaman mungkin sehingga pengguna tidak terlalu membutuhkan perangkat berspesifikasi tinggi.
- 3) **Mendukung Banyak Bahasa Pemrograman:** VS Code bisa dijalankan untuk membuat atau mengedit kode sumber berbagai

programming language, seperti *Node.js*, *JavaScript*, *TypeScript*, dan masih banyak lagi.

- 4) Performa Cepat: VS Code menawarkan performa yang cepat dan responsif.
- 5) Multiplatform: Selain mendukung *Windows*, VS Code juga dapat digunakan di Mac OS, Linux, dan sistem operasi lainnya.

2.4.2 Apache

Apache HTTP Server adalah perangkat lunak server web sumber terbuka yang populer dan kuat. Ini digunakan oleh jutaan situs web di seluruh dunia untuk menyajikan konten web kepada pengguna. *Apache HTTP Server*, atau hanya disebut sebagai *Apache*, adalah perangkat lunak server web yang dikembangkan dan dipelihara oleh *Apache Software Foundation*. Ini berfungsi sebagai *platform* untuk menyajikan konten web, mulai dari halaman statis hingga aplikasi web dinamis. *Apache* adalah pilihan umum karena keandalannya, fleksibilitas, dan dukungan yang luas.

2.4.3 Composer

Composer adalah alat manajemen dependensi untuk bahasa pemrograman PHP. Ini memungkinkan pengembang untuk mendefinisikan dan mengelola dependensi yang diperlukan untuk proyek PHP mereka dalam file konfigurasi yang disebut "*composer.json*". Dengan menggunakan *Composer*, pengembang dapat dengan mudah mengelola pustaka, kerangka kerja, dan paket PHP lainnya yang dibutuhkan oleh proyek mereka. *Composer* juga digunakan untuk membuat sebuah proyek laravel dengan menggunakan *command line* yang telah disediakan.

2.4.4 Node.js

Node.js adalah lingkungan runtime JavaScript yang dibangun di atas mesin JavaScript V8 dari Chrome. Ini memungkinkan pengembang untuk menjalankan JavaScript di sisi server, yang sebelumnya terbatas hanya untuk eksekusi di browser. Dengan Node.js, pengembang dapat membuat aplikasi

web yang cepat dan berskala menggunakan JavaScript di kedua sisi, baik di sisi klien maupun server.

2.4.5 **Arduino IDE**

Arduino Integrated Development Environment (IDE) adalah platform perangkat lunak *open-source* yang digunakan untuk menulis, mengedit, dan mengunggah kode ke papan mikrokontroler *Arduino*. Platform ini sangat populer dalam bidang pendidikan dan penelitian karena kemudahan penggunaannya serta dukungan komunitas yang besar. *Arduino IDE* memfasilitasi pengembangan proyek elektronik dengan menyediakan lingkungan pemrograman yang intuitif. IDE ini mendukung bahasa pemrograman yang mirip dengan C/C++ dan menawarkan berbagai alat untuk membantu pengembangan dan *debugging* proyek. *Arduino IDE* banyak digunakan dalam pendidikan untuk mengajarkan dasar-dasar pemrograman dan elektronika. Dalam penelitian, platform ini sering digunakan untuk *prototyping* dan pengembangan cepat proyek *Internet of Things (IoT)*[22].

2.4.6 **ESP 32**

ESP32 adalah mikrokontroler berdaya rendah dengan *Wi-Fi* dan Bluetooth terintegrasi, yang dirancang untuk aplikasi *Internet of Things (IoT)*. Dikembangkan oleh *Espressif Systems*, ESP32 menawarkan kemampuan pemrosesan tinggi dan fitur konektivitas yang canggih, menjadikannya pilihan populer untuk proyek *IoT* dan *embedded systems*. ESP32 adalah *system-on-chip (SoC)* yang memiliki dua inti (*dual-core*) dengan kemampuan pemrosesan yang tinggi. Fitur utamanya termasuk *Wi-Fi*, *Bluetooth Low Energy (BLE)*, GPIOs, ADC, DAC, dan berbagai protokol komunikasi seperti SPI, I2C, dan UART. ESP32 banyak digunakan dalam pendidikan untuk mengajarkan konsep *IoT* dan *embedded systems*. Dalam penelitian, ESP32 digunakan untuk *prototyping* cepat dan pengembangan aplikasi *IoT* yang membutuhkan konektivitas nirkabel dan pemrosesan data[23].

2.4.7 MLX90640

MLX90640 adalah sensor gambar termal berbasis teknologi inframerah yang dikembangkan oleh Melexis. Sensor ini digunakan untuk mengukur distribusi suhu permukaan objek tanpa kontak fisik, dengan resolusi 32x24 piksel dan cakupan sudut pandang yang luas. Sensor ini banyak digunakan dalam aplikasi pemantauan suhu, keamanan, dan penginderaan jauh. MLX90640 menggunakan prinsip termografi inframerah untuk mendeteksi radiasi inframerah yang dipancarkan oleh objek. Sensor ini mengubah radiasi inframerah menjadi sinyal listrik yang kemudian diolah menjadi data suhu digital. MLX90640 digunakan dalam pendidikan untuk mengajarkan konsep dasar penginderaan termal dan dalam penelitian untuk mengembangkan aplikasi pemantauan suhu yang canggih, seperti deteksi panas tubuh dalam sistem keamanan dan *monitoring* suhu dalam sistem industri[24].

UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA