

BAB III

TINJAUAN PUSTAKA

3.1 Teori Pendukung

3.1.1 *Cloud Computing*

Menurut Purbo (2012:7), *Cloud Computing* adalah gabungan dari pemanfaatan komputer dengan layanan atau berbasis *internet* yang dapat di akses hanya melalui *internet* ada yang bersifat berbayar atau *free*. Sistem *Cloud* masih belum banyak diketahui orang, di era sekarang teknologi semakin maju dan orang ingin serba instan, terutama dalam mengakses *file* setiap hari kalau menggunakan penyimpanan bisa, ada kalanya hilang, lupa atau terinfeksi virus seperti di *flashdisk*.

Menurut Pratama (2014:60) Berdasarkan layanan *cloud computing* dibedakan menjadi tiga model yaitu:

1. *Insfrastructur AS a Service (IAAS)*

Insfrastructur AS a Service (IAAS) merupakan jenis layanan pada *cloud computing* yang menekankan kepada layanan penyedia sarana jaringan, komputer *server*, media penyimpanan (*storage*), *processor*, beserta dengan proses virtualisasi, yang menunjang proses komputasi.

2. *Platform AS A Service (PAAS)*

Platform AS A Service (PAAS) atau cloud *PAAS* merupakan jenis layanan pada *cloud computing* yang menekankan kepada penyediaan *Platform* untuk membantu proses pengembangan perangkat lunak secara cepat dan mudah. Layanan yang disediakan *Platform* yang disediakan oleh cloud *PAAS* umumnya juga berbasis *web*, dimana didalamnya telah tersedia banyak fitur yang memudahkan *programmers* dan pengguna awam didalam mengembangkan aplikasi tanpa memerlukan banyak proses penulisan sumber kode (*coding*).

3. *Software AS A Service (SAAS)*

Software AS A Service (SAAS) merupakan jenis layanan yang diberikan oleh teknologi *cloud computing* kepada para penggunanya dalam bentuk pemakaian bersama perangkat lunak (aplikasi). Umumnya layanan *SAAS* disediakan dalam bentuk tatap muka berbasis *web*. Bisa dikatakan *SAAS* merupakan jenis layanan *cloud computing* yang paling banyak digunakan dan paling mudah digunakan oleh para pengguna komputer, khususnya pengguna akhir yang tidak terlalu membutuhkan pengetahuan teknis didalam instalasi dan konfigurasi. Cukup dengan sebuah komputer atau perangkat *mobile*, sistem operasi, aplikasi *web browser*, dan koneksi *internet* atau *internet* saja,

seorang pengguna komputer dapat dengan mudah menggunakan layanan *cloud computing* tipe SAAS ini.

3.1.2 OwnCloud

Menurut Afrianto (2013:3), *Owncloud* merupakan salah satu perangkat lunak berbasis berkas gratis (lisensi *AGPL*) dan bebas disertai juga edisi bisnis dan edisi perusahaan, menyediakan pengamanan yang baik bagi pengguna aplikasi untuk membagi dan mengakses data yang secara terintegrasi dengan perangkat teknologi informasi yang tujuannya mengamankan, melacak dan melaporkan penggunaan data.

Dalam layanan *Cloud Computing* ada beberapa Komponen yang diperlukan, yaitu :

1. Cloud Clients

Seperangkat komputer atau *software* yang secara khusus didesain untuk penggunaan layanan berbasis *cloud computing*. Contohnya *Windows mobile, Symbian, Internet Explorer, Firefox, Chrome*.

2. Cloud Services

Produk, layanan dan solusi yang dipakai dan disampaikan secara *real-time* melalui media *internet*. Contohnya *web service* seperti *Amazon, Google maps, Yahoo! dan Paypal*.

3. *Cloud Applications*

Aplikasi-aplikasi yang memanfaatkan *cloud computing* dalam arsitektur *software* sehingga penggunaanya tidak perlu meng-*install* namun bisa menggunakan aplikasi dengan menggunakan komputer. Contohnya *Facebook, Google Apps*.

4. *Cloud Platform*

Layanan berupa platform komputasi yang terdiri dari *hardware* dan *software* infrastruktur. Contohnya *Phyton Django, Rubyon Rails, .Net*,

5. *Cloud Storage*

Melibatkan proses penyimpanan data sebagai sebuah layanan. Misalnya *Google Big Table, Amazon SimpelDB, Nirvanix CloudNas, MobileM iDisk*.

6. *Cloud Infrastruture*

Penyimpanan infrastruktur komputasi sebagai sebuah layanan. Contohnya *Sun Grid, GoGrid, Skytab*, dan *Amazon Elastic Computer Cloud*.

Adapun tujuan dari *cloud* adalah:

1. Mencegah *server down* pada diakses orang banyak
2. Menganalisa seberapa kuat *server* sanggup menampung banyak *user*

3. Menjaga *server* dalam kondisi stabil.

Selain itu manfaat dari *cloud* adalah :

1. Skalabilitas, melihat kemampuan *server* untuk menangani penambahan beban *user* yang banyak.
2. Keandalan, merupakan kemungkinan suatu sistem yang mampu melaksanakan fungsi tertentu pada kondisi tertentu tanpa mengalami kegagalan.

Ada beberapa jenis dari infrastruktur *cloud* yaitu :

1. **Public Cloud**

Menurut Andriani (2013:14), *Public Cloud* dapat diakses oleh beberapa pelanggan dengan koneksi *internet* dan mengakses ruang *cloud*. *Public cloud* juga dapat diartikan penggunaan *cloud* yang dipakai untuk masyarakat umum dan membayar saat selesai pemakaian atau biasa disebut dengan istilah *a pay as you go manner*.

2. **Private Cloud**

Private Cloud di dirikan hanya untuk grup atau organisasi dan akses ke dalam *cloud* terbatas pada grup atau organisasi tersebut. Layanan *cloud computing* untuk pusat data *internal* dari sebuah grup atau organisasi tersebut mendapatkan manfaat yang besar dari pemakaian *cloud computing* dimana masyarakat umum tidak mendapatkannya.

3. *Community Cloud*

Community Cloud dipakai dan dibagi diantara dua tau lebih organisasi yang mempunyai kepentingan yang sama pada penggunaan *cloud*. *Community cloud* bias dikelola oleh organisasi atau pihak ketiga dan bisa berada pada lokasi yang sama atau tidak.

4. *Hybrid Cloud*

Hybrid Cloud yang pada dasarnya merupakan kombinasi dari dua *cloud* dimana dapat dimasukkan campuran *public*, *private*, atau *community*. Tujuan dari *hybrid cloud* adalah ketika kehabisan kapasitas saat menggunakan *private cloud*, kita dapat langsung beralih ke *public cloud* atau *community cloud* untuk kapasitas tambahan.

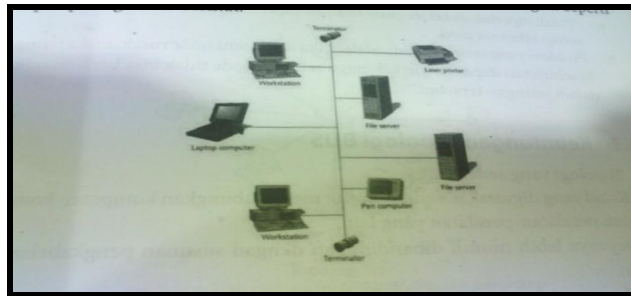
3.1.3 Topologi Jaringan

Menurut Winarno (2013:39) topologi adalah set aturan untuk mendeskripsikkan pengaturan komputer, *printer*, dan peranti lainnya yang terhubung melalui jaringan. Ada 5 jenis topologi berupa, Topologi *Bus*, Topologi *Star*, Topologi *Ring*, Topologi *Mesh*, dan Topologi *Tree*.

1. Topologi *Bus*

Menurut Badrul (2012: 38) Topologi *Bus* merupakan topologi yang banyak digunakan pada masa penggunaan kabel.

Dengan menggunakan *T-Connector* (dengan *terminator 50 ohm* pada ujung *network*), maka komputer atau perangkat jaringan lainnya bisa dengan mudah dihubungkan satu dengan yang lain.

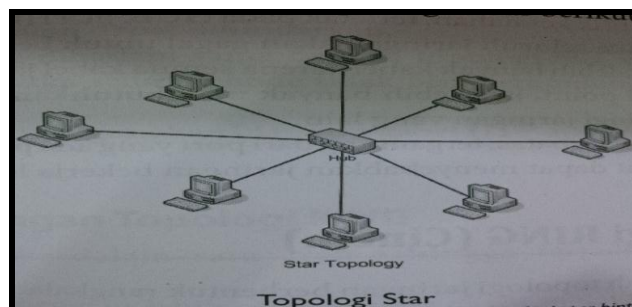


Sumber: Badrul (2012:38)

Gambar 3.1 Topologi Bus

2. Topologi Star

Menurut Badrul (2012:40) Topologi *Star* merupakan topologi jaringan yang paling sering digunakan. pada topologi *star*, kendali terpusat dan semua *link* harus melewati pusat yang menyalurkan data tersebut ke semua simpul atau komputer yang dipilihnya.

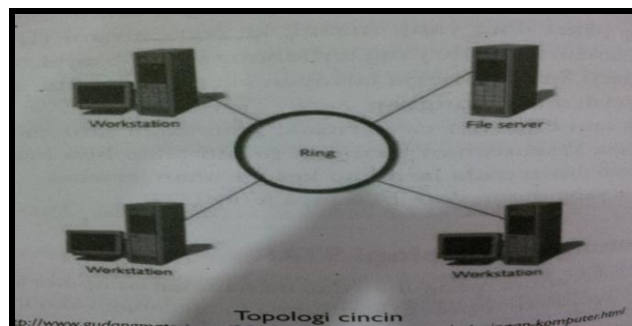


Sumber: Badrul (2012:40)

Gambar 3.2 Topologi Star

3. Topologi *Ring*

Menurut Badrul (2012:42) Topologi *Ring* adalah topologi jaringan berbentuk rangkaian titik yang masing-masing terhubung kedua titik lainnya, sehingga membentuk jalur melingkar membentuk cincin.

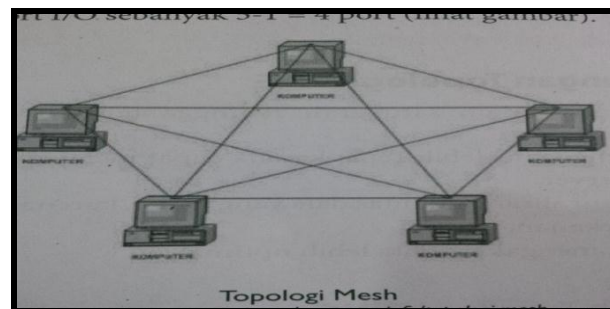


Sumber: Badrul (2012:42)

Gambar 3.3 Topologi *Ring*

4. Topologi *Mesh*

Menurut Badrul (2012:43) Topologi *Mesh* adalah suatu bentuk hubungan perangkat antar perangkat dimana setiap perangkat terhubung secara langsung ke perangkat lain yang ada di dalam jaringan.

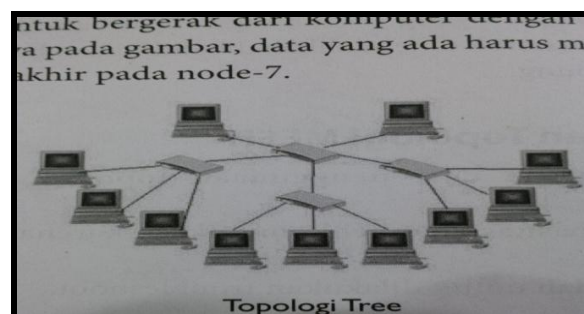


Sumber: Badrul (2012:43)

Gambar 3.4 Topologi Mesh

5. Topologi Tree

Menurut Badrul (2012:45) Topologi *Tree* adalah kombinasi karakteristik antara topologi *star* dan topologi *bus*. Topologi ini terdiri atas kumpulan topologi bintang yang dihubungkan dalam satu topologi *bus* sebagai tulang punggung atau *backbone*.



Sumber: Badrul (2012:45)

Gambar 3.5 Topologi Tree

3.2 Penelitian Terdahulu

No	Judul	Penulis/Tahun	Hasil
1	Analisa Perbandingan Kinerja <i>WebServer Apache</i> dan <i>Nginx</i> Menggunakan <i>Httpperf</i> Pada Portal Berita (Studi Kasus beritalinux.com)	Intan Ferina Irza, Zuhendra dan Efrifon (2017:76)	Hasil dari penelitian ini adalah mengetahui kinerja dari <i>webserver</i> untuk mengatasi <i>request</i> yang berlebihan dan membuat pemilik <i>webserver</i> tidak akan salah memilih layanan <i>server</i> yang tepat. Parameter yang diukur pada penelitian ini berupa <i>throughput</i> , <i>connection</i> , <i>request</i> , <i>replay</i> dan <i>error</i> . Dengan beban <i>connection</i> yang digunakan adalah 100, 500 dan 1000 dengan <i>rate/second</i> , dengan beban 10, 50 dan 100. Metode penelitian yang digunakan adalah kuantitatif dengan format deskriptif

No	Judul	Penulis/Tahun	Hasil
2	Perbandingan Perfoma <i>Reverse Proxy</i> <i>Caching Nginx</i> dan <i>Varnish</i> Pada <i>WebServer</i> <i>Apache</i>	Muhammad Luthfi, Mahendra Data dan Widhi Yahya (2018:1462)	Hasil dari penelitian ini adalah kedua <i>reverse proxy caching</i> yang diujikan memiliki <i>management</i> sumber daya perangkat keras <i>cpu</i> yang lebih baik dari pada <i>webserver apache</i> tanpa <i>reverse proxy caching</i> . Tetapi penggunaan <i>memory server apache</i> tanpa <i>reverse proxy caching</i> memiliki penggunaan <i>memory</i> yang lebih rendah. Parameter yang digunakan pada parameter ini adalah <i>full traffic, half traffic dan quarter traffic</i> , dengan beban 625, 1250, 2500 permenit. Metode penelitian yang digunakan adalah studi literature, Analisa kebutuhan, perancangan lingkungan pengujian, imlementasi, pengujian dan analisa.

No	Judul	Penulis/Tahun	Hasil
3	Studi Literatur :Teknologi <i>Cloud Computing</i>	Lola Yorita Astri (2016:840)	Hasil dari penelitian ini adalah untuk menyediakan pemahaman yang lebih baik mengenai <i>cloud computing</i> teknologi yang berkembang berkaitan dengan <i>cloud computing</i> , konsep dasar, arsitektur dan tantangan penelitian di bidang <i>cloud computing</i> . Metode penelitian yang digunakan adalah penelitian studi literature.
4	Pemanfaatan <i>Cloud Computing</i> Dalam Pengembangan Bisnis	Anik Andriani (2013:14)	Hasil dari penelitian ini adalah menganalisa pemanfaatan <i>cloud computing</i> sehingga diperoleh cara yang tepat sesuai kebutuhan metode studi literatur <i>cloud computing</i> dan pemanfaatan <i>cloud</i> . Metode penelitian yang digunakan adalah kuantitatif yang dilakukan dengan studi literature.