

BAB II DASAR TEORI

2.1 KAJIAN PUSTAKA

Penyimpanan data *remote* dalam kluster terdistribusi menjadi fokus perhatian yang sebagian besar solusi yang digunakan dalam konteks ini mengadopsi strategi penyimpanan *indeks* tabel besar (*big tabel index storage*). Namun, dalam praktiknya, kinerja strategi penyimpanan *indeks* tabel besar cenderung menurun ketika skenario menjadi lebih kompleks. Penyebab fenomena ini dianalisis dalam penelitian ini. Untuk meningkatkan kinerja baca dan tulis penyimpanan data jauh yang terdistribusi, penelitian ini mengusulkan strategi penyimpanan berbasis perangkat lunak ceph *cluster*. Strategi ini mengkapsulasi gambar penginderaan jarak jauh dalam bentuk objek melalui strategi pengelolaan metadata untuk mencapai pengambilan data berkelanjutan dalam ruang dan waktu, serta menemukan lokasi kluster data berkelanjutan melalui perhitungan serupa hash[5].

Dalam penelitian yang di lakukan yang merancang deduplikasi yang efisien algoritma berdasarkan arsitektur penyimpanan terdistribusi ceph *cluster*. *Algoritme* ini menggunakan deduplikasi data tingkat blok *online* teknologi untuk menyelesaikan pemotongan data, yang tidak mempengaruhi proses penyimpanan data di ceph atau mengubah antarmuka lain dan fungsi di ceph. Tanpa bergantung pada *node* pusat mana pun, itu algoritma mempertahankan karakteristik ceph dengan merancang objek hash khusus untuk menyimpan data sidik jari, dan menggunakan algoritma *CRUSH* untuk menilai duplikasi data berdasarkan perhitungan, bukan pencarian *global*. Algoritma mengganti data duplikat dengan objek yang dihapus duplikatnya, yang menyimpan sidik jari mereka dengan ruang penyimpanan lebih sedikit. Jurnal ini membandingkan efek dari ukuran blok yang berbeda sehubungan dengan kinerja dan tingkat deduplikasi melalui eksperimental studi, dan pilih ukuran blok yang paling sesuai di implementasi prototipe. Hasil percobaan menunjukkan hal itu algoritma ini tidak hanya dapat menghemat ruang penyimpanan secara efektif tetapi juga meningkatkan pemanfaatan *bandwidth* saat membaca dan menulis data duplikat[6].

Penelitian ini membahas tentang pentingnya penerapan teknik keamanan tingkat lanjut dan keandalan aplikasi dan layanan dalam menghadapi tantangan bisnis-teknis kontemporer. Salah satu solusi yang diusulkan dalam makalah ini adalah implementasi *cluster server* ceph dengan ketersediaan tinggi menggunakan *platform* virtualisasi proxmox. Teknik pemagaran berbasis IPMI juga diperkenalkan sebagai faktor peningkatan keandalan mesin *virtual* yang berjalan di dalam *cluster*. Penelitian tersebut menemukan bahwa solusi virtualisasi *open source*, *cluster server* yang didesain dengan baik, dan teknik pemagaran yang canggih dapat secara signifikan meningkatkan keandalan dan ketersediaan layanan dan aplikasi. Dalam konteks bisnis-teknis saat ini, penting untuk mengimplementasikan teknik keamanan tingkat lanjut dan mengoptimalkan sumber daya komputasi. Penelitian ini menunjukkan bahwa dengan menggunakan *cluster server* ceph yang dirancang dengan baik, *platform* virtualisasi proxmox, dan teknik pemagaran berbasis IPMI, perusahaan dapat meningkatkan keandalan dan ketersediaan layanan dan aplikasi mereka. Ini memberikan solusi yang efektif dalam mengatasi tantangan yang dihadapi dalam dunia bisnis teknologi saat ini[7].

Penelitian ini membahas pengembangan *platform crowdsourcing* untuk pelabelan citra penginderaan jauh. Untuk mengumpulkan dan membaca data secara efisien, penulis mengusulkan metode desain *grid* adaptif yang memungkinkan penginderaan jauh dipecah menjadi ubin-ubin yang dapat diproses secara paralel. Selain itu, klaster ceph digunakan untuk penyimpanan data. Untuk memastikan kualitas hasil, penulis mengusulkan mekanisme kontrol kualitas yang melibatkan ahli dan pengguna *platform*. Sebuah prototipe sistem pelabelan *online* juga telah dikembangkan, yang memungkinkan pengguna untuk memberi label pada citra tanpa menginstal perangkat lunak atau mengunduh citra besar secara langsung. Dengan adanya *platform* ini, dataset pelatihan yang besar dapat dikumpulkan untuk klasifikasi penginderaan jauh[8].

Kemudian ada beberapa penelitian tentang raid *cluster* yang menjadi dasar untuk penelitian ini, penelitian tersebut merupakan penelitian terbaru paper atau jurnal yang ada. Penelitian yang membahas tentang cara mengevaluasi raid yang didekluster sehubungan dengan kinerja pemulihan raid dibandingkan dengan ZFS

RAIDZ. Selain itu, jurnal secara formal menganalisis keandalan *deClustered* raid dalam hal *mean-time-to-data-loss* (MTTDL) dan menemukan bahwa kinerja pemulihan yang ditingkatkan mengarah ke keandalan penyimpanan yang lebih tinggi dibandingkan dengan raid tradisional[9].

Penelitian yang membahas untuk mengusulkan arsitektur raid baru, yang disebut FreeRAID, untuk memanfaatkan interaksi antara pengontrol RAID dan SSD untuk mengoptimalkan masa pakai *array* SSD berbasis *flash*. FreeRAID menambahkan fase baru yang dapat dieksploitasi pada siklus hidup blok *flash*. Pada FreeRAID, ruang *flash* dipisahkan menjadi ruang normal dan ruang yang dapat dieksploitasi, dan keduanya digunakan untuk menyajikan data normal dan data yang dapat dieksploitasi, masing-masing. Jurnal merancang skema manajemen ruang ganda untuk pengontrol RAID agar dapat mengalokasikan SSD secara cerdas ruang berdasarkan status umurnya. Di dalam SSD, jurnal mengusulkan sebuah FTL adaptif untuk pengontrol SSD untuk menjaga keandalan dan efisiensi ruang memori *flash*. Jurnal menerapkan *prototipe* FreeRAID berdasarkan simulator *array* SSD [10].

Penelitian ini mengusulkan skema penskalaan berdasarkan kode *N-Code* untuk RAID-6 dalam sistem penyimpanan SNC (*Storage Network Code*). Melalui data eksperimen yang nyata, dapat ditemukan bahwa skema SNC dapat secara signifikan mengurangi jumlah data migrasi yang diperlukan untuk memenuhi persyaratan migrasi data minimal. Dibandingkan dengan skema *Round-Robin*, skema ini mampu mengurangi jumlah migrasi data sebesar 76,52-88,94% dan mempersingkat total waktu migrasi data sebesar 54,5-62,4%. Selain itu, skema SNC menunjukkan keandalan yang lebih tinggi dalam menjaga integritas data selama proses migrasi. Penggunaan *N-Code* juga membantu dalam meningkatkan efisiensi penyimpanan dengan menyediakan redundansi yang lebih baik, sehingga mengurangi risiko kehilangan data. Dalam skenario dunia nyata, penerapan skema ini dapat menghasilkan penghematan biaya operasional dan peningkatan kinerja sistem secara keseluruhan. Dengan demikian, penelitian ini memberikan kontribusi yang signifikan dalam bidang manajemen data dan penyimpanan, menawarkan solusi yang lebih efektif dan efisien untuk penskalaan dan migrasi data pada sistem RAID-6.[11].

Perkembangan teknologi yang cepat menyebabkan kebutuhan penyimpanan data semakin berkembang. Salah satu untuk memperbesar kapasitas penyimpanannya dengan metode *Clustered file system*. Pada pengujian ini membandingkan kecepatan *upload* dan *download file* dan *write/read file* pada GlusterFS dan ceph. Pengujian *transfer file* menggunakan *file* dengan ukuran 500MB, 10 kali pengujian, dan menggunakan aplikasi teracopy. Dari pengujian maka diperoleh hasil untuk *upload file* bahwa metode GlusterFS lebih cepat 11,5% daripada ceph dengan rata-rata *upload file* GlusterFS lebih tinggi sebesar 3,57MB/s dan CephFS sebesar 3,20MB/s, hasil yang diperoleh untuk *download file* bahwa metode GlusterFS lebih cepat 11,3% daripada ceph dengan rata-rata *upload file* GlusterFS lebih tinggi 4,13MB/s dan CephFS sebesar 3,71MB/s, hasil yang diperoleh untuk *write file* bahwa metode GlusterFS lebih cepat 106% dari pada ceph dengan perbandingan sebesar 11,34kB/s dan 8,05kB/s, untuk *read file* bahwa metode GlusterFS lebih cepat 37% daripada ceph dengan perbandingan sebesar 3,10kB/s dan 2,25kB/s. Dari analisis tersebut bahwa metode GlusterFS lebih baik 100% dengan menggunakan 2 *node* yang masing-masing memiliki *virtual disk* yang dapat digabung dan mempercepat *performance*-nya, sedangkan ceph terbagi 3 *node* dimana 1 *node* digunakan sebagai MON yang berisikan penyimpanan metadata dan *pool* data yang memiliki proses lebih banyak sehingga mengakibatkan turunnya *performance* pada *file system* tersebut[12].

Pada penelitian ini membahas tentang perbandingan kinerja antara penyimpanan terdistribusi ceph dan konfigurasi raid dalam lingkungan proxmox. Untuk melaksanakan penelitian ini, pengujian akan dilakukan menggunakan aplikasi fio untuk mengukur sejumlah parameter kunci, termasuk latensi, kecepatan *transfer*, dan *bandwidth*. Penelitian bertujuan untuk melakukan analisis mendalam terkait kinerja *transfer* data antara solusi ceph dan raid. Selain itu, penelitian ini juga bertujuan untuk mengukur efektivitas implementasi *disaster recovery* pada kedua teknologi tersebut. Dengan demikian, penelitian ini diharapkan akan memberikan wawasan yang berharga dalam pemahaman tentang keunggulan dan kelemahan masing-masing solusi penyimpanan dan bagaimana mereka dapat berkontribusi pada keberlanjutan dan keandalan sistem dalam lingkungan proxmox

Tabel 2. 1 Kajian Penelitian Sebelumnya

<i>Year</i>	<i>Author</i>	<i>Objective</i>	<i>Testing Methodology</i>	<i>Tools</i>	<i>Result</i>
2022	Erick Wijaya	Penelitian membahas perbandingan antara GlusterFS dengan CephFS	<i>Storage Cluster</i> CephFS dengan GlusterFs	KVM, GlusterFs dan cephFS	GlusterFs lebih cepat di bandingkan cephFS
2021	Xinyu Tang	Penelitian ini membahas perbandingan kinerja strategi penyimpanan <i>indeks</i> dan <i>ceph cluster</i>	<i>Storage cluster</i> ceph dengan <i>Library</i> GDAL	<i>Ceph cluster</i> dan <i>Library</i> GDAL	Penggunaan ceph membuktikan kelayakan, efektivitas, dan stabilitas strategi.
2019	Jinpen Wang	Merancang deduplikasi pada algoritma <i>ceph cluster</i>	Algoritma deduplikasi <i>ceph cluster</i>	<i>Ceph cluster</i>	Efektif dapat menghemat ruangan penyimpanan
2019	Aleksandar Bajić	Implementasi <i>ceph cluster server</i> dengan <i>platform virtualisasi</i> .	Algoritma IPMI (<i>The Intelligent Platform Management Interface</i>)	<i>Ceph Cluster</i> dan Virtualisasi	Meningkatkan keandalan dan Layanan <i>cluster server</i> .
2021	Zhaoyan Shen	Merancang skema manajemen ruang ganda untuk mengontrol raid 6	Merancang FTL adaptif untuk SSD	FreeRAID, SSD	Masa pakai SSD meningkan hingga 3,07x

<i>Year</i>	<i>Author</i>	<i>Objective</i>	<i>Testing Methodology</i>	<i>Tools</i>	<i>Result</i>
2020	Yu Hu	Merancang skema penskalaan untuk Raid 6	Menguji waktu migrasi di raid 6	Arsitektur <i>round-robin</i>	Mempersikat total waktu migrasi data 54%
2019	Zhi Qiao	Membanding raid dengan ZFS RAIDZ	Menguji keandalan <i>deClustered</i> RAID	Arsitektur MTTDL	Menunjukkan kinerja tinggi di bandingkan raid konvensional.
2024	Akbar Usamah	Membandingkan Kinerja pendistribusian data ceph <i>Cluster</i> dan raid	Menguji Perbandingan kecepatan <i>transfer</i> antara ceph <i>cluster</i> dan RAID	Ceph <i>cluster</i> dan raid.	

2.2 DASAR TEORI

2.2.1 *Cloud Storage*

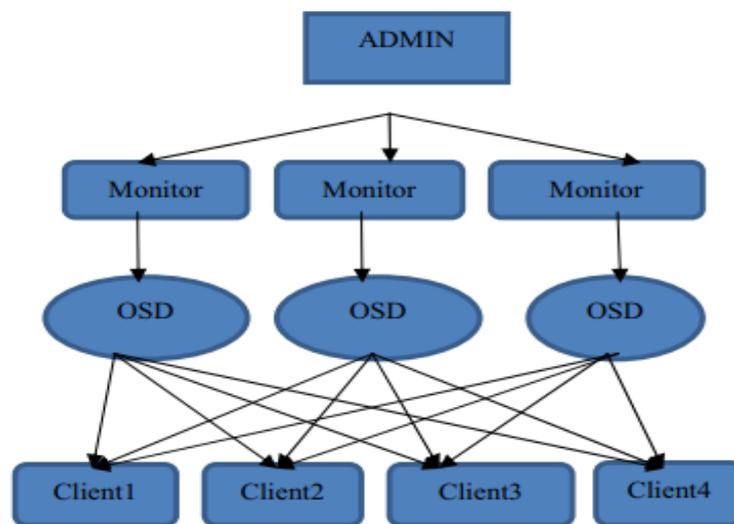
Cloud Storage adalah sebuah teknologi penyimpanan data digital yang memanfaatkan adanya *server virtual* sebagai media penyimpanan[13]. *Cloud computing* adalah sebuah model komputasi, dimana sumber daya seperti prosesor, *storage*, *network*, dan *software* menjadi abstrak dan diberikan sebagai layanan di jaringan atau internet menggunakan pola akses remote. Model billing dari layanan ini umumnya mirip dengan model layanan publik. Ketersediaan *on-demand* sesuai kebutuhan, mudah untuk dikontrol, dinamik dan skalabilitas yang hampir tanpa batas adalah beberapa atribut penting dari *cloud computing*. Sebuah setup infrastruktur model *cloud computing* biasanya di kenali sebagai *cloud*[14].

2.2.2 *Ceph Cluster*

Ceph cluster adalah *software defined storage* yang bersifat *open-source* banyak digunakan untuk membangun sistem penyimpanan data yang andal. Di sistem penyimpanan data berbasis ceph koneksi jaringan diperlukan untuk mengintegrasikan perangkat penyimpanan yang membangun *cluster ceph*[15]. Arsitektur ceph mengasumsikan bahwa sistem pada skala *petabyte* pada dasarnya dinamis: sistem besar pasti dibangun secara bertahap, kegagalan *node* adalah hal yang biasa dan bukan pengecualian, dan kualitas serta karakter beban kerja terus berubah seiring waktu. ceph memisahkan operasi data dan metadata dengan menghilangkan tabel alokasi *file* dan menggantinya dengan fungsi pembangkit. Hal ini memungkinkan ceph memanfaatkan kecerdasan yang ada di osd untuk mendistribusikan kompleksitas seputar akses data, memperbarui serialisasi, replikasi dan keandalan, deteksi kegagalan, dan pemulihan[16].

Ceph cluster menyimpan data ke dalam blok-blok data menggunakan RBD (*RADOS Workload Device*). Selain mendukung redundansi data, dengan konfigurasi yang tepat ceph dapat meningkatkan kinerja sistem penyimpan *back-end tervirtualisasi*. ceph ini telah diimplementasikan untuk membantu pengelolaan dan pertukaran data menggunakan *Software Defines Networking* (SDN) yang tidak memerlukan perangkat khusus pada sistem penyimpanan, salah satunya adalah OSIRIS. Sistem mengecek keadaan *harddisk* yang terdapat pada *pool*. Apabila terjadi kerusakan pada *harddisk*, sistem masih dapat berjalan, akan

tetapi hanya dengan sebuah *harddisk*. Hal ini dikarenakan sistem yang bersifat *redundant*. *Harddisk* yang tidak dilakukan penggantian maka dapat mengakibatkan kegagalan sistem. Apabila tidak terjadi kerusakan, maka sistem berjalan normal. Apabila dilakukan pergantian *harddisk* yang rusak, maka sistem diharapkan melakukan replikasi data. Replikasi data yang dilakukan yaitu antara sebuah *harddisk* normal dengan *harddisk* baru yang menggantikan *harddisk* yang rusak[13].



Gambar 2. 1 Ceph Cluster Architecture[4].

Pada gambar 2.1 *Cluster* penyimpanan ceph terdiri dari beberapa *daemon* perangkat lunak yang berbeda. Masing-masing *daemon* ini menangani fungsionalitas ceph *cluster* yang unik dan menambahkan nilai pada komponen terkait, komponen tersebut antara lain :

1. Rados (*Reliable Autonomic Distributed Object Store*)

Adalah dasar dari kluster penyimpanan ceph. Segala sesuatu di ceph disimpan dalam bentuk objek, dan penyimpanan objek RADOS bertanggung jawab untuk menyimpan objek ini, terlepas dari tipe datanya[6].

2. Ceph *Daemons*

Data disimpan di ceph *object storage device* (OSD) dalam bentuk objek. Ini adalah satu-satunya komponen *cluster* ceph tempat data pengguna

sebenarnya disimpan dan data yang sama diambil ketika klien mengeluarkan operasi baca[6].

3. *Monitor Ceph* (MON)

Melacak kesehatan seluruh klaster dengan menyimpan peta status klaster, yang mencakup peta OSD, MON, PG, dan CRUSH. Semua *node cluster* melapor untuk memantau *node* dan berbagi informasi tentang setiap perubahan statusnya. Monitor memelihara peta informasi terpisah untuk setiap komponen[6].

4. *Rados workload Device* (RBD)

Menyediakan penyimpanan blok, yang dapat dipetakan, diformat, dan dipasang seperti *disk* lainnya ke *server*. Perangkat blok ceph dilengkapi dengan fitur penyimpanan perusahaan seperti penyediaan tipis dan *snapshot*[6].

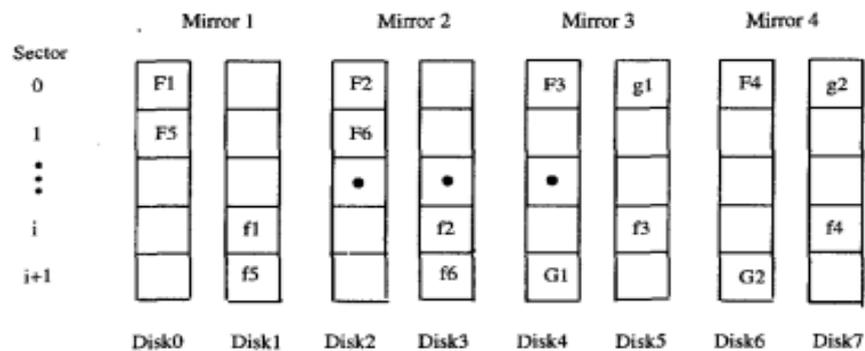
2.2.3 RAID (*Redundant Array of Independent Disk*)

Sistem penyimpanan *redundant arrays of cheap disks* (RAID) dikembangkan sekitar tahun 1988 untuk meningkatkan keandalan sistem penyimpanan. raid dibuat dengan adanya permasalahan pada kegagalan data pada *hard disk drive* (HDD). sistem raid mengikuti proses *Poisson* yang *homogen*[18]. Ide dasar dibalik raid adalah untuk menghubungkan beberapa *array disk* fisik independen, atau kelompok, bersama-sama untuk menyediakan kapasitas penyimpanan yang besar, akses yang lebih cepat untuk membaca data, redundansi data dan, pada gilirannya, tingkat kinerja tinggi dan/atau toleransi kesalahan yang lebih besar. Ada banyak tingkatan sistem raid yang berbeda, tingkatan ini memberikan peringkat redundansi, pengecekan kesalahan, kapasitas, dan biaya yang berbeda semua level ini berputar membagi ruang penyimpanan yang tersedia menjadi volume yang dihasilkan komputer dan menyimpannya dalam *volume* secara distributif di antara beberapa *node*[19].

RAID dibagi ke dalam beberapa skema, yang disebut dengan "raid Level". Pada awalnya, ada lima buah raid level yang pertama kali dikonsepsikan, tetapi seiring dengan waktu, level-level tersebut berevolusi, yakni dengan

menggabungkan beberapa level yang berbeda dan juga mengimplementasikan beberapa level *proprietary* yang tidak menjadi standar raid[20].

Pada gambar 2.2 Teknik raid umumnya diimplementasikan dengan menggunakan perangkat keras raid *controller* yang mendukung raid 0, raid 1, raid 5, raid 6 dan beberapa *hybrid* raid. Teknik raid 2, raid 3 dan raid 4 sudah jarang digunakan karena dari sisi kinerja dan manfaat sudah tergantikan oleh raid 5. Sementara implementasi raid dalam bentuk perangkat lunak dapat ditemui berupa bagian dari sistem operasi. Dibandingkan dengan perangkat keras raid, penggunaan perangkat lunak raid lebih membebani sistem operasi dan mempengaruhi kinerja sistem[21].



Gambar 2. 2 Raid Architecture[22].

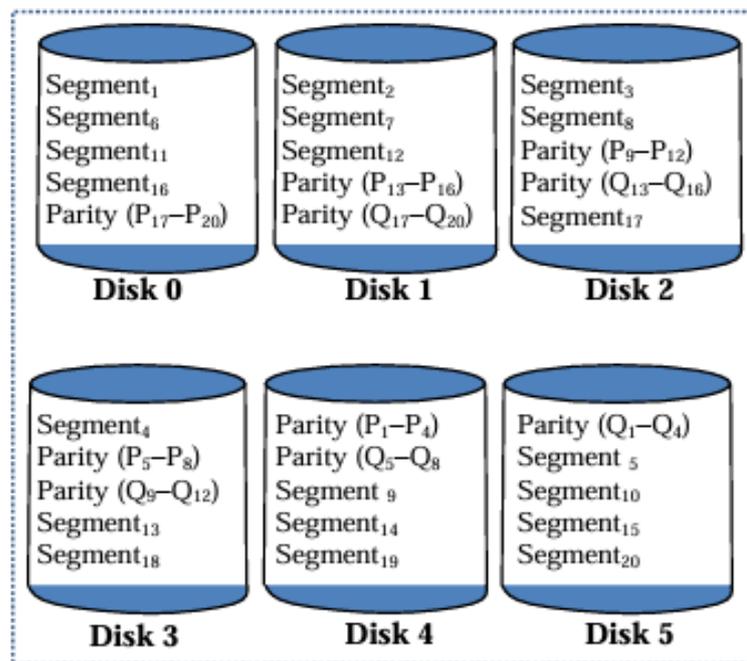
2.2.3.1 Raid Level-6 (*Striping* dengan paritas ganda)

RAID Level-6 mirip dengan raid 5 dengan menggunakan *striping disk* dan distribusi paritas ke semua *drive*, namun berbeda dengan menggunakan dua blok paritas independen per *stripe* sebagai alternatif dari satu[19]. Karena menggunakan paritas ganda, Level ini bertindak untuk menawarkan lebih banyak integritas data dengan menggunakan skema paritas ganda. Ditambah dengan sebelumnya keuntungan, masih banyak manfaat lain yang dirasakan yang memberikan momentum untuk level ini [19]:

1. Redundansi data dan paritas ganda menghasilkan perlindungan data tingkat tinggi. Pada gilirannya, level ini digunakan untuk data sensitif yang

memerlukan tingkat perlindungan yang sangat tinggi terhadap kehilangan demi kehilangan[19].

2. Kecepatan baca yang tinggi: ini karena data tersebar di lebih dari satu *disk drive*[19].
3. Kinerja yang baik: terutama untuk lingkungan yang menerapkan banyak transaksi *input/output* kecil. Namun demikian, waktu penulisannya lebih lama dibandingkan raid level-5. Hal ini disebabkan banyaknya permintaan tulis yang dapat terjadi secara bersamaan pada *disk drive* yang sama, bisa lihat pada gambar 2.3[15].



Gambar 2. 3 Pola Workload dari RAID level-6[15].

2.2.4 Virtualisasi

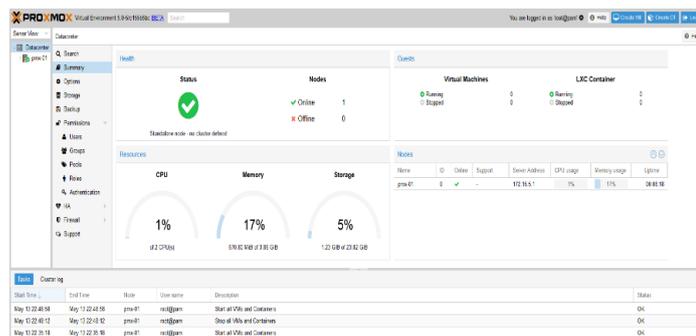
Virtualisasi atau *virtualization* adalah sebuah teknik atau cara untuk membuat sesuatu dalam bentuk virtualisasi, tidak seperti kenyataan yang ada. Virtualisasi juga digunakan untuk mengemulasikan perangkat fisik komputer, dengan cara membuatnya seolah-olah perangkat tersebut tidak ada (disembunyikan) atau bahkan menciptakan perangkat yang tidak ada menjadi ada, Teknologi virtualisasi umum digunakan di pusat data, *cloud computing*, dan lingkungan *desktop* untuk

meningkatkan efisiensi, fleksibilitas, dan ketersediaan sumber daya komputasi.[23].

2.2.5 Proxmox

Proxmox merupakan *software open-source virtualization platform* untuk menjalankan *virtual appliance* dan *virtual Machine*. Proxmox VE (*Virtual Environment*) adalah distro khusus yang didekasikan secara khusus sebagai mesin virtualisasi yaitu KVM dan *OpenVZ*[24]. *OS level virtualization* merupakan suatu bentuk teknologi virtualisasi yang memungkinkan kernel sistem operasi pada mesin *host*/inang mengeksekusi *user space* yang berbeda melalui suatu bentuk isolasi sumber daya. Kernel pada mesin inang mengkspos seluruh struktur *hardware* yang dimilikinya pada *guest* yang terisolasi, tetapi terbatas pada jumlah sumber daya, seperti ukuran RAM, jumlah inti *processor*, dan ukuran ruang *disk*[25].

Proxmox berbasis linux dengan arsitektur *x86_64*, oleh karena itu proxmox hanya dapat di-install pada mesin yang mendukung arsitektur *x86_64* atau berbasis 64 bit. Selain itu, Proxmox juga membutuhkan prosesor yang mendukung virtualisasi pada perangkat keras (*hardware virtualization*), baik prosesor Intel dengan teknologi Intel-VT nya maupun prosesor AMD dengan teknologi AMD-V nya, bisa lihat gambar 2.4[26].



Gambar 2. 4 Proxmox Dashboard[26].

2.2.6 IPv4

IPv4 adalah sebuah jenis pengalamatan jaringan yang digunakan di dalam protokol jaringan TCP/IP yang menggunakan protokol IP versi 4. Panjangnya adalah 32-bit, dan secara teoritis dapat mengalami hingga 4 miliar *host* komputer di seluruh dunia, Singkatnya, IPv4 adalah protokol internet

yang dominan digunakan untuk komunikasi data di internet, namun keterbatasannya telah memicu pengembangan dan adopsi bertahap protokol IPv6 yang lebih baru.[27].

2.2.7 SSH

SSH (*secure shell hosting*) adalah protokol atau aplikasi yang memungkinkan pertukaran data antara dua perangkat jaringan yang lebih aman dibandingkan dengan telnet, rsh dan rlogin. SSH banyak digunakan pada sistem berbasis linux dan unix untuk mengakses akun shell. SSH pertama kali dikembangkan oleh *openBSD project* dan kemudian versi rilis p (*port*) di *manage* oleh team *porting* ke sistem operasi lainnya, termasuk linux. Dengan SSH semua percakapan antara *server* dan klien dienkripsi, artinya apabila percakapan tersebut disadap, penyadap tidak akan memahami isinya[28].

2.2.8 FIO

Fio adalah alat yang banyak digunakan dan sangat fleksibel untuk melakukan *benchmarking* I/O. Ini mendukung beberapa sistem operasi dan mesin I/O. Ia bekerja pada perangkat blok, serta *file*. *Deutsche Telekom* menambahkan dukungan untuk mesin I/O baru ke fio, aplikasi FIO memiliki berbagai kegunaan dalam dunia penyimpanan data. Pertama, FIO dapat digunakan untuk menilai kinerja perangkat penyimpanan baru sebelum diimplementasikan dalam sistem produksi. Ini memungkinkan administrator sistem untuk memastikan bahwa perangkat penyimpanan baru memenuhi persyaratan kinerja yang diperlukan. Kedua, FIO dapat digunakan untuk membandingkan kinerja perangkat penyimpanan yang berbeda, seperti *hard drive* dan SSD. Ini membantu dalam memilih perangkat penyimpanan yang paling optimal untuk kebutuhan tertentu. Terakhir, FIO dapat digunakan untuk menganalisis kinerja sistem penyimpanan yang ada. Dengan menjalankan pengujian kinerja, *administrator* sistem dapat mengidentifikasi masalah kinerja dan mengoptimalkan konfigurasi sistem untuk meningkatkan efisiensi dan kecepatan.[29].

2.2.9 Ubuntu 20.04

Ubuntu linux adalah sistem operasi gratis dan sumber terbuka berbasis kernel linux. Sistem operasi ini dikembangkan oleh Canonical Ltd., sebuah perusahaan perangkat lunak yang berbasis di Inggris, dan merupakan salah satu

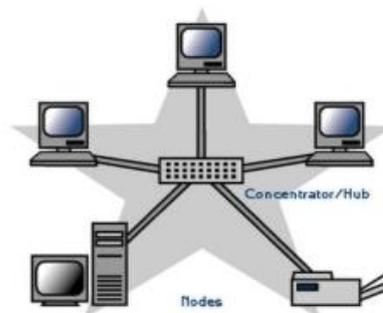
distribusi linux yang paling populer tersedia, Ubuntu 20.04 adalah versi *Long-Term Support* (LTS) dari distribusi Linux Ubuntu yang dirilis oleh Canonical. Berikut adalah beberapa poin penting tentang Ubuntu 20.04[31].

2.2.10 Hardisk

Hard Disk Drive (HDD) atau biasa dikenal dengan *harddisk* saja adalah sebuah media penyimpanan sekunder pada sebuah komputer. Meskipun disebut sebagai media penyimpanan sekunder namun pada kenyataannya fungsinya adalah sangat penting bahkan tidak bisa ditinggalkan lagi untuk kebutuhan sebuah komputer. Hal tersebut sangat jelas mengingat kebutuhan akan *software* berupa program maupun aplikasinya, serta data yang diolah membutuhkan media penyimpanan yang sangat besar, yang tidak cukup hanya ditampung oleh sebuah media penyimpanan utama berupa ROM (*Read Only Memory*) dan RAM (*Random Access Memory*)[32].

2.2.11 Topologi Star

Topologi *star* merupakan topologi yang banyak digunakan diberbagai tempat, karena kemudahan untuk menambah, mengurangi, atau mendeteksi kerusakan jaringan yang ada. Karakteristik topologi ini yaitu setiap *node* berkomunikasi langsung dengan *central node*, *trafficdata* mengalir dari *node* ke *central node* dan kembali lagi, mudah dikembangkan karena setiap *node* hanya memiliki kabel yang langsung terhubung ke *central node*, keunggulan jika satu kabel *node* terputus maka yang lainnya tidak akan terganggu untuk lebih jelas bisa lihat gambar 2.5 [33].



Gambar 2. 5 Topologi Start[33].

2.2.12 IOPS

IOPS (*input/output operations per second*) adalah metrik yang digunakan untuk mengukur performa input/output sistem penyimpanan, khususnya perangkat penyimpanan seperti *hard disk drive* (HDD) dan *solid-state drive* (SSD). Iops menunjukkan jumlah operasi *input/output* yang dapat dilakukan dalam satu detik. Secara spesifik, iops mengacu pada jumlah operasi *input/output* (baca atau tulis) yang dapat dilakukan oleh perangkat penyimpanan dalam satu detik. Semakin tinggi nilai iops, semakin cepat kemampuan perangkat penyimpanan tersebut dalam memproses operasi baca/tulis data. [34].

2.2.13 KiB/s

Pada bulan desember 1998, komisi elektroteknik internasional (IEC) menyetujui standar IEC baru yang menciptakan awalan khusus untuk kelipatan biner yang hanya terdiri dari dua huruf pertama dari metrik awalan dan menambahkan dua huruf pertama dari kata "biner". Contohnya, Kilobyte (KB) atau Gigabyte (GB), istilah yang baru adalah kibibyte (KiB) atau gibibyte (GiB), KiB/s (*Kibibytes per Second*) adalah sebuah unit pengukuran laju *transfer* data atau *throughput* yang biasa digunakan dalam dunia komputer dan teknologi informasi[35].