



# Pusat Data untuk Pemerintahan

Mardhani Riasetiawan

SE Ak, MT

Lab Sistem Komputer dan Jaringan  
Departemen Ilmu Komputer dan Elektronika, F MIPA UGM @ 2016

# Daftar Isi

<b>DAFTAR ISI</b>	<b>2</b>
<b>BAB 1. PUSAT DATA</b>	<b>4</b>
1.1 DEFINISI PUSAT DATA	5
1.2 LAYANAN PUSAT DATA	6
1.3 PARADOX BIAYA DAN REVENUE	7
1.4 PENGGUNA PUSAT DATA	8
1.4.1 LAYANAN INTERNET KOMERSIAL	8
1.4.2 E-GOVERNMENT	8
1.4.3 PENDIDIKAN	9
1.4.5 LEMBAGA RISET	10
<b>BAB 3. PENGEMBANGAN PUSAT DATA</b>	<b>11</b>
3.1 JARINGAN DATA NASIONAL	11
3.2 SATELIT	11
3.3 KABEL	12
3.4 SERVER FARM	12
3.5 LISTRIK	13
3.6 ARSITEKTUR	13
3.7 DISASTER RECOVERY CENTER	14
SPARE PARTS	15
PENGAMANAN	15
FISIK	15
3.8 CYBER SECURITY	17
3.9 MANAJEMEN	19
<b>BAB 4. PENGEMBANGAN DATA CENTER</b>	<b>21</b>
4.1 WEB SERVER MANAGEMENT	22
4.2 DATABASE SERVER MANAJEMEN	25
4.3 FILE SERVER MANAJEMEN	27
4.4 IMPLEMENTASI SECURITY	32
<b>BAB 5. VIRTUALISASI</b>	<b>34</b>
5.1 VIRTUALISASI SERVER	34
5.2 PENGGUNAAN VIRTUALISASI	35
5.2 TEKNOLOGI VIRTUALISASI SERVER	40
FULL VIRTUALIZATION	41
PARA-VIRTUALIZATION	41
HARDWARE-ASSISTED VIRTUALIZATION	42
5.3 PROXMOX VE	43
5.3.1 PERBANDINGAN PERANGKAT LUNAK VIRTUALISASI	46
5.3.2 PROXMOX VE CLUSTER	47
5.3.3 DATA STORAGE	49
<b>BAB 6. VIRTUAL DATA CENTER</b>	<b>51</b>

<b>6.1 IMPLEMENTASI VIRTUAL DATA CENTER</b>	<b>51</b>
MENJALANKAN PROXMOX	51
TAMPILAN HOST POXMOX	52
VIRTUAL MACHINE	52
<b>BAB 7. BIG DATA</b>	<b>54</b>

---

## Bab 1. Pusat Data

---



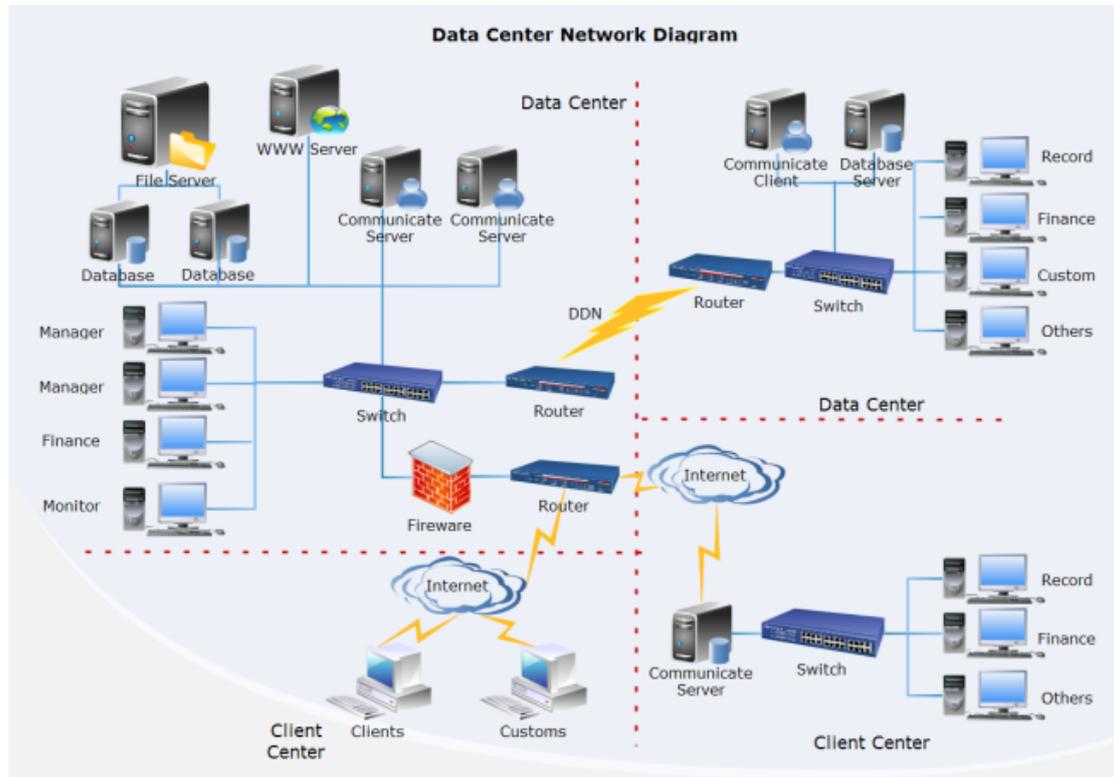
Pusat Data merupakan fasilitas yang memiliki kemampuan untuk mengatur, mengelola dan menyelenggarakan layanan teknologi informasi dan komunikasi dalam bentuk layanan. Pusat Data memiliki infrastruktur, konektivitas, pengelolaan, manajemen dan

alokasi sumber daya untuk keperluan ketersediaan layanan jangka panjang, kehandalan dan keamanan aset teknologi informasi dan komunikasi. Pusat Data lazimnya adalah terpusatnya fasilitas berupa sekumpulan komputer server, jaringan, penyimpanan data, prosedur keamanan dan pengelolaan yang didasarkan pada standar tertentu. Pusat Data yang menjadi sentral layanan teknologi informasi dan komunikasi menjadi aset vital bagi terselenggaranya layanan, baik berjalannya sebuah layanan sistem informasi, dan akses dari pihak-pihak yang berkepentingan. Pusat Data menjadi strategis karena menunjukkan eksistensi aset yang sangat berharga yaitu data dan informasi.

Dalam perspektif Nasional, sebuah negara, instansi pemerintah, organisasi bisnis untuk menjaga kepentingan, dan aset pentingnya keberadaan Pusat Data menjadi penting. Terselenggaranya layanan sistem informasi, akses informasi publik sampai keperluan tata kelola pemerintahan akan melalui, disimpan dan dikelola pada Pusat Data. Nilai aset data dan informasi yang dikelola Pusat Data berorientasi jangka panjang baik secara penyimpanan dan penggunaannya di masa mendatang. Pusat Data memungkinkan juga pengelolaan aset data dan informasi menjadi terkordinasi dengan baik, mewujudkan terhubungkan data dan informasi dari berbagai sumber dan penjagaan keamanan bersama yang kuat. Dalam kacamata negara, Pusat Data adalah aset yang harus dilindungi dan dikelola selayaknya aset-aset penting/vital negara yang lain. Negara dapat hadir dalam pengelolaan aset ini dengan menyediakan Pusat Data Nasional yang dapat digunakan untuk kepentingan Pemerintahan, Militer, dan Publik. Pengelolaan dapat dilakukasa secara terbuka, terbatas dan tertutup tergantung dari subyek/objek Pusat Datanya. Dengan bentang luas negara kesatuan Republik Indonesia kebutuhan Pusat Data menjadi penting

untuk mengintegrasikan Indonesia dalam kaca mata teknologi informasi dan komunikasi.

## 1.1 Definisi Pusat Data



Kumpulan server dan system penyimpanan data membutuhkan fasilitas untuk menampung semua sumber daya yang dimiliki. Fasilitas tersebut harus memenuhi kondisi server yang memungkinkan untuk melakukan pengaturan sumber daya, pengaturan udara serta memiliki system pengamanan fisik. Fasilitas yang menjadi pusat penampungan data (*Data center*) ini memiliki beberapa kriteria khusus dalam perancangannya, antara lain :

1. *Availability*, *Data center* mampu menjalankan operasi secara berkelanjutan dan terus menerus dalam kondisi apapun.
2. *Scalability* dan *Flexibility*, *Data Center* mampu beradaptasi dengan penambahan kebutuhan atau teknologi baru tanpa merubah substansi *data center* secara keseluruhan.
3. *Security*, *Data Center* mampu melindungi asset data yang tersimpan pada server secara fisik maupun non-fisik.

## 1.2 Layanan Pusat Data

*Data Center* menyediakan beberapa layanan untuk mendukung fungsinya sebagai fasilitas yang menampung semua transaksi yang terjadi pada server. Secara umum layanan utama yang disediakan oleh *data center* antara lain :

### ***Business Continuity Infrastructure***

Data center memiliki aspek-aspek yang mendukung kelangsungan bisnis ketika terjadi kondisi kritis. Aspek tersebut meliputi kriteria pemilihan lokasi data center, Kuantitas ruang data center, Layout ruangan dan Instalasi data center, kebutuhan system elektrik serta jaringan infrastruktur yang *scalable*.

### ***Data Center Security Infrastruktur***

Data center memiliki pengamanan fisik dan non fisik untuk menjaga asset data yang dimilikinya. Fitur *security system* yang harus dimiliki *Data Center* meliputi akses keluar masuknya user ke ruangan *Data Center*, serta manajemen keamanan pada system yang berjalan pada server.

### ***Application Optimization***

*Data center* memiliki beberapa layer dalam menjalankan aplikasi yang berjalan pada server. Konfigurasi di tiap *layer* dapat meningkatkan waktu respon suatu server. Layer 4(*end to end flow control, error detection & correction*) dan Layer 5(*Token Management, Data synchronization*) adalah *layer* yang menentukan optimasi pada aplikasi yang berjalan di server.

### ***IP Infrastructure***

*Data Center* memiliki Infrastruktur IP sebagai service utamanya. Service ini terdapat pada layer 2( *Service Device, acces probability, loop-free, predictable, scalable*) dan layer 3 (Dukungan terhadap default gateway). Pada layanan ini mengatur mengenai *Quality of service (QoS)*, *multicast*, *Private LAN* dan *policy-based*

### ***Storage***

Data center memiliki fitur utama sebagai infrastruktur penyimpanan. Layanan ini membahas mengenai arsitektur SAN, *fibre channel switching*, *replication*, *backup* serta *archival*.

### 1.3 Paradox Biaya dan Revenue

Pusat Data yang identik dengan fasilitas fisik komputer, server, penyimpanan, koneksi dan perangkat lainnya memang dibangun dengan dukungan kapasitas energi, dan lainnya yang tidak sedikit. Pusat Data juga harus ditempatkan pada lokasi yang secara fisik lahan aman dari gangguan baik bencana alam, masalah sosial dan berada relatif dekat dengan pusat bisnis. Pusat Data memerlukan pengelolaan 24 jam selama 7 hari dan 364 hari dalam setahun, yang mengharuskan ada pola penyediaan layanan terus menerus. Hal ini menyebabkan Pusat Data identik dengan investasi yang cukup besar. Investasi besar ini biasanya muncul pada fase awal penyediaan Pusat Data.

Selayaknya investasi, maka Pusat Data diharapkan nantinya akan menjadi business enabler yang menghasilkan revenue, dalam bentuk uang ataupun value added yang lainnya. Investasi yang baik dan kemudian dikelola dengan sangat baik untuk orientasi nilai, maka dengan sendirinya nilai investasi akan naik dimasa mendatang, dan menghasilkan revenue yang tidak sedikit. Bagi negara investasi Pusat Data selayaknya membangun infrastruktur jalan untuk menumbuhkan aktivitas-aktivitas produktif yang menghasilkan nilai ekonomis yang tidak hanya saat itu tapi terus menerus dimasa depan.

Pusat Data dipandang sebagai aset yang memerlukan pengelolaan jangka panjang. Nilai investasi akan naik seiring dengan pola pengelolaannya yang semakin baik. Pusat Data yang dapat menampung cukup banyak aset data dan informasi juga akan menaikkan investasi Pusat Data. Nilai investasi yang semakin membesar akan menjadi daya tarik bagi peminat layanan untuk menggunakan secara maksimal untuk menghasilkan nilai ekonomis lainnya.

## **1.4 Pengguna Pusat Data**

Target pengguna Pusat Data Nasional dengan karakteristik kehidupan berbangsa dan bernegara Indonesia sangat potensial beragam. Bentang luas negara kepulauan Indonesia sudah menunjukkan potensi target pengguna akan sangat variatif bagi dari sisi keperluan, dan kemanfaatannya.

### **1.4.1 Layanan Internet Komersial**

Penduduk Indonesia yang sudah mencapai angka 240 juta lebih, dengan jumlah konektivitas lebih dari 70 juta, maka dengan sendirinya penduduk digital Indonesia akan menjadi pengguna layanan Pusat Data Nasional ini. Pengguna ini akan menggunakan jasa provider koneksi internet baik melalui jalur perusahaan-perusahaan telekomunikasi, jalur koneksi dedicated dan lainnya ke pusat-pusat sumber data dan informasi. Selain akan memfasilitasi kebutuhan konektivitas internet komersial, Pusat Data akan merangsang berkembangnya jalur-jalur koneksi internet komersial, dan yang dedicated.

Layanan internet komersial yang dibutuhkan akan terbentang dari ujung-ujung Indonesia, dikumpulkan pada simpul-simpul koneksi, dan mengarah aksesnya ke Pusat Data Nasional. Pengguna akan semakin bertambah selain karena daya beli yang tinggi pada perangkat komunikasi yang dapat digunakan untuk mengakses internet, dan juga apabila semakin banyak data dan informasi yang disediakan Pusat Data, dengan sendirinya pengguna akan tumbuh pesat.

### **1.4.2 e-Government**

Pusat Data Nasional mendukung terciptanya integrated e-Government. Saat ini banyak pemerintah daerah, instansi pemerintah dan lainnya menyediakan fasilitas Pusat Data secara tertutup untuk kepentingan masing-masing. Tidak tersedianya Pusat Data Nasional menyebabkan variasi infrastruktur Data Center sangat tinggi, dan munculnya pulau-pulau informasi yang tersebar hampir diseluruh instansi dan pemerintah daerah di Indonesia.

Pusat Data Nasional utamanya untuk memberikan fasilitas layanan untuk berjalannya e-Government. Pusat Data akan memberikan layanan integratif dan

terpercaya bagi penduduk, dan memberikan jaminan layanan e-government yang lebih stabil bagi pemerintah dan instansi lainnya. E-Government membutuhkan faktor kepercayaan pada kehandalan layanan, yang didukung dengan Pusat Data yang mumpuni. Pusat Data Nasional akan menjembatani terciptanya single ID, National Single Window, Integrasi Data dan Informasi, dan single source of data bagi negara Indonesia.

Pusat Data memberikan model layanan e-government yang akan lebih rapi dan terkelola secara integratif. Pusat Data memungkinkan juga layanan e0government lintas pemerintah daerah, instansi dan lembaga. Hal ini akan mendorong sinergi dan kerjasama untuk menciptakan efisiensi dan efektifitas proses bisnis pemerintahan dan reformasi birokrasi yang berkesinambungan.

### **1.4.3 Pendidikan**

Pusat Data dapat dimanfaatkan bagi lembaga-lembaga pendidikan untuk menempatkan sumber materi ajar, aktivitas belajar mengajar sampai dengan publikasi karya-karya akademik. Pusat Data yang mewakili kapasitas besar akan memudahkan dunia pendidikan Indonesia untuk selalu memproduksi dan menampung pengetahuan yang didapatkannya. Pendidikan akan menjadi lebih terencana, dan menaikkan kualitas pendidikan melalui manajemen pengetahuan yang terpusat.

Dunia pendidikan akan memanfaatkan Pusat Data dalam hal untuk akses terhadap bahan ajar yang shareable, tukar menukar pengetahuan, terwujudnya integrasi jenjang pendidikan, dan tentu saja aksesibilitas bagi siswa akan lebih mudah disediakan.

Pusat Data bagi dunia pendidikan akan menciptakan simpul-sipun education network yang membuat instansi pendidikan di Indonesia bukan bersaing fisik, tetapi berkolaborasi untuk mencerdaskan kehidupannya. Pendidikan Indonesia yang memiliki growth user yang itu pelajar, siswa akan terus berkembang banyak dan akan menjadi calon pengguna layanan secara umum. Pusat Data di pendidikan akan berkontribusi juga pada pola perilaku dalam memanfaatkan sumber daya data dan informasi dengan mengakses Pusat Data.

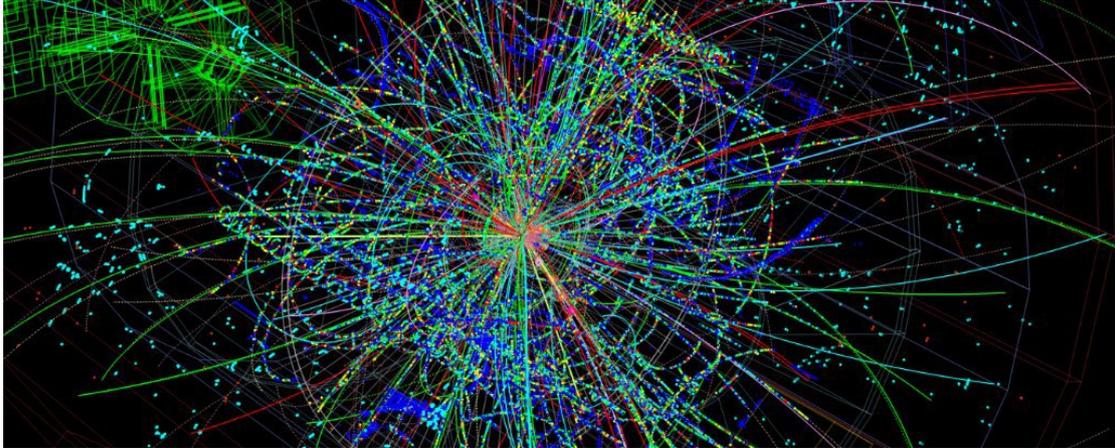
### **1.4.5 Lembaga Riset**

Pusat Data akan berperan sangat penting bagi Lembaga Riset. Lembaga Riset membutuhkan konektivitas, penyimpanan besar dan kemampuan komputasi yang cukup besar untuk keperluan penelitian. Lembaga Riset lazimnya akan memngumpulkan bahan, materi sampau dengan laporan penelitiannya kedalam Pusat Data Nasional, yang akhirnya akan mewujudkan sebuah Kumpulan-kumpulan inovasi Nasional dalam jangka panjang. Indonesia akan memiliki solusi-solusi masa depan berbasis penelitian yang sudah menjadi etalase di Pusat Data Nasional.

Lembaga Riset juga dapat membangu dalam hal pengelolaan dan pengaturan Pusat Data sesesua dengan standar industri, keamanan dan layanan yang prima. Pusat Data Nasional akan menunmbuhkan peluang-peluang inovasi dan kreativitas yang tinggi pada dunia ilmu pengetahuan bidang manapun.

## Bab 3. Pengembangan Pusat Data

---



Pusat Data Nasional tidak semudah membalikkan telapak tangan. Tantangan dalam pengembangan Pusat Data Nasional di Indonesia dapat dipahami memerlukan beberapa strategi agar dapat berhasil diwujudkan. Ada beberapa tantangan yang harus menjadi perhatian pada penyediaan Pusat Data Nasional.

### 3.1 Jaringan Data Nasional

Luar bentang pulau Indonesia yang sangat panjang, terpisah-pisah menyebabkan jaringan data nasional tidak bisa dengan mudahnya dengan model kabel dan lainnya. Jaringan Data Nasional Indonesia dapat menggunakan hybrid network yang fokusnya adalah perluasan coverage jaringan itu sendiri dengan media yang bisa dikombinasikan. Inisiatif awal dalam pembangunan jaringan data nasional baik melalui jalur nirkabel komunikasi selular, koneksi perkotaan berbasis dedicated line, jaringan komunikasi kabel, satelit hingga fiber optic sudah tersedia mayoritas di kota-kota besar.

### 3.2 Satelit

Satelit bagi Indonesia masih menjadi salah satu penyediaan jaringan data nasional yang ditujukan untuk melingkupi daerah-daerah yang tidak dapat terlindungi jalur data berbasis kabel. Satelit juga menimbulkan isu waktu dan banyaknya satelit yang harus mengitari wilayah Indonesia yang luas. Teknologi satelit sendiri belum 100%

dikuasai oleh Indonesia, sehingga ketergantungan terhadap pembuatan , peluncuran dan operasional belum 100% .

Satelit sebagai salah satu simpul jaringan data nasional masih sangat bisa diandalkan terutama untuk memberikan akses daerah-daerah terpencil atau teluk di Indonesia.

### **3.3 Kabel**

Jaringan kabel yang menghubungkan pusat-pusat pemerintahan, lembaga pendidikan, bisnis dan rumah-rumahan menjadi salah satu indikator broadband network index. Indonesia masih didominasi kota-kota besar yang memiliki jaringan data melalui kabel. Jaringan data nasional ini perlu diperluas ke kota-kota setelit dan kemudian dicabangkan kesimpul-simpul kota kecil lainnya.

Tidak dipungkiri investasi jaringan data nasional ini sangat besar. Pemerintah perlu berani melakukan share kepada industri untuk menyediakan jaringan data nasional ini. Beberapa negara sudah menggunakan standar kecepatan jaringan data berbasis kabel di Gigabyte speed, ukuran inilah yang harus dikejar terus menerus dalam penyediaan jaringan data nasional.

Aspek pemeliharaan jaringan data nasional, model redundansi jaringan data nasional, dan ring on jaringan data nasional juga harus dilengkapi lambat laun untuk meningkatkan jaminan layanan yang tinggi dan kemandirian konektivitas dari lainnya.

### **3.4 Server Farm**

Pusat Data biasanya terdiri atas banyak server yang dikelola dengan mekanisme tertentu agar dapat berbagi sumber daya, berbagi layanan dan dikelola dengan tingkat skalabilitas yang tinggi. Kumpulan server yang bekerjasama untuk menyediakan sebuah layanan Pusat Data biasanya disebut Server Farm. Teknologi yang menyediakan mekanisme ini sangat banyak dan beragam, sehingga Pusat Data cukup memiliki ketersediaan teknologi pendukung yang tidak perlu dikhawatirkan. Server Farm ini juga memberikan tawaran kekuatan komputasi yang besar dengan layanan yang menjanjikan. Rielnya, Server Farm akan memudahkan

pengelolaan berbagai macam tipe, spesifikasi, dan konfigurasi server kedalam layanan yang mudah dikelola.

### 3.5 Listrik

Berkumpulnya cukup banyak server dalam jumlah, tipe dan variasinya, kemudian digunakan untuk memproses beban tertentu, konsekuensi logisnya akan membutuhkan pasokan sumber daya listrik yang tidak sedikit. Pusat Data juga harus beroperasi penuh, minimal down time, dan memiliki kemampuan back up daya yang cukup menjadikan kebutuhan sumber daya listrik menjadi keharusan.

Penyediaan sumber tenaga listrik perlu juga diatur di Pusat Data agar tidak boros pemakaian energinya dan disesuaikan dengan beban/load pekerjaan di Pusat Data. Konsep Green Data Center, Renewable Data Center menjadi beberapa pilihan alternatif solusi penyediaan energi bagi Data Center.

### 3.6 Arsitektur

Data center atau dalam bahasa Indonesia disebut pusat data adalah suatu fasilitas tempat yang digunakan untuk meletakkan sistem komputer beserta komponen-komponennya seperti server, power capacity, dan network tool. Sistem ini meliputi sistem komunikasi dan penyimpanan data. Keberadaan data center sangat diperlukan oleh sebuah perusahaan bisnis, karena memiliki banyak manfaat untuk kinerja sebuah perusahaan yang profesional. Sebuah data center juga banyak memiliki banyak manfaat digunakan dalam organisasi maupun instansi pemerintahan.

Dalam pembangunan suatu data center perlu diperhatikan kriteria dalam suatu data center. Kriteria perancangan sebuah data center secara umum antara lain adalah:

- Ketersediaan

Data center diciptakan untuk mampu memberikan operasi yang berkelanjutan dan terus-menerus bagi suatu perusahaan baik dalam keadaan normal maupun dalam keadaan terjadinya suatu kerusakan yang berarti atau tidak. Data center harus dibuat sebisa mungkin mendekati zero-failure untuk seluruh komponennya.

- Scalability dan Flexibility

Data center harus mampu beradaptasi dengan pertumbuhan kebutuhan yang cepat atau ketika adanya servis baru yang harus disediakan oleh data center

tanpa melakukan perubahan yang cukup berarti bagi data center secara keseluruhan.

- Security

Data center menyimpan berbagai aset perusahaan yang berharga, oleh karenanya sistem keamanan dibuat seketat mungkin baik pengamanan secara fisik maupun pengamanan non-fisik.

Komponen-komponen utama yang terdapat pada *Data Center* adalah sebagai berikut :

1. Server

Merupakan komponen utama dari sebuah data center dimana dalam server dilakukan komputasi atau pengolahan proses dari sebuah data center.

2. Daya (*Power Capacity*)

Bertanggung jawab untuk mendistribusikan kekuatan listrik untuk peralatan listrik / elektronik yang berhubungan dengan data center.

3. Pendingin (*Cooling Capacity*)

Komponen ini digunakan untuk melakukan pendinginan mesin agar mesin tidak terlalu panas. Dikarenakan sebuah data center yang sibuk akan memiliki dampak atau efek berupa heat atau panas.

4. Kabel (*Cabling*) and Perangkat Jaringan

Kabel digunakan untuk menyambungkan/menghubungkan setiap komponen dari data center agar sebuah data center dapat bekerja. Dibutuhkan juga perangkat jaringan lainnya yang mensupport kerja Kabel, seperti switch.

5. Keamanan fisik (*Physical security*)

Dalam hal ini adalah casing dari data center, digunakan sebagai pelindung untuk melindungi kontak fisik dari luar, seperti debu dan benturan.

6. Ruang rak dan lantai (*Rack space and raised floor*)

- Rak digunakan untuk penyimpanan server di data center dan untuk pemanfaatan ruang yang lebih baik.
- Kebanyakan data center memiliki lantai terangkat. Bertujuan untuk memberikan ruang tambahan di bawah untuk pemasangan kabel dan tujuan ventilasi.

### 3.7 Disaster Recovery Center

Pusat Data dituntut pula untuk tidak memiliki downtime, intinya layanan terus berjalan apapun yang terjadi. Untuk memastikannya Pusat Data memerlukan fasilitas

kembarnya sebagai bagian strategi jaga-jaga apabila Pusat Data utama tidak beroperasi, yaitu Disaster Recovery Center.

DRC dibuat sama persisi dengan Pusat Data, diletakkan berbeda lokasi dengan standar tertentu, dan memiliki model pengelolaan yang sama dengan Pusat Data, termasuk mekanisme pengaktifan apabila terjadi sesuatu. DRC menjadi aset penting juga, karena secara otomatis akan memiliki semua aset data dan informasi dari Pusat Data secara real time atau terjadwal. DRC juga perlu memiliki kemampuan setara untuk memastikan tidak ada penurunan kualitas layanan Pusat Data secara keseluruhan.

### **Spare Parts**

Pusat data juga memerlukan jaminan ketersediaan suku cadang peralatannya. Pusat data memerlukan kapasitas cadangan untuk menyediakan suku cadang utama yang diperlukan apabila memerlukan penggantian. Suku Cadang ini sangat penting sehingga pada situasi terburuk penggantian hardware misalnya hanya membutuhkan waktu yang singkat dan diterima pengguna.

Pusat Data juga perlu mengelola interoperabilitasnya dengan suku cadang pendukung, apabila dukungan suku cadang sudah akan berakhir. Proses upgrading Pusat data harus dijadwalkan sesuai dengan antisipasi ketersediaan suku cadang perangkat keras lainnya.

### **Pengamanan**

Pusat Data sebagai objek vital dan aset memerlukan model pengamanan yang ekstra. Data dan informasi sebagai barang berharga memerlukan tempat berupa peti besi yang dijaga berlapis dan dengan mekanisme yang sangat aman. Pengamanan Data Center dapat dilakukan baik secara fisik dan cyber, kenapa? Karena tipe ancaman yang dapat menimpa Pusat Data adalah ancaman fisik dan ancaman cyber.

### **Fisik**

Pengamanan fisik dilakukan mekanisme untuk membuat aman fasilitas fisik, operasional, sistem dan pendukung dari Pusat Data. Pengamanan fisik didominasi dengan mekanisme pengamanan by human, memanfaatkan fasilitas fisik pendukung untuk memperkuat keamanan fisik Pusat Data.

## Pusat Data untuk Pemerintahan @ 2016

Fasilitas Pusat Data memiliki karakteristik reliabilitas layanan, skalabilitas sumber daya, stabilitas kinerja dan keamanan yang handal. Pusat Data memerlukan desain arsitektur, konfigurasi, dan implementasi layanan Pusat Data yang berorientasi pada layanan jangka panjang. Infrastruktur yang terbatas dengan ketersediaan sumber daya dan kebutuhan pemrosesan yang semakin tinggi membutuhkan manajemen sumber daya yang baik. Pusat Data perlu dibangun dengan pendekatan yang meminimalkan resiko baik secara ketersediaan sumber daya dan keamanannya.

Sebagai sebuah organisasi yang sangat kompleks, dengan unit kerja dan proses bisnis yang sangat banyak, sudah menjadi keharusan bagi universitas untuk memiliki Pusat Data (Data Center) yang memenuhi standar.

Pusat data tersebut selain berfungsi untuk menempatkan komputer server yang memuat seluruh aplikasi beserta komponen-komponen terkaitnya, seperti sistem telekomunikasi dan penyimpanan data harus memiliki desain dan perencanaan yang memperhatikan minimum aspek-aspek berikut :

1. Lokasi aman, memenuhi syarat sipil bangunan, geologi, vulkanologi, dan topografi.
2. Terproteksi dengan sistem cadangan, untuk sistem catudaya, pengatur udara/lingkungan, dan komunikasi data.
3. Menerapkan tata kelola standar data center meliputi :
  - Standar Prosedur Operasi
  - Standar Prosedur Perawatan
  - Standar dan Rencana Pemulihan dan Mitigasi Bencana
  - Standar Jaminan Kelangsungan Bisnis

### 3.8 Cyber Security

Indonesia membutuhkan model dan pola mengelola informasi dalam perspektif *secure* dan *defense*. *Secure* memberikan kemampuan bagi individu, organisasi atau entitas untuk secara mandiri memberikan jaminan keamanan yang dibangun secara *by design*. Entitas secara sengaja dan terencana mengelola sumber daya secara aman, baik dari gangguan internal maupun eksternal. Entitas *by process* tumbuh untuk menginisiasi dan mengembangkan model informasi yang aman secara alamiah, *step by step* mengikuti pola dari kajian ilmiah atau standar industry.

*Defense* memiliki pendekatan berbeda dalam mengelola sumber daya secara aman. Pada perspektif *defense*, sumber daya dikelola secara aktif dengan menginisiasi model keamanan yang preventif. Mekanisme entitas untuk menganalisa dan mencegah secara aktif dalam mencegah resiko keamanan. Tujuannya adalah mencegah resiko terjadi dan resiko yang sudah ada membesar.

Proses untuk membangun, mengelola dan secara terus menerus meningkatkan kemampuan *secure* dan *defense* menjadi tanggungjawab Pemerintah, industri, dan komunitas. Pola mengelola sumber daya informasi untuk keamanan ini mengoptimalkan dan memberdayakan kontribusi masing-masing pihak dalam merangkai tata kelola *security* yang baik. Pemerintah memberikan kepastian hukum dalam aktivitas *secure* dan *defense*, termasuk juga membentuk entitas negara yang mengkoordinasikan pola keamanan yang dibangun untuk negara. Beberapa pendekatan negara lain melakukannya melekat pada instrument pertahanan dan keamanan negara. Industri memberikan *best practice* dalam penyelenggaraan tata kelola *security* baik instrument infrastruktur, *tools*, dan studi kasus yang dipunyai. Komunitas termasuk didalamnya peneliti, perguruan tinggi, pemerhati keamanan dapat dikelola untuk membangun *security network* yang luas.

Pendekatan tata kelola dan manajemen keamanan dalam perspektif nasional adalah membangun kepedulian dan isu bersama yang diterima oleh *stakeholder*.

1. Pada tahap berikutnya membangun infrastruktur dan entitas pengelola keamanan informasi.
2. Tahap ini membangun struktur yang bisa nantinya berkembang sesuai tuntutan isu dan trend keamanan itu sendiri.

3. Tahap ketiga adalah membangun awareness dan kampanye yang terencana akan pentingnya data, informasi dan asset.
4. Tahap keempat adalah membangun aksi bersama untuk melakukan respon bersama terhadap perkembangan dari keamanan itu sendiri.
5. Pada akhirnya, Indonesia akan memiliki mekanisme *best practice* dalam pengelolaan keamanan informasi dengan dibangunnya *knowledge based/management on security*.

Soft Backup memfokuskan pada pola dan strategi pencadangan perangkat lunak, dan data. Beberapa strategi yang disiapkan adalah sebagai berikut:

### **Software**

setiap perangkat lunak yang dipakai baik dalam administrasi perkantoran sampai dengan dukungan teknis minila memiliki paket instalasi sebagai berikut:

### **Ready to Install**

Aplikasi memiliki cadangan yang bersifat siap di install. Sifat ini dipakai untuk aplikasi-aplikasi yang bersifat sistem operasi dan aplikasi pendukung yang tidak memiliki kerekatan tinggi dengan data (data terpisah dengan aplikasi)

### **Ready to restore**

Aplikasi memiliki cadangan yang bersifat siap di restore. Sifat ini dipakai untuk aplikasi-aplikasi yang secara operasional sangat terikat dengan data. Sehingga ketika terjadi insiden potensi data loss sangat tinggi, misalnya Database, file management dan lainnya

### **Data**

Prinsip DRC adalah data lebih berharga dari sistem, sehingga pola penanganan software data harus dilakukan secara terencana dan baik. Terkait dengan segala infrastruktur penyimpanan. Isu yang diangkat antara lain adalah arsitektur SAN, fibre channel switching, replikasi, backup serta archival.

### **Integrasi Aplikasi**

Integrasi aplikasi menjadi suatu hal yang tidak dapat diabaikan. Banyaknya ragam aplikasi yang perlu dikembangkan dan dioptimalkan membawa konsekuensi

kebutuhan integrasi satu aplikasi dengan aplikasi yang lain. Mekanisme integrasi dapat dilakukan dengan cara memetakan interkoneksi proses bisnis antar aplikasi yang kemudian dikembangkan dengan mekanisme *web services* ataupun *sharing database*

### 3.9 Manajemen

Manajemen Pusat Data Nasional dikelola secara bersama-sama oleh stakeholder, yaitu Pemerintah yang diwakili oleh Kemkominfo RI, Perwakilan Pihak Swasta, Militer dan Akademisi. Hal ini lazim dilakukan untuk komite sejenis dibidang teknologi informasi dan komunikasi. Konteks Pusat data ini, menjadikan Indonesia harus memiliki Komite Teknologi Informasi dan Komunikasi Nasional yang bekerja dibawah supervisi Menteri Kominfo.

Pemerintah memegang peranan penting untuk memastikan Pusat Data memenuhi prinsip dan mplementasi teknologi informasi agar mampu mendukung optimalisasi *key success factor* sebagai berikut:

#### 1. **Cost Optimization**

*Cost Optimization* merupakan *key success factor* dari strategi **Enable the business through cost effective timely IT Solution**. Berkaitan dengan faktor ini, perlu menganalisis kemungkinan yang dapat diperoleh dari implementasi teknologi agar solusi TI menjadi lebih efektif dari sisi biaya dan waktu untuk dapat digunakan.

#### 2. **Support Availability**

*Support Availability* merupakan *key success factor* dari strategi **Provide reliable continuous service delivery**. Faktor ini meliputi 2 hal yang perlu menjadi perhatian, yang pertama adalah faktor keberlanjutan pengembangan sistem informasi dari sisi pengembang dan yang kedua adalah ketersediaan dukungan baik teknis dan non-teknis yang memastikan penggunaan sistem dalam jangka waktu yang lama.

#### 3. **Security & Scalability**

*Security & Scalability* merupakan *key success factor* dari strategi **Build robust modular & scalable IT infrastructure**. Semangat pengembangan sistem adalah keterbukaan dan kebersamaan. Hal ini tentu saja perlu mendapatkan perhatian lebih pada saat sistem akan diimplementasikan. Selain itu perlu diketahui juga bagaimana kemampuan sistem untuk dapat diintegrasikan dengan sistem-sistem lain yang sudah digunakan sebelumnya dan umumnya merupakan *proprietary software*.

#### 4. **Compliance & Risk Management**

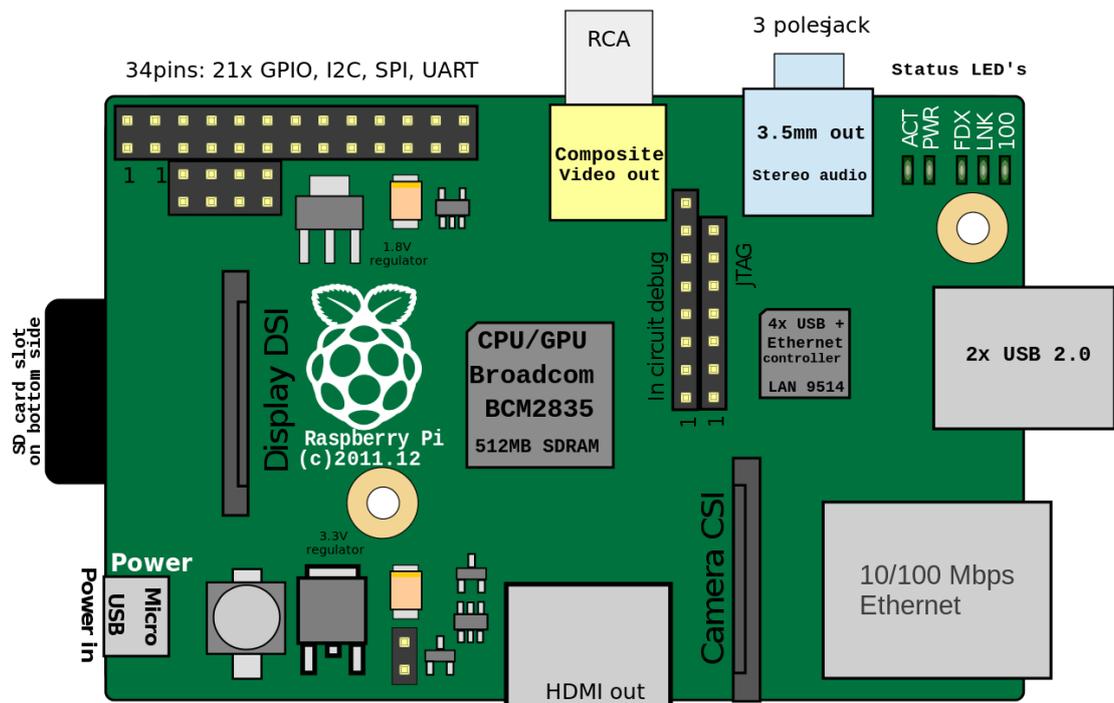
*Compliance & Risk Management* merupakan *key success factor* dari strategi **Ensure compliance and governance**. Tidak ada produk tanpa kelemahan, dan setiap kelemahan pasti mengandung resiko. Faktor ini berhubungan erat dengan regulasi yang mengatur manajemen resiko. Berkaitan dengan hal tersebut perlu mengidentifikasi secara seksama apa saja resiko yang mungkin timbul dari implementasi sistem dan bagaimana strategi antisipasinya. Selain itu perlu dipastikan bahwa implementasi sistem tidak bertentangan dengan segala regulasi yang harus dipatuhi lembaga.

#### 5. **Resource Management**

*Resource Management* merupakan *key success factor* dari strategi yang menjadi pondasi dari strategi-strategi lainnya yaitu **Competent and engaged IT Human Resources**. Ketersediaan sumber daya manusia (SDM) baik *fresh graduate* maupun tenaga ahli berpengalaman yang mampu mengembangkan, memodifikasi, menggunakan dan memelihara suatu teknologi menjadi syarat mutlak agar implementasi teknologi berjalan baik. Poin ini menjadi tantangan yang harus distrategikan bagaimana menyiapkan komposisi SDM baik di sisi *end-user* maupun sisi *developer* dan *administrator* yang menguasai seluk-beluk sistem informasi.

## Bab 4. Pengembangan Data Center

Pada awalnya, data center selalu dibangun dalam sebuah ruangan yang besar untuk membantu operasi dari suatu organisasi dan digunakan komputer-komputer super yang kompleks dalam hal pemeliharaan dan pengoperasian. Komponen - komponen yang digunakan dalam data center cukup banyak, sehingga membutuhkan kabel yang banyak untuk konektivitas dari semua komponen tersebut, server yang digunakan kompleks mulai dari ukurannya yang besar sampai dengan orang yang handal dalam menangani server tersebut, serta membutuhkan daya yang sangat besar. Hal ini juga memicu para pakar untuk bekerja mencari solusi data center yang baik. Dengan komponen yang kompleks dan banyak maka akan sebanding pada sisi biaya. Untuk pembangunan suatu data center biaya yang dibutuhkan sangat besar, ini bisa dilihat dari pemeliharaan data center yang khusus, peralatan khusus, sampai dengan orang specialist untuk menangani data center ini. Tetapi hal tersebut dapat di sederhanakan dengan membuat sebuah mini data center dengan menggunakan Raspberry Pi.



Dengan hanya menggunakan Raspberry Pi sebagai mini data center, data center yang akan dibuat dapat dengan mudah dirancang dan dilakukan maintenance. Banyak manfaat lain dalam penggunaan Raspberry Pi sebagai sebuah mini data center, diantaranya :

- Dokumentasi sudah banyak beredar sehingga mudah dalam melakukan modifikasi
- Ukuran yang kecil dapat mengurangi penggunaan space
- Selain dapat mengurangi space, ukuran tersebut membuat raspberry sangat fleksible untuk dibawa.

Dalam sisi arsitektur, mini data center ini akan menggunakan 3 Raspberry Pi yang digunakan sebagai server. Dimana masing-masing server atau Raspberry Pi memiliki peran atau fungsi yang berbeda-beda. Masing-masing Raspberry Pi tersebut digunakan untuk Web Server, Database Server dan File Server. Dalam implementasiya ketiga server tersebut akan dihubungkan dalam sebuah jaringan dengan menggunakan switch, serta power capacity yang digunakan sebagai sumber energi.



#### 4.1 Web Server Management

Web Server adalah sebuah perangkat lunak (*software*) di dalam server yang berfungsi untuk menerima permintaan (*request*) dari klien dan mengirimkan respon

dari permintaan tersebut ke *client*. Permintaan ini biasanya berupa halaman web dan dikirim melalui protokol HTTP atau HTTPS dengan menggunakan *web browser*. Respon yang dikirim oleh server berupa halaman web, umumnya berbentuk sebagai dokumen HTML (Hypertext Markup Language). Fungsi utama dari web server adalah mengirimkan berkas yang diminta oleh klien melalui suatu protokol komunikasi tertentu.

Nginx adalah salah satu contoh dari sekian banyak perangkat lunak web server yang sering digunakan. Nginx web server dengan kode sumber terbuka (*open source*) yang dibuat oleh Igor Sysoev pada tahun 2002 dan dirilis ke publik pada tahun 2004. Web server lainnya adalah Apache, Apache Tomcat, Lighttpd, Litespeed, Zeus Web Server, dan lain-lain.

Nginx merupakan web server yang terkenal sangat ringan, cepat dan minim dalam penggunaan memori. Nginx dirancang untuk digunakan dalam mesin yang mempunyai spesifikasi perangkat keras yang minimal. Oleh karena itu, nginx cocok untuk perangkat raspberry pi.

Untuk melakukan instalasi nginx pada raspberry pi, kita perlu mengetahui IP Address dari raspberry pi tersebut dan melakukan koneksi menggunakan ssh agar dapat masuk ke dalam shell atau terminal dari raspberry. Jika kita menghubungkan raspberry menggunakan monitor, maka tidak diperlukan koneksi ssh, hanya perlu dengan membuka terminal raspberry dari desktop. Langkah instalasinya adalah sebagai berikut :

1. Update repositori dari raspberry pi. Hal ini dimaksudkan agar paket aplikasi dari raspberry kita tersinkronisasi dengan server.

```
$ sudo apt-get update
```

2. Lakukan instalasi Web Server NGINX dengan segala komponen tambahannya :

```
$ sudo apt-get install nginx php5-fpm php5-mysql php5-gd php5-curl
```

3. Lakukan konfigurasi PHP

```
$ sudo nano /etc/php5/fpm/php.ini
```

4. Cari tulisan "cgi.fix\_pathinfo". Hapus tanda ";" di depannya dan ubah nilainya dari 1 menjadi 0.

```
cgi.fix_pathinfo=0
```

5. Save dan close dengan cara tekan ctrl+x >> y >> enter.
6. Lakukan konfigurasi file www.conf :

```
$ sudo nano /etc/php5/fpm/pool.d/www.conf
```

7. Cari tulisan "listen" dan ubah nilainya menjadi berikut :

```
listen = /var/run/php5-fpm.sock
```

8. Save dan close.
9. Restart service dari PHP.

```
$ sudo service php5-fpm restart
```

10. Backup konfigurasi default dari nginx, supaya ketika kita ada kesalahan konfigurasi, masih bisa mengembalikan ke konfigurasi bawaan nginx.

```
$ sudo cp /etc/nginx/sites-available/default /etc/nginx/sites-available/default.bak
```

11. Buka konfigurasi default nginx.

```
$ sudo nano /etc/nginx/sites-available/default
```

12. Ubah konfigurasinya menjadi seperti ini :  
Hal ini dilakukan agar nginx bisa memproses file berekstensi php.

```
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
    root /var/www/html;  
    index index.php index.html index.htm index.nginx-debian.html;  
    server_name localhost;  
  
    location / {  
        try_files $uri $uri/ =404;  
    }  
}
```

```
location ~ \.php$ {
    include snippets/fastcgi-php.conf;
    fastcgi_pass unix:/var/run/php5-fpm.sock;
}

location ~ /\.ht {
    deny all;
}
}
```

13. Save dan close.
14. Lakukan restart pada service nginx.

```
$ sudo service nginx restart
```

## 4.2 Database Server Manajemen

Database adalah program komputer yang menyediakan layanan data ke komputer atau program komputer. Sedangkan istilah Database Server ini merujuk kepada sebuah komputer yang didedikasikan untuk menjalankan program server database. Database sistem manajemen database yang sering menyediakan fungsi server, dan beberapa DBMSs (misalnya, MySQL) secara eksklusif bergantung pada model klien-server untuk akses data.

Database server menyediakan beberapa manfaat yaitu:

1. Semua data untuk organisasi dapat disimpan di satu lokasi.
2. Database server menambahkan tingkat keamanan data.
3. Database server menyediakan layanan database management service dimana data disusun dengan cara tertentu sehingga meningkatkan pencarian dan pengambilan data.
4. Beberapa client dapat mengakses data yang disimpan di database server dalam satu waktu tanpa saling mengganggu satu sama lain.

Dalam database server, sering digunakan model Client-Server. Dimana model client-server dapat diartikan sebagai model dari suatu sistem yang membagi proses

sistem antara server yang mengolah database dan client yang menjalankan aplikasi. Database server mengurangi beban akses data oleh client pada server. Database dapat diakses oleh beberapa client secara bersamaan dimana data yang diakses hanya atau diubah berasal dari satu sumber yaitu database pada server.

Untuk melakukan manajemen database, dibutuhkan suatu DBMS atau Database Management System yang berfungsi untuk mengontrol seluruh transaksi yang terjadi dalam satu database. Salah satu DBMS yang sering dipakai dan yang akan digunakan adalah MySQL. MySQL adalah sistem manajemen database SQL yang bersifat Open Source dan paling populer saat ini. Sistem Database MySQL mendukung beberapa fitur seperti multithreaded, multi-user, dan SQL Database Management System (DBMS). Database ini dibuat untuk keperluan sistem database yang cepat, handal dan mudah digunakan.

Untuk mengimplementasikan MySQL pada mini data center yang akan di buat langkah pertama adalah instalasi MySQL Server. Berikut langkah dari intaasi MySQL Server:

1. Tutup koneksi ke web-server dan lakukan koneksi ke db-server dengan putty :

```
IP Address = 192.168.1.102
Port      = 22
```

2. Sebelum instalasi lakukan perintah berikut :

```
$ sudo apt-get update
```

3. Lakukan instalasi MySQL Server dengan perintah :

```
$ sudo apt-get install mysql-server
```

4. Selama proses instalasi, akan ditanyakan password root untuk mysql server. Silahkan isi password dengan "raspberry" atau sesuai dengan keinginan, dan jangan sampai lupa.

5. Setelah selesai proses instalasi, jalankan perintah instalasi database.

```
$ sudo mysql_install_db
```

6. Selanjutnya jalankan perintah berikut untuk hal keamanan mysql.

```
$ sudo mysql_secure_installation
```

7. Silahkan masukkan root password dari mysql ketika diminta. Selanjutnya, akan ditanya apakah ingin mengganti root password, silahkan tekan "N" jika tidak ingin mengganti password.
8. Selanjutnya akan ditanyakan banyak pertanyaan, tekan saja ENTER untuk memilih pilihan default.

Setelah melakukan instalasi MySQL Server perlu dilakukan konfigurasi MySQL untuk memperbolehkan remote access. Berikut langkahnya :

1. Buka file konfigurasi MySQL

```
$ sudo nano /etc/mysql/my.cnf
```

2. Silahkan cari tulisan "bind-address" dan ganti nilainya dengan IP dari Database Server (dalam hal ini 192.168.1.102).

```
bind-address = 192.168.1.102
```

3. Save dan close dengan cara tekan ctrl+x >> y >> enter.
4. Supaya konfigurasi baru dapat berubah, restart MySQL.

```
$ sudo service mysql restart
```

### 4.3 File Server Manajemen

File server adalah sebuah computer terpasang ke jaringan yang memiliki tujuan utama menyediakan lokasi untuk akses share / berbagi disk, yaitu berbagi penyimpanan file computer (seperti document, file suara, foto, video/film, dan gambar) yang dapat di akses oleh workstation yang melekat pada jaringan computer. Sebuah file server biasanya tidak melakukan suatu perhitungan, dan tidak menjalankan program atas nama client. Hal ini dirancang terutama untuk memungkinkan penyimpanan cepat dan pengambilan data dimana perhitungan berat di sediakan oleh workstation.

Untuk langkah-langkah instalasi file server adalah sebagai berikut :

1. Instalasi NFS untuk File Server
  - a. Lakukan perintah update.

```
$ sudo apt-get update
```

- b. Lakukan instalasi NFS server dengan perintah :

```
$ sudo apt-get install nfs-kernel-server
```

## 2. Buat Share Directory pada File Server

- a. Buat suatu direktori untuk penempatan file seafile.

```
$ sudo mkdir /mnt/seafiler-server
```

- b. Ubah ownership dari direktori tersebut agar remote host (web-server) bisa melakukan baca tulis pada direktori tersebut dengan menggunakan web browser.

```
$ sudo chmod 777 /mnt/seafiler-server  
$ sudo chown nobody:nogroup /mnt/seafiler-server
```

## 3. Konfigurasi NFS Exports pada File Server

- a. Buka /etc/exports

```
$ sudo nano /etc/exports
```

- b. Lakukan konfigurasi agar client (web-server) dalam melakukan koneksi ke NFS Server. Umumnya format konfigurasinya seperti ini :

```
directory_to_share client(share_option1,...,share_optionN)
```

- c. Tambahkan baris berikut di paling bawah :

```
/mnt/seafiler-server 192.168.1.101(rw, sync, no_subtree_check)
```

- d. Berikut akan dijelaskan arti dari opsi NFS tersebut :

- **rw** : opsi ini berarti memberikan akses read dan write pada komputer klien terhadap direktori tersebut.
- **sync** : opsi ini memaksa NFS untuk menulis perubahan dahulu sebelum membalas. Hal ini untuk mencegah terjadinya file corruption.

- **no\_subtree\_check** : secara default, server akan memeriksa apakah file yang diminta klien berada pada partisi yang sesuai. Jika opsi ini diaktifkan, server tidak melakukan pemeriksaan dan dapat mempercepat proses transfer file.
- **no\_root\_squash** : secara default, user root pada klien yang melakukan request akan dianggap sebagai user nobody oleh server. Jika opsi ini diaktifkan, user root pada klien memiliki hak yang sama dengan user root pada server.

e. Save dan close dengan cara tekan ctrl+x >> y >> enter.

f. Ekspor ulang direktori yang baru saja ditambahkan :

```
$ sudo exportfs -a
```

g. Setelah itu restart service dari NFS untuk mengaplikasikan perubahan yang telah dibuat.

```
$ sudo service nfs-kernel-server restart
```

4. Memperbaiki Error pada NFS Server tidak dapat Start saat Boot.

a. Pada Raspbian Jessie, service nfs-kernel-server tidak dapat berjalan saat boot. Sehingga kita perlu melakukan restart ulang service nfs-kernel-server setelah boot. Ada sedikit trik untuk memperbaiki hal tersebut, yaitu dengan menggunakan crontab.

b. Pertama, buat terlebih dahulu bash script untuk melakukan restart nfs-kernel-server.

```
$ cd  
$ mkdir script/  
$ cd script/  
$ nano nfs-restart.sh
```

c. Tulislah seperti ini :

```
#!/bin/bash  
  
sudo service nfs-kernel-server restart
```

- d. Save dan close dengan cara tekan ctrl+x >> y >> enter.
- e. Setelah itu, lakukan seting pada crontab.

```
$ sudo crontab -e
```

- f. Jika ditanya editor yang dipakai, pilih saja nano.
- g. Tambahkan baris berikut ini di paling bawah :

```
@reboot /home/pi/script/nfs-restart.sh
```

- h. Save dan close dengan cara tekan ctrl+x >> y >> enter.
- i. Dengan crontab ini, maka nfs-kernel-server yang sebelumnya tidak dapat berjalan saat boot, sekarang telah bisa berjalan saat boot. Maka ketika client booting, tidak akan gagal melakukan mount ke direktori nfs.

#### 5. Instalasi NFS pada klien (Web Server)

- a. Tutup koneksi ke file-server dan lakukan koneksi ke web-server dengan putty :

```
IP Address = 192.168.1.101  
Port      = 22
```

- b. Lakukan instalasi NFS Client.

```
$ sudo apt-get install nfs-common
```

#### 6. Buat Mount Point pada Web Server

- a. Buat direktori untuk mounting NFS.

```
$ sudo mkdir -p /home/seafile/seafile-data
```

- b. Lakukan mounting direktori file server ke direktori yang baru saja dibuat.

```
$ sudo mount 192.168.1.103/mnt/seafile-server /home/seafile/seafile-data
```

\* note : Jika gagal mounting, coba restart service nfs-kernel-server pada file-server.

- c. Untuk mengecek apakah sudah ter-mounting dengan benar, bisa dilakukan dengan perintah :

```
$ df -h
```

- d. Jika berhasil, maka akan muncul seperti ini :

```
Filesystem                Size  Used Avail Use% Mounted on
/dev/root                  15G  3.5G  11G  25% /
devtmpfs                  214M    0  214M   0% /dev
tmpfs                     218M    0  218M   0% /dev/shm
tmpfs                     218M  4.5M  213M   3% /run
tmpfs                     5.0M  4.0K  5.0M   1% /run/lock
tmpfs                     218M    0  218M   0% /sys/fs/cgroup
/dev/mmcblk0p1             60M   20M  41M  34% /boot
tmpfs                     44M    0  44M   0% /run/user/1000
192.168.1.103:/mnt/seaf... 3.5G  3.3G   69M  98% /home/seaf.../seaf...-data
```

- e. Jika ingin melihat semua NFS Shares yang telah ter-mount pada web-server, bisa dilakukan dengan perintah :

```
$ mount -t nfs4
```

- f. Maka, hasilnya akan seperti ini :

```
192.168.1.103:/mnt/seaf... on /home/seaf.../seaf...-data type nfs4
(rw,relatime,vers=4.0,rsiz...=65536,wsiz...=65536,namlen=255,hard,proto=tcp,port=0,t
imeo=600,retrans=2,sec=sys,clientaddr=0.0.0.0,local_lock=none,addr=192.168.1.10
3)
```

## 7. Membuat Mounting Otomatis pada Direktori NFS

- a. Buka file fstab.

```
$ sudo nano /etc/fstab
```

- b. Tambahkan konfigurasi berikut di baris paling bawah.

```
# Konfigurasi automount NFS saat boot
```

```
192.168.1.103:/mnt/seafiler-server /home/seafiler/seafiler-  
data nfs rw 0 0
```

## 4.4 Implementasi Security

Data Center memegang peranan penting dalam operasional suatu perusahaan, jika data center mereka diserang atau diretas, maka operasional perusahaan bisa terganggu, bahkan informasi rahasia perusahaan bisa diambil oleh orang yang tidak berhak. Oleh karena itu keamanan merupakan hal yang penting dan perlu diperhatikan dalam mendesain infrastruktur data center. Keamanan data center dapat dibedakan menjadi 2 bagian, yaitu keamanan fisik dan keamanan logika / software. Ada tiga aspek yang perlu diperhatikan dalam keamanan logika, yaitu :

- *Confidentiality* (Kerahasiaan)

*Confidentiality* artinya memastikan bahwa data dan informasi tidak bisa di akses ataupun di baca oleh pihak yang tidak berwenang.

- *Integrity* (Integritas)

*Integrity* artinya memastikan bahwa data dan informasi tidak diubah oleh pihak yang tidak berwenang.

- *Availability* (Ketersediaan)

*Availability* artinya memastikan bahwa data dan informasi harus selalu tersedia ketika ada yang membutuhkan.

Pada perancangan data center ini, kami melakukan beberapa konfigurasi keamanan sebagai berikut :

1. Membatasi akses root dari mysql dengan password. Memberi password untuk akun root dari mysql bisa dilakukan ketika pertama kali melakukan instalasi mysql atau bisa dilakukan dengan perintah berikut :

```
$ mysqladmin -u root password newpass
```

Kemudian masukkan password baru untuk akun root dari mysql.

2. Membatasi akses root dari mysql agar hanya bisa diakses melalui DB Server. Akun root mysql sebaiknya hanya boleh diakses dari localhost server saja, yaitu DB Server. Membuat akun root bisa di akses oleh server lain ataupun dari luar

membuat segala database yang ada di server menjadi rawan. Untuk membatasi akses, bisa dilakukan dengan perintah berikut dan ikuti langkahnya :

```
$ sudo mysql_secure_installation
```

3. Membuat user mysql khusus untuk aplikasi tertentu dan membatasi akses. Dengan membuat user khusus untuk database dari suatu aplikasi tertentu dan membatasi aksesnya hanya untuk IP tertentu, maka hal ini akan meningkatkan keamanan pada database server. Cara untuk membuat user dan membatasi akses adalah sebagai berikut :

```
$ mysql -u root -p
```

```
mysql> CREATE DATABASE `nama_database`;  
mysql> CREATE USER 'nama_user'@'192.168.1.1' IDENTIFIED BY 'password';  
mysql> GRANT ALL PRIVILEGES ON `nama_database`.* TO 'nama_user'@'192.168.1.1';
```

Ganti *nama\_database* dengan nama database yang diinginkan, *nama\_user* dengan nama user mysql yang diinginkan, *192.168.1.1* dengan IP Address yang dapat meremote user tersebut, *password* dengan password yang diinginkan, dan *ALL PRIVILEGES* dengan privileges yang akan diberikan ke user tersebut, seperti misalnya *create*, *delete*, *insert*, *select*, *update* dan lain-lain, bisa dilihat pada dokumentasi mysql.

4. Membatasi akses NFS mounting pada File Server NFS atau Network File System pada file server harus dilakukan pembatasan akses terhadap mounting pada direktori yang akan dibagikan. Hal ini akan membuat direktori yang akan kita bagikan hanya bisa di-mounting di IP Address tertentu. Caranya adalah pada saat melakukan konfigurasi NFS Export pada file server, kita memberikan pengaturan IP Address sebagai berikut:

```
/mnt/seafiler-server 192.168.1.101(rw,sync,no_subtree_check)
```

“*/mnt/seafiler-server*” adalah direktori dari file server yang akan kita bagikan ke client. “*192.168.1.101*” adalah IP Address dari client, yang artinya hanya koneksi dari IP tersebut yang dapat melakukan mounting NFS dari file server.

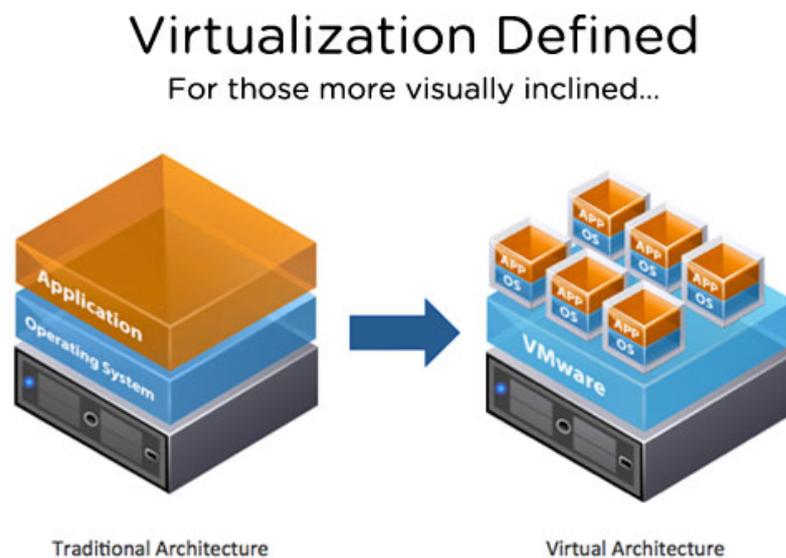
## Bab 5. Virtualisasi

---

Bab ini menjelaskan konsep-konsep yang digunakan dalam penelitian. Pada bagian awal akan dijelaskan mengenai konsep virtualisasi *server* dan konsep *Data Center* secara umum yang menjadi dasar untuk menyelesaikan permasalahan pada penelitian ini..

### 5.1 Virtualisasi Server

Evolusi teknologi virtualisasi dipicu oleh berkembangnya level abstraksi dari perangkat keras dan perangkat lunak. Penggunaan bahasa pemrograman tingkat tinggi seperti BASIC, C, Java, Python membuat pemrogram aplikasi tidak perlu lagi memikirkan segala hal mengenai sistem operasi dimana aplikasi tersebut akan berjalan. Hal ini dimungkinkan karena salah satu tugas sistem operasi adalah menyediakan abstraksi yang membebaskan program dari hal yang kompleks dan detail mengenai manajemen memori dan *input/output* (I/O). Walaupun begitu, sistem operasi harus mengetahui pada perangkat keras mana dia berada. Informasi *state* seperti alamat MAC dan IP, konfigurasi memori, dan perhitungan prosesor terikat pada *state* dari sistem operasi pada saat instalasi sistem. Informasi *state* ini mengunci sistem operasi pada perangkat keras spesifik dimana dia diinstal dan akan menyulitkan apabila diperlukan proses pemulihan kegagalan perangkat keras atau pada saat peningkatan sistem (*maintenance*).



Para arsitek perangkat keras dan perangkat lunak kemudian memikirkan bagaimana

menghadapi kekurangan ini dengan memikirkan konsolidasi. Awal mula teknologi virtualisasi berasal dari teknologi memori *virtual*. Konsep teknologi memori *virtual* pertama kali dikembangkan oleh fisikawan Fritz-Rudolf Güntsch dari Technische Universität Berlin pada tahun 1956. Dalam tesis doktoralnya ia mendeskripsikan mesin dengan 6 buah 100-blok *word* memori inti utama dan ruang alamat 1000 100-blok *word*, dengan perangkat keras yang secara otomatis memindahkan blok antara memori utama dan memori sekunder. Komputer The Atlas dari Universitas Manchester adalah sistem pertama yang membangun teknologi memori *virtual* ini (Scheffy, 2007). Teknologi memori *virtual* membuat sistem dengan jumlah memori fisik terbatas membuatnya seolah-olah memiliki memori lebih banyak dari memori fisik. Pada komputer The Atlas, sistem operasi menyimpan semua *image* memori dan data aplikasi pada memori sekunder, yaitu *hard drive* sistem, dan mengirim bagian dari *image* memori yang dibutuhkan kepada memori DRAM ketika program dieksekusi.

Pada saat ini teknologi virtualisasi telah berkembang pada berbagai sisi teknologi, terutama pada virtualisasi *server* dan penyimpanan (*storage*). Cheng (2014) menyebutkan bahwa mesin *server* yang berkembang semakin cepat membuat kita relatif sulit untuk memakai semua sumber daya tersebut apabila kita hanya menginstal satu sistem operasi. Selain itu biaya untuk menyewa *server* pada Data Center cenderung mahal. Virtualisasi *server* mengatasi masalah ini dengan cara melakukan sentralisasi beberapa *server* menjadi sebuah sistem tunggal dan membuat beberapa objek sistem di dalamnya yang disebut mesin *virtual*. Virtualisasi *server* mengimplikasikan setiap mesin *virtual* memiliki CPU sendiri (disebut vCPU), memori, *hard disk* dan kartu jaringan sendiri sesuai dengan kebutuhan pengguna. Dibuatlah perangkat lunak *hypervisor*, atau disebut juga sebagai *virtual machine manager* (VMM), untuk mengatur pembagian memori pada perangkat. *Hypervisor* membuat beberapa sistem operasi *guest* pada mesin *virtual* untuk bisa berjalan bersama-sama pada komputer *host*. Fungsi *hypervisor* ada diantara sistem operasi komputer dan *kernel* perangkat keras. Ada dua jenis *hypervisor*, yaitu *hypervisor native* atau *bare metal* yang berjalan langsung pada perangkat keras *host*, dan *hypervisor hosted* yang berjalan dalam lingkungan sistem operasi.

## 5.2 Penggunaan Virtualisasi

Pada subbab sebelumnya telah diketahui bahwa penggunaan awal virtualisasi adalah sebagai proses melakukan konsolidasi. Pada virtualisasi *server*, konsolidasi

*server* berarti mengambil beberapa bagian dari *server* berbeda dan kemudian melakukan migrasi menjadi mesin *virtual* yang berjalan pada *server* tunggal. Lebih lanjut, konsolidasi berarti juga mengambil sejumlah *server* berbeda dan melakukan migrasi menjadi sejumlah *server* yang lebih sedikit, dengan beberapa mesin *virtual* yang berjalan pada tiap *server*. Dengan konsolidasi *server* dihasilkan penggunaan perangkat keras, energi dan pendinginan, serta lisensi perangkat lunak yang lebih sedikit yang akhirnya meningkatkan utilisasi. Berikut beberapa penggunaan virtualisasi (Golden, 2011):

- **Pengujian dan Pengembangan Aplikasi**

Pengujian dan pengembangan aplikasi pada beberapa kasus perlu dilakukan pada beberapa sistem operasi yang berbeda atau versi yang berbeda dari sistem operasi. Hal ini dilakukan supaya memenuhi persyaratan aplikasi untuk fungsionalitas, skalabilitas, atau untuk melihat dimana terjadi ketidaksesuaian. Pengujian dan pengembangan aplikasi seperti ini membutuhkan sistem dengan berbagai konfigurasi. Dengan menggunakan virtualisasi, penguji atau pengembang bisa mereplikasi sistem dalam satu perangkat keras. Ini berarti meminimalkan penggunaan *server* yang hanya dipakai beberapa waktu saja.

Dalam kasus lain, pengujian dan pengembangan aplikasi bisa terjadi kerusakan, bukan hanya pada aplikasi namun juga sistem operasi dibawahnya. Untuk melakukan perbaikan diperlukan proses instalasi ulang yang akhirnya mengurangi produktivitas. Dengan virtualisasi, hanya aplikasi dan mesin *virtual* yang berjalan yang akan terkena dampaknya.

- **Private Cloud Computing**

*Cloud computing* adalah pemanfaatan internet sebagai tempat yang menyediakan layanan teknologi yang dibutuhkan, dan *private cloud* adalah lingkungan dimana layanan *cloud* tersedia hanya dalam satu *enterprise*. Virtualisasi adalah komponen utama dari *Cloud Computing* karena ia menyediakan perubahan kebutuhan sumber daya pengguna relatif lebih cepat.

- **Kualitas Layanan**

Kualitas pada layanan yang diberikan suatu perusahaan menjadi hal yang penting untuk diperhatikan. Bagaimana perusahaan tersebut menjaga aplikasi dan

infrastruktur dibawahnya tetap tersedia dan berjalan dengan baik. Virtualisasi bisa meningkatkan kualitas layanan karena ia menghilangkan ketergantungan pada perangkat keras. Dengan memvirtualisasi sistem, kita bisa dengan cepat merespon kegagalan apapun, baik pada perangkat keras, jaringan, bahkan perangkat lunak virtualisasi itu sendiri. Perangkat lunak virtualisasi dapat mencegah kegagalan dengan memindahkan beban kerja dari sistem, seperti memori, *disk*, dll., yang menunjukkan gejala akan masalah.

- **Failover**

Perangkat lunak virtualisasi secara terus menerus memonitor setiap status mesin *virtual*. Oleh karena itu, ia telah dikonfigurasi untuk memulai mesin *virtual* baru apabila ia mengetahui mesin *virtual* yang sebelumnya berjalan tidak lagi terlihat. Karena perangkat lunak ini harus memulai mesin *virtual* dengan melihat pada *image*-nya saja, maka waktu ketidakterediaan dari mesin *virtual* tersebut bisa diminimalkan hingga beberapa detik atau menit saja. Ini merupakan perubahan lebih baik dari beberapa jam dan bahkan hari yang diperlukan pada sistem yang tidak divirtualisasi.

- **High Availability (HA)**

Ketersediaan tinggi (HA) merupakan pengembangan konsep dari *failover* pada *server* yang memiliki tambahan perangkat keras. Pada saat mesin *virtual* mengalami kegagalan, dibanding dengan memulai pada perangkat keras yang sama, ia akan memulainya pada perangkat keras yang berbeda. Ini akan mengatasi kegagalan mesin *virtual* karena perangkat keras. HA mengandalkan kemampuan perangkat lunak virtualisasi yang dapat mengkoordinasi beberapa mesin. Perangkat lunak yang mengkoordinasi tersebut secara terus-menerus memonitor *hypervisor* dan mesin-mesin *virtual*nya. Apabila *hypervisor* pada satu mesin tidak lagi merespon, perangkat lunak tersebut akan melakukan pengaturan untuk memulai mesin-mesin *virtual* yang tadinya berada pada perangkat keras yang mengalami kegagalan kepada perangkat keras baru.

Virtualisasi seperti ini lebih kompleks karena bagian dari *state* mesin *virtual*, yang terdiri dari alamat jaringan dan sumber daya *storage*, harus dipindahkan sebagaimana mesin *virtual*nya. Apabila *state* ini tidak ikut dipindah, maka yang terjadi adalah mesin *virtual* yang baru tidak bisa terkoneksi ke jaringan atau tidak bisa mengakses *storage*. Kekurangan dari HA adalah perangkat lunak tidak bisa memindahkan *state* dari memori terakhir kepada mesin baru. Oleh karena itu,

walaupun kita bisa memulai kembali mesin *virtual* kita, namun aplikasi di dalamnya tidak bisa melanjutkan pekerjaannya dari *state* terakhir.

- **Clustering**

*Clustering* didesain untuk memastikan tidak ada data yang hilang ketika terjadi kegagalan perangkat lunak atau perangkat keras. *Clustering* membutuhkan biaya lebih dan infrastruktur yang kompleks karena perlu tambahan perangkat keras. Ini dibuat sebagai sistem yang mencerminkan sistem asli yang selalu tersedia dan siap untuk mengambil alih ketika sistem utama mengalami kegagalan. Perangkat lunak virtualisasi mengkoordinasi dua mesin *virtual* yang berjalan pada mesin berlainan. Mesin *virtual* ini identik dalam hal sistem operasi dan konfigurasi aplikasi, tetapi berbeda dalam hal detail koneksi jaringan dan perangkat keras lokal. Satu mesin *virtual* bertugas sebagai *server* utama yang menjalankan aplikasi sementara mesin *virtual* kedua menyediakan kapasitas cadangan ketika *server* utama mengalami kegagalan. Perangkat lunak virtualisasi secara terus-menerus berkomunikasi dengan mesin *virtual cluster* untuk memastikan bahwa mereka masih bekerja. Ini biasa disebut sebagai *heartbeat*, menandakan kelangsungan mesin *virtual*. *Server* utama juga secara terus-menerus mengirimkan setiap perubahan kepada *server* cadangan sehingga *state* mesin *virtual* merefleksikan mesin *virtual* lainnya. Pada saat terjadi kegagalan pada *server* utama, perangkat lunak mendeteksi ketidakterdapatannya dan memindahkan aplikasi kepada *server* cadangan. Pengguna mungkin melihat ada sedikit keterlambatan pada respon aplikasi, namun aplikasi tidak ada perubahan dan terus berjalan karena *state* dari mesin *virtual* telah disinkronisasi. Proses pemindahan aplikasi secara *real time* ini disebut sebagai *live migration*.

- **Fleksibilitas Operasional TI**

Infrastruktur teknologi informasi diharapkan bisa fleksibel menghadapi perubahan operasi dan kondisi, ia harus bisa ditambah pada saat dibutuhkan atau dikurangi agar dapat berjalan secara efisien. Dengan adanya redundansi pada virtualisasi kita bisa mendapatkan fleksibilitas ini. Miller (2012) menyebutkan salah satu fleksibilitas bisa kita dapatkan pada saat melakukan perawatan atau peningkatan perangkat keras dengan cara memindahkan mesin *virtual* kepada *server* lain sehingga kita bisa melakukan perubahan pada perangkat keras lama. Terdapat tiga tipe proses pemindahan atau migrasi pada teknologi virtualisasi. Setiap

tipe migrasi dibedakan dengan jumlah waktu yang dibutuhkan ketika beban kerja tidak tersedia dan pengiriman *state* aplikasi pada saat proses migrasi.

Tabel 1: Tipe migrasi

<b>Tipe</b>	<b>Downtime</b>	<b>Transfer state</b>
<b>Cold</b>	Beberapa menit hingga jam	Tidak
<b>Warm</b>	Puluhan detik	Ya
<b>Live</b>	Kurang dari 1 detik	Ya

Migrasi *cold* memerlukan proses pemutusan antara *host* asal dengan aplikasi. Kemudian memindahkan data dari *storage* asal ke *storage* baru atau rekonfigurasi *shared storage*. Terakhir memulai kembali *host* baru dan aplikasi yang telah dimigrasi. Aplikasi tidak bisa meneruskan pekerjaan dari *state* terakhir. Sementara pada migrasi *warm* tidak ada proses pemutusan dan mulai kembali. Ia akan menghentikan sementara mesin *virtual* dari sistem asal. Kemudian menyalin mesin *virtual* dan *image memory* dari sistem awal ke sistem baru. Proses dari mesin *virtual* akan dilanjutkan dan *image memory* asal akan dihapus. Migrasi *live* membutuhkan waktu yang lebih sedikit dari migrasi *warm*. Ia akan terlebih dahulu menyalin mesin *virtual* dan *image memory* ke sistem lain. Pada saat dibutuhkan proses migrasi ia hanya perlu menghentikan sementara aplikasi dan memindahkan data terakhir ke mesin *virtual* baru. Kemudian ia akan menyerahkan kontrol mesin *virtual* ke sistem baru.

- **Load Balancing**

*Load balancing* membuat penggunaan sumber daya mesin menjadi lebih efektif. Dibandingkan mesin *virtual* cadangan hanya diam menerima pembaruan dari *server* utama dan tidak menjalankan aplikasi, dengan *load balancing* ia akan menjalankan sebagian dari beban kerja, memastikan sumber dayanya digunakan.

- **Server Pooling**

Perangkat lunak pada *server pooling* mengendalikan sekumpulan (*pool*) server *virtual*. Dibandingkan dengan apabila kita memasang mesin *virtual* pada server tertentu, perangkat lunak akan memperhitungkan pada *server* fisik mana mesin *virtual* itu baik untuk dijalankan. Selain itu, ia akan memantau setiap mesin

*virtual* dan *server* untuk menentukan alokasi sumber daya. Apabila mesin *virtual* perlu direlokasi pada *server* yang lebih baik, perangkat lunak akan secara otomatis melakukan migrasi. Apabila keseluruhan sistem mencapai utilisasi maksimum, kita bisa menambahkan *server* baru dan kemudian perangkat lunak akan menyeimbangkan beban kerja untuk membuat sistem yang efektif. Kita tidak bisa mengetahui secara spesifik dimana mesin *virtual* berjalan, sehingga *storage* harus tersedia dalam jaringan sehingga mesin *virtual* dari berbagai *server* dapat mengaksesnya.

- **Pemulihan Bencana**

Kegagalan seperti kegagalan mesin *virtual* atau kegagalan bagian dari perangkat keras bisa kita hindari dengan cara-cara yang telah disebutkan sebelumnya, namun kita tidak bisa menghindari situasi bencana. Pemulihan bencana terjadi ketika seluruh *Data Center* mengalami kegagalan sementara atau permanen. Secara umum kita harus menyediakan *Data Center* cadangan, kemampuan *back up* sistem operasi dan aplikasi, dan rencana pemulihan kegagalan yang matang bagi perusahaan. Pada sistem konvensional, waktu yang diperlukan untuk melakukan *back up* sistem cukup lama sehingga tidak efektif dan menggunakan *Data Center* dan infrastruktur cadangan membutuhkan biaya besar sehingga tidak efisien. Virtualisasi adalah teknologi yang baik untuk mengatasi permasalahan ini karena *image* dari mesin *virtual* bisa diambil dalam berkas dan bisa dijalankan oleh *hypervisor*. Artinya virtualisasi membutuhkan *server* yang lebih sedikit, dan dengan perangkat lunak virtualisasi yang sudah ada pada sistem cadangan, kita bisa lebih cepat menjalankan kembali mesin *virtual*.

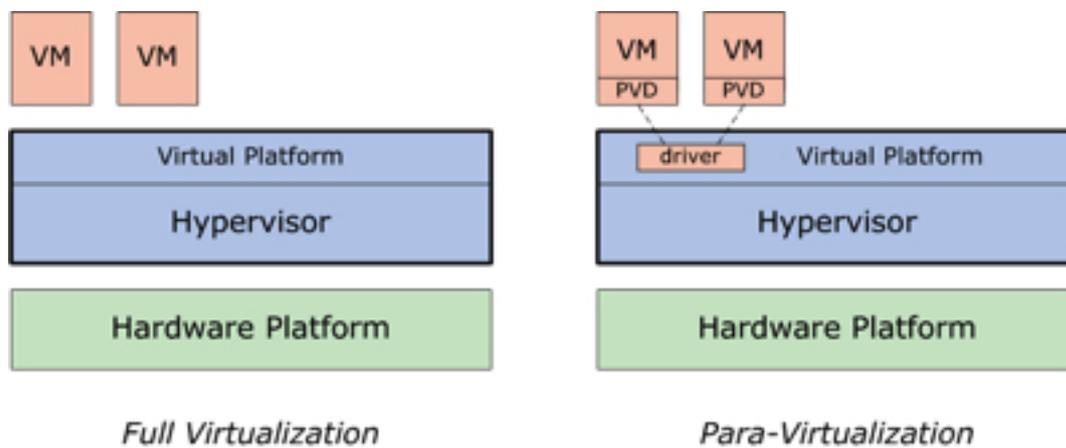
## 5.2 Teknologi Virtualisasi Server

Secara sederhana, virtualisasi *server* adalah cara untuk menjalankan beberapa sistem operasi pada satu mesin, apakah mesin *virtual* tersebut memiliki sistem operasi yang berbeda, memiliki beberapa versi dari sistem operasi yang sama, dan sebagainya. Cheng (2014) menyebutkan untuk membedakan tipe-tipe virtualisasi *server* dapat dilakukan dengan membandingkan cara Control Processing Unit (CPU) mengeksekusi kode. Pada CPU terdapat urutan *ring* proteksi dari 0 ke 3 dibawah arsitektur x86 yang berfungsi untuk memproteksi perangkat keras dengan cara pembatasan akses. Normalnya, *Ring 0* (disebut juga mode *supervisor*) digunakan pada sistem operasi untuk memonitor dan mengontrol sumber daya

sistem. *Ring 3* (disebut juga mode pengguna) digunakan pada aplikasi pengguna, dan apabila ingin mengakses perangkat keras diperlukan panggilan sistem ke mode *supervisor*.

### Full Virtualization

Pada virtualisasi ini, VMM diletakkan pada *Ring 0* sementara sistem operasi *guest* diletakkan pada *Ring 1*. Walaupun begitu, beberapa panggilan sistem hanya bisa dilakukan pada *Ring 0*, sehingga proses yang disebut *binary translation* dilakukan untuk menerjemahkan panggilan sistem tersebut. Proses ini menyebabkan performa berkurang. Pada virtualisasi ini, sistem operasi *guest* tidak mengetahui dia sedang divirtualisasikan, sehingga tidak membutuhkan modifikasi *kernel*. Gambar (3) menunjukkan struktur sederhana dari tipe virtualisasi ini.



Gambar 3: Perbandingan sistem tradisional dan full virtualization (Cheng, 2014)

### Para-virtualization

*Para-virtualization* disebut juga virtualisasi pada level sistem operasi. Serupa dengan virtualisasi *full*, tetapi *custom drivers* diinstal pada sistem operasi *guest* agar dapat mengakses CPU. Hal ini menyebabkan performa dari sistem operasi *guest* mendekati pada mesin fisik karena proses penerjemahan tidak diperlukan. Walaupun begitu dibutuhkan modifikasi *kernel* sehingga *guest* tidak dapat menjalankan sistem operasi yang berbeda dari sistem operasi *host*. Gambar (3.4) menunjukkan struktur dari virtualisasi ini.

## Hardware-Assisted Virtualization

Pabrikan CPU memperkenalkan fungsionalitas baru untuk platform virtualisasi yang terdapat pada Intel VT-x dan AMD-V. *Ring 0* sampai 3 dikategorikan kepada mode *non-root*, dan level baru yaitu -1 diperkenalkan sebagai mode *root*. Sistem operasi *guest* kini diinstal pada *Ring 0*, yang berarti ia bisa mengakses perangkat keras secara langsung dan tidak diperlukan modifikasi *kernel*. Diagram berikut memperlihatkan struktur virtualisasi ini:

Geng (2015) menyebutkan istilah *data center* disebut juga dengan berbagai istilah lain seperti *data warehouse*, *server farm*, dan seterusnya. Istilah *data center* tersebut dapat didefinisikan sebagai peralatan elektronik yang memiliki fungsi pemrosesan data, penyimpanan data, dan komunikasi. Fungsi tersebut, yaitu pemrosesan data tersedia pada *server*, penyimpanan data pada perangkat *storage*, dan komunikasi pada perangkat jaringan, yang secara bersama-sama memproses, menyimpan, dan mentransmisi informasi digital.

Konsolidasi *Data Center* diperlukan untuk mendapatkan layanan yang efektif dan efisien. Konsolidasi dapat dilakukan dengan melakukan sentralisasi dan virtualisasi (Hausman dan Cook, 2011). Sentralisasi *Data Center* dilakukan untuk merestrukturisasi penyimpanan sumber daya informasi yang terpisah. Selain itu akan didapatkan penyederhanaan administrasi dan pengurangan biaya dan kompleksitas. Walaupun begitu, tetap diperhatikan akses konektivitas dan otentikasi pengguna secara *remote* sebagai pemulihan apabila terjadi bencana dan mendukung keberlanjutan operasi. Sementara itu, virtualisasi dapat mengurangi kompleksitas dengan mengurangi *server* fisik yang harus dikelola. *Server virtual* bisa terdapat beberapa mesin *virtual* sesuai dengan kapabilitas pemrosesan, memori, dan *storage host*. Sistem *virtual* bisa terdiri dari sistem operasi yang sama maupun berbeda sesuai dengan teknologi virtualisasi yang digunakan. Karena *server virtual* pada esensinya adalah sebuah berkas pada *host*, ia dapat dipindahkan dari satu *host* fisik ke *host* lain. Hal ini menjaga layanan penting untuk tetap berjalan pada saat terjadi kegagalan perangkat keras. *Data Center* yang dibuat dengan teknologi virtualisasi dengan mesin *virtual* sebagai elemen pemrosesan dasar disebut *Virtual Data Center* (Wei, dkk., 2011)

## 5.3 Proxmox VE

Proxmox VE adalah sebuah proyek *open source*, dikembangkan dan didukung oleh Proxmox Server Solution GmbH. Proxmox VE menggunakan *kernel* Linux dan berbasis Debian GNU/Linux Distribution. Proxmox VE v3.x dilisensi di bawah GNU Affero General Public License, versi 3 (GNU AGPL, v3). Akses manajemen mesin *virtual* dapat dilakukan lewat *web UI*: tampilan konsol terintegrasi dengan mesin *virtual*, integrasi dan pengelolaan Proxmox VE Cluster, teknologi AJAX untuk update sumber daya dinamis, dan akses yang aman ke semua mesin *virtual* melalui enkripsi SSL. Pengguna bisa menginstal sistem operasi berbeda pada sebuah mesin tunggal atau pada *cluster* dari sekumpulan mesin. Proxmox VE menyediakan dua solusi untuk virtualisasi: mesin *virtual* berbasis *kernel* yaitu KVM dan OpenVZ yang berbasis kontainer sebagai alternatif.

### 1. KVM (*Kernel-based Virtual Machine*)

KVM merupakan virtualisasi jenis *hardware-assisted* dengan *kernel* Linux yang dimodifikasi dibangun dengan modul KVM. KVM mengintegrasikan kemampuan *hypervisor* kepada *kernel* Linux *host* dan dapat menyederhanakan manajemen serta meningkatkan kinerja dalam lingkungan *virtual*. KVM dapat mendukung berbagai jenis platform. KVM sendiri tidak melakukan emulasi atau virtualisasi. Sebagai gantinya QEMU dipilih sebagai *emulator* berbasis perangkat lunak untuk mensimulasikan perangkat keras untuk lingkungan virtualisasi. Sebuah mesin virtualisasi KVM biasanya terdiri dari komponen-komponen berikut: *device driver* untuk mengelola perangkat keras virtualisasi, komponen *user-space* untuk mengemulasi perangkat keras, dan model I/O langsung berasal dari QEMU. *Overhead* muncul pada beberapa proses menggunakan QEMU. Salah satu pilihan adalah menggunakan VirtIO yang merupakan *para-virtualization* pada I/O untuk membuat *buffer* pada sistem *guest* dan QEMU yang akan mempercepat performa I/O dan mengurangi *overhead*.

Pada KVM terdapat beberapa pilihan antarmuka jaringan, yaitu mode *bridged*, NAT, dan tanpa perangkat jaringan. Pada mode *bridged* membuat pengguna memilih *network adapter* selain standarnya. Mode ini membutuhkan DHCP *server* untuk mendapatkan IP atau harus dikonfigurasi secara manual di dalam *vm*.

### 2. Container Virtualization (*OpenVZ*)

Teknologi yang dipilih untuk *server* Linux karena relatif lebih cepat. OpenVZ adalah virtualisasi pada level sistem operasi berbasis *kernel* dan sistem operasi *host* GNU/Linux. OpenVZ memungkinkan sebuah *server* fisik untuk menjalankan beberapa sistem operasi terisolasi (*guest*) yang dikenal sebagai kontainer, *Virtual Private Server* (VPS), atau *Virtual Environments* (VEs). Teknologi OpenVZ memiliki *kernel* Linux yang dibagi dengan *server host* dan menciptakan beberapa kontainer terisolasi menjadi *server* fisik tunggal. Oleh karena itu tidak ada *overhead* pada kontainer karena bisa mengakses perangkat keras secara langsung. Hal ini memungkinkan memori *server host* yang digunakan lebih sedikit dan memastikan bahwa aplikasi tidak bertabrakan. Setiap kontainer berjalan dan mengeksekusi persis seperti *server standalone*. OpenVZ terbatas dalam hal ia mengharuskan baik OS *host* maupun *guest* adalah berbasis Linux. Karena semua kontainer membagi sistem *kernel* dari sistem operasi *host*, ada kemungkinan masalah apabila saat menjalankan sistem dan sistem *guest* mengakses sumber daya CPU yang sama. Selain itu, OpenVZ menyimpan *berkas* kontainer sebagai *berkas* normal pada *host*, oleh karena itu tidak direkomendasikan menggunakan OpenVZ apabila ada data penting dan rahasia yang disimpan pada mesin *virtual*. Sebuah kontainer dapat melakukan proses *reboot* sendiri dan memiliki akses *root*, alamat IP yang unik, memori, proses, *berkas*, aplikasi, *system library* dan *berkas* konfigurasi.

Ada dua buah pilihan konfigurasi antarmuka jaringan pada OpenVZ. Pilihan pertama adalah mode *routed*. Pada mode ini dipakai antarmuka jaringan standar Proxmox, dan hanya menspesifikasikan alamat IP. Mode ini hanya dipakai jika *guest* sistem memiliki subnet yang sama dengan *host* sistem. Pilihan kedua adalah mode *bridged* yang sama pada KVM.

Ada beberapa fitur dan kelebihan dari Proxmox VE. Cheng (2014) menyebutkan fitur utama dari Proxmox VE bisa diringkas menjadi berikut:

- **Open Source:** Perangkat lunak ini secara penuh sumber terbuka di bawah General Public License, versi 3 (GNU AGPL, v3), yang berarti kita bisa secara bebas melihat, mengubah, atau menghapus kode sumber, dan mendistribusikan versi kita sendiri selama kita tetap tunduk pada lisensi.
- **Live Migration:** Kita bisa memindahkan mesin *virtual* yang berjalan dari *server* fisik satu ke *server* lainnya tanpa ada waktu henti.

- **High Availability:** Pada Proxmox mode *HA Cluster*. Ketika satu node gagal, mesin *virtual* di dalamnya akan dipindahkan ke *node* yang masih bisa berjalan untuk memastikan gangguan layanan minimal.
- **Bridged Networking:** Proxmox VE memberikan pengguna untuk bisa membuat jaringan *private* antar mesin *virtual*. *Bridge* dipasangkan pada kartu jaringan fisik.
- **Flexible storage:** Berbagai macam pilihan storage tersedia, termasuk teknologi storage lokal maupun berbasis jaringan seperti LVM, iSCSI, NFS, Gluster filesystem, dan CEPH filesystem.
- **OS template:** Proxmox VE membolehkan pengguna membuat template sistem operasinya sendiri untuk pengembangan lebih lanjut, atau bisa juga mengunduh template melalui internet dan mengimpornya ke sistem.
- **Scheduled backup:** Antarmuka pengguna disediakan kepada pengguna sehingga kita bisa menyiapkan strategi *backup* kita sendiri. Berkas *backup* yang sudah dikonfigurasi bisa disimpan secara lokal atau berbagai *storage* pilihan yang didukung.
- **Command-Line (CLI) tool:** Selain melalui *Web UI*, Proxmox VE juga menyediakan berbagai manajemen CLI yang membuat pengguna bisa mengakses kontainer mesin *virtual*, mengelola sumber daya, dsb.

Mesin *virtual* membutuhkan jenis sumber daya yang tepat agar dapat meningkatkan performa dan memenuhi kebutuhan pengguna. Sumber daya tersebut diantaranya adalah *hard disk*, memori, dan jaringan. Proxmox menyediakan beberapa tipe sumber daya ini. Cheng (2014) menyebutkan pada Proxmox VE terdapat dua teknologi *hard drive* mesin *virtual* baik OpenVZ maupun KVM, yaitu:

- a. **RAW:** Pada format berkas raw, ruang *disk* dialokasikan ketika mesin *virtual* dibuat dan digunakan sesuai besar yang telah dispesifikasikan pengguna. Walaupun pengguna mungkin tidak memakai seluruh ruang yang telah disediakan, namun secara umum hal ini akan memberikan performa *hard disk* yang lebih baik.
- b. **QCOW2:** Dikembangkan dari format QCOW, yang menyediakan kemampuan *thin-provisioning* untuk *hard drive* yang digunakan oleh QEMU. *Thin-provisioning* berarti ketika ruang *disk* dispesifikasikan dengan besar tertentu oleh pengguna, berkas *image* yang lebih kecil dari ruang tersebut akan

dibuat. Ukuran besar tadi akan menjadi besar maksimum *image disk*. Besar berkas kemudian akan berkembang sesuai penggunaan di dalam sistem *guest*.

### 5.3.1 Perbandingan Perangkat Lunak Virtualisasi

Terdapat berbagai macam perangkat lunak virtualisasi, baik berbayar maupun *open source*. Berikut beberapa perangkat lunak virtualisasi beserta fitur-fiturnya. Secara umum perbedaan tiap-tiap perangkat lunak ditunjukkan pada tabel (3.2).

Tabel 2: Perbandingan Perangkat Lunak Virtualisasi

Platform Virtualisasi	Proxmox VE	Xen Server	VMware ESX/ESXi	Hyper-V Server
Lisensi	GNU GPL v3	GNU GPL v2	Proprietary	Gratis
Open Source	Ya	Ya	Tidak	Tidak
Base OS	Linux	Linux	Vmkernel	Windows
Console OS	Tidak	Ya	Ya	Tidak
Management tools	Web UI	Xen Center	vSphere Client vCenter	Hyper-V Manager
Host HA	Ya	Ya	Ya (vCenter)	Ya
Guest HA	Ya	Ya (XenMotion)	Ya (vMotion)	Ya (Live Migration)
Dukungan storage	LVM, NFS, iSCSI, RBD, NAS, SAS	NFS, iSCSI, Fibre Channel, NAS, SAS, CIFS	Fibre Channel, iSCSI, NAS, SAS	SMB, iSCSI, Fibre Channel, NAS, SAS
Dukungan teknologi virtualisasi	FV (KVM) PV (OpenVZ)	PV HV	FV PV HV	PV HV
Model lisensi	Per CPU	Per CPU	Per CPU	Per VM & host

1. **Proxmox VE:** Proxmox VE adalah *hypervisor* yang bersifat *open source* berbasis GNU/Linux (berbasis Debian) dengan *kernel* berbasis RHEL dan dikeluarkan dibawah GNU AGPL v3. Keunikan Proxmox adalah ia menyediakan manajemen pusat berbasis web. Teknologi yang dipakai adalah Open Virtuozzo (OpenVZ) dan Mesin *virtual* berbasis *kernel* (*Kernel-based Virtual Machine/KVM*).
2. **XenServer:** Sebuah *hypervisor native* berbasis GNU/Linux dikembangkan oleh Xen Project dan dikeluarkan dibawah GNU AGPL v2 sebagai *open*

*source*. Untuk XenServer, konsep *domain* dipakai untuk semua mesin *virtual*, dan *domain* yang paling istimewa (*domain* yang memperkenankan akses langsung ke perangkat keras)– **dom0**, digunakan oleh *hypervisor* untuk mengatur mesin *virtual* **domU** lain. XenServer mendukung *para-virtualization*, yang membuat pengguna dapat menjalankan mesin *virtual* pada CPU tanpa dukungan virtualisasi, misalnya yaitu Intel VT-x atau AMD-V tidak dibutuhkan.

3. **Vmware ESX/ESXi:** Sebuah *hypervisor bare-metal* dikembangkan oleh Vmware berbasis *kernel* yang disesuaikan bernama *vmkernel*. Perbedaan antara Vmware ESX dan ESXi yaitu ESXi adalah versi gratis dari ESX dengan beberapa pembatasan sumber daya. ESX mempunyai daftar kompatibilitas mencakup beberapa *driver* untuk kartu jaringan dan kartu SCSI. Selain *para-virtualization* dan virtualisasi *hardware-assisted*, ESX juga menyediakan virtualisasi *full*. Terdapat dua perangkat manajemen yang tersedia: vSphere client dan vCenter server. Vsphere client cukup untuk administrasi operasi dari sebuah ESX sementara vCenter server mampu untuk mengelola beberapa ESX, termasuk konfigurasi fitur seperti *high availability* dan *live migration*.
4. **Hyper-V server:** Platform virtualisasi berpemilik (*proprietary*) diproduksi oleh Microsoft dan berjalan dibawah platform Windows, yaitu Windows Server 2008 dan di atasnya. Hyper-V menyediakan pilihan migrasi yang lebih baik karena tidak hanya menyediakan *live migration*, tetapi juga pergerakan tidak terbatas *guest* antar *host*. Apabila kita mengaktifkan *Active Directory domain*, Hyper-V dapat menyediakan replika dari mesin *virtual*, yang membuat pengguna dapat menyalin mesin *virtual* spesifik dari *site* sumber ke *site* target secara asinkronus melalui WAN atau *secure VPN*.

### 5.3.2 Proxmox VE Cluster

Proxmox VE 2.x dan semua versi di atasnya menyediakan manajemen pusat pada *web UI* dari beberapa *server*. Pada Proxmox VE Cluster terdapat sistem berkas Proxmox Cluster (*pmxcfs*) yang merupakan sistem berkas berbasis basisdata untuk menyimpan berkas konfigurasi, yang termasuk konfigurasi *cluster* dan konfigurasi mesin *virtual*, yang akan direplikasi secara waktu nyata pada setiap *node* menggunakan *Corosync Cluster Engine*. Proxmox VE Cluster tidak memerlukan *node* khusus untuk manajemen. Semua *node* mempunyai kapabilitas manajemen sehingga mengeliminasi titik kegagalan (*single point of failure*) (Kovari dan Dukan,

2012). Perbedaan dasar Proxmox VE dengan mesin tunggal dan *cluster* dapat dilihat pada tabel (3.3).

Tabel 3: Perbedaan Proxmox VE Mesin Tunggal dan Cluster

Produk	Proxmox VE	Proxmox VE Cluster
Jumlah <i>node</i>	1	2 (minimal)
<i>Quorum</i>	Tidak	Ya
<i>Storage</i>	Lokal	Berbagi ( <i>Shared</i> )
Kemampuan HA	Tidak	Ya

Pada tabel (3.3) disebutkan Proxmox VE Cluster membutuhkan *shared storage* untuk menyimpan data mesin *virtual* yang akan dideskripsikan pada subbab berikutnya. Proxmox VE Cluster mempunyai fungsi *live migration* yang berbeda antara KVM dan OpenVZ.

KVM menggunakan teknik *pre-copy memory migration* untuk proses migrasi. Terdapat dua fase pada teknik ini, fase pertama adalah fase pemanasan. Pada fase ini, Proxmox mencoba menyalin semua halaman memori dari *server* asal ke *server* tujuan ketika mesin *virtual* masih berjalan. Ketika ada perubahan dari halaman memori asal, halaman memori yang berkorespondensi di mesin tujuan ditandai kotor (*dirty*). Halaman memori yang ditandai kotor akan disalin ulang hingga memori yang harus disalin lebih besar dari memori yang ditandai kotor. Fase berikutnya yaitu berhenti-dan-salin, mesin *virtual* yang akan dimigrasi akan diberhentikan dan memori kotor sisa yang ada akan disalin. Fase terakhir tersebut adalah keseluruhan *downtime* dari mesin *virtual*, dan kemudian pekerjaan mesin *virtual* akan dilanjutkan pada mesin tersebut. Teknik ini dapat memulihkan vm apabila terjadi kesalahan migrasi.

Pada OpenVZ menggunakan teknik *post-copy memory migration*. Langkah pertama migrasi adalah, data di dalam *container* OpenVZ disalin kepada *server* tujuan. Langkah selanjutnya vm yang berada pada *node* asal ditangguhkan, sebagian sumber daya sistem, termasuk CPU, *register*, *non-paging memory*, akan dikirimkan kepada *server* tujuan. Proses ini disebut *pre-paging*. Langkah terakhir yaitu vm akan dilanjutkan pada *server* tujuan, sementara *node* asal akan mengirimkan *memory page* yang tersisa ke *server* tujuan. Ketika vm baru

mengakses *memory page* yang belum terkirim, terjadi kegagalan *page*. Untuk mengatasi masalah ini vm akan diarahkan ulang ke *node* asal. Pengarahan ulang ini dapat mengurangi kinerja sistem sehingga membutuhkan waktu migrasi yang lebih lama.

### 5.3.3 Data Storage

Hausman dan Cook (2011) menyebutkan ada beberapa konfigurasi *data storage*, diantaranya adalah *Direct attached storage (DAS)*. DAS adalah perangkat *storage* yang menyambung langsung dengan komputer, baik secara internal maupun eksternal. Permasalahan DAS ada pada *data sharing* dan ketika dibutuhkan pemulihan bencana. Selain itu layanan berkas memakai sumber daya *server* yang sama dengan layanan lain yang menyebabkan penurunan kinerja.

*Shared storage* adalah *data storage* yang terhubung dengan jaringan dan dapat diakses oleh komputer atau *server* dalam jaringan tersebut. Karena terhubung dengan jaringan muncul istilah *storage area network (SAN)* dan *network-attached storage (NAS)*. *Network-Attached Storage (NAS)* adalah *storage* pada level berkas. NAS berjalan pada sistem operasi untuk menyediakan *file system* dan menggunakan DAS. Kelebihan NAS dibanding DAS adalah NAS dapat melakukan *backup*. *Storage* ini dibagikan melalui protokol SMB atau NFS. Karena ia berada pada level berkas, pengaturan *permission* berbeda bisa diaplikasikan kepada berkas dan direktori. *Storage* kemudian akan dimasukkan (*mount*) melalui protokol.

*Storage Area Network (SAN)* adalah *storage* pada level *block*. SAN menggunakan protokol dan jaringan yang didedikasikan sehingga didapatkan komunikasi antara *storage* dan *server* yang lebih cepat. Protokol yang biasa digunakan untuk memasang (*mount*) perangkat SAN antara lain iSCSI dan *fibre channel*. Ketika *storage* dimasukkan, ia akan terlihat sebagai *disk* lokal di dalam sistem, sehingga sistem operasi bisa mengelola sistem berkas dari *disk*.

Terlepas dari ketiga jenis *storage* yang telah disebutkan, Proxmox mendukung beberapa sistem berkas terdistribusi. Proxmox menyediakan pilhan *storage* berikut:

- directory
- logical volume manager (LVM)
- iSCSI target

## Pusat Data untuk Pemerintahan @ 2016

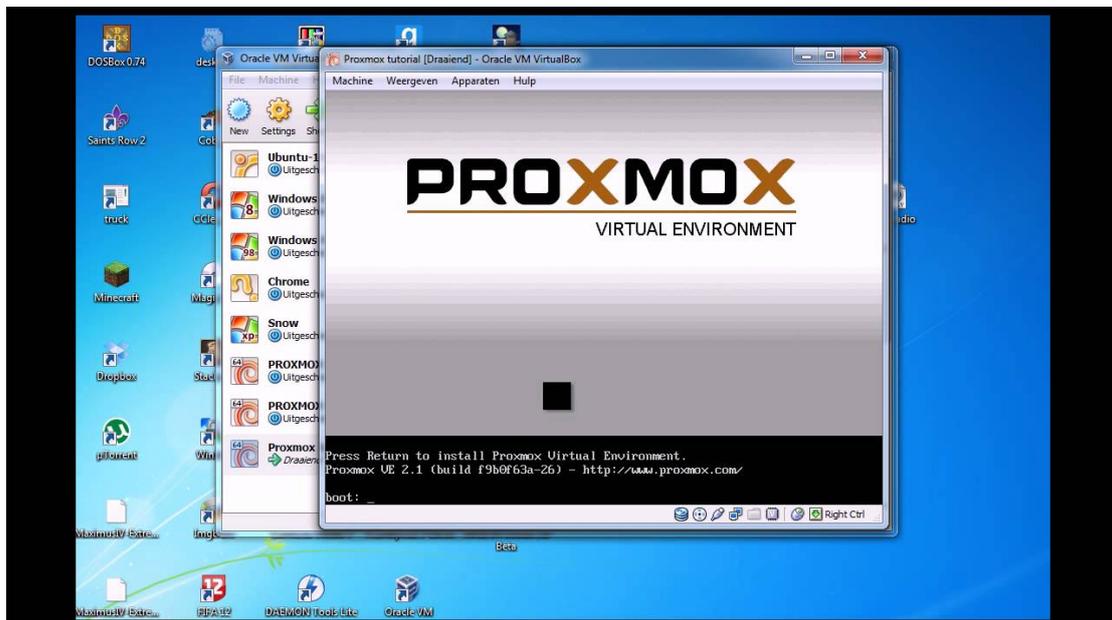
- NFS share
- Gluster filesystem
- Ceph RBD

## Bab 6. Virtual Data Center

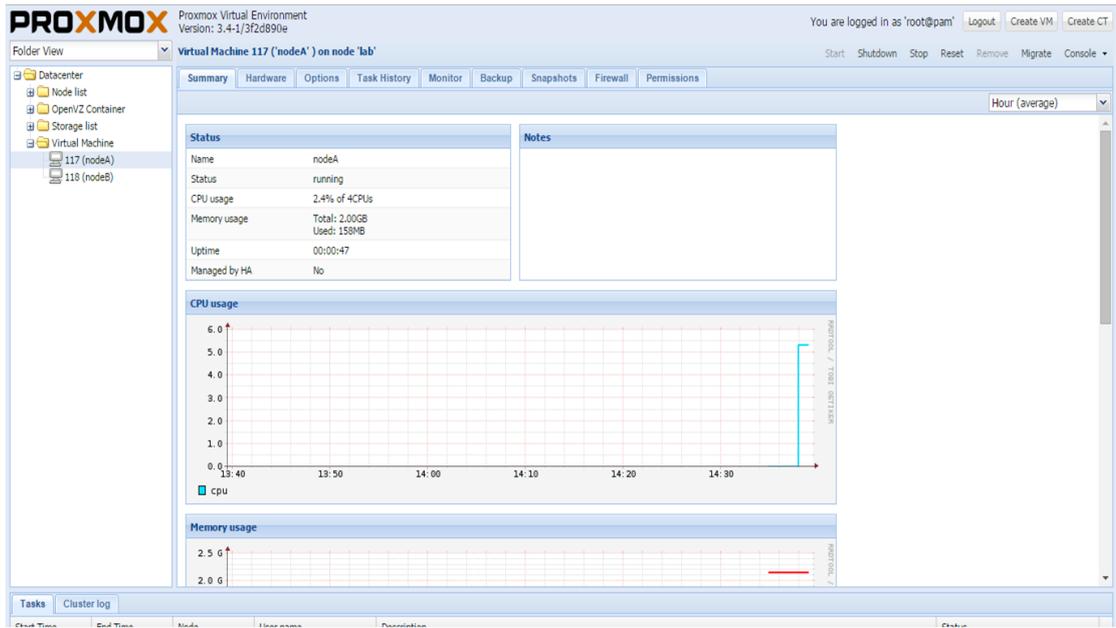
### 6.1 Implementasi Virtual Data Center

Bagian ini menjelaskan langkah-langkah implementasi *Virtual Data Center* yang merupakan sistem *cluster* dengan *shared storage*. Pada mesin *server host* dilakukan instalasi *hypervisor* Proxmox VE *bare metal*, sehingga tidak memerlukan sistem operasi di bawahnya. Dua buah *Guest* Proxmox dibuat melalui antarmuka *web host* Proxmox, yaitu *node A* dan *node B*, sesuai dengan rancangan penelitian dan spesifikasi perangkat keras yang telah disebutkan pada subbab 5.1.1. Setelah mesin *virtual* dibuat, proses instalasi dilakukan dengan membuat sistem *cluster* pada *guest hypervisor*. Kemudian dilakukan proses instalasi DRBD sebagai *shared storage*. Proses terakhir membuat *container* OpenVZ dan mesin *virtual* KVM pada antar muka *web guest* Proxmox .

#### Menjalankan Proxmox



## Tampilan Host Poxmox



## Virtual Machine

The screenshot shows the 'Create: Virtual Machine' dialog box in Proxmox VE, specifically the CPU configuration tab. The dialog has several tabs: General, OS, CD/DVD, Hard Disk, CPU, Memory, Network, and Confirm. The CPU tab is active, showing the following configuration:

- Sockets:** 1
- Cores:** 4
- Enable numa:**
- Type:** host
- Total cores:** 4

At the bottom of the dialog, there are 'Back' and 'Next' buttons.

**Create: Virtual Machine** [X]

General OS CD/DVD Hard Disk CPU Memory **Network** Confirm

Bridged mode  
VLAN Tag: no VLAN  
Bridge: vubr0  
Firewall:

NAT mode  
 No network device

Model: VirtIO (paravirtualized)  
MAC address: auto  
Rate limit (MB/s): unlimited  
Multiqueues:  
Disconnect:

Back Next

## Bab 7. Big Data

---

*Big data* memiliki definisi dalam beberapa perspektif yaitu perspektif pengetahuan dan proses. *Big Data* merupakan data yang memiliki volume besar, keragaman dan kompleksitas yang variatif. Hal ini membutuhkan arsitektur, teknis, algoritma dan model analisa baru yang ditujukan untuk mengelola, memanfaatkan dan memunculkan pengetahuan yang ada didalamnya. The McKinsey Global Institute (2011) mendefinisikan *Big Data* sebagai kumpulan data yang memiliki ukuran besar dan melebihi kapasitas perangkat lunak database untuk mengelolanya dan menganalisisnya. *Big Data* muncul dari proses transaksi data, interaksi data dan observasi data yang terus menerus. Data yang dihasilkan akan berukuran besar dari *Megabyte, Gigabytes, Terabytes, Petabytes, Exabyte, Zettabyte, Yottabyte*, dan seterusnya.

Karakteristik *Big Data* juga berkembang seiring dengan perubahan pada lingkungan teknologi, bisnis dan ilmiah (Riasetiawan, 2011). Karakteristik *Big Data* dapat menjadi 4V yang terdiri atas *Volume, Velocity, Variety*, dan *Veracity*. *Volume* masih memiliki definisi yang sama dengan sebelumnya, identic dengan besaran atau jumlah. *Velocity* juga masih memiliki definisi yang hamper sama yaitu kecepatan dalam menghasilkan data. *Variety* focus pada bentuk format jenis dan struktur. Data tidak terstruktur, text, multimedia menjadi sebagian contohnya. *Veracity* menyoroti mengenai ketidakpastian pada aspek data konsistensi, dan kelengkapan data, *ambiguitas, latency, deception*, dan model *approximations*. Pada definisi berikutnya, *Big Data* menjadi 5V dengan adanya penambahan *Value* yaitu nilai sebuah data dikarenakan makna yang dimiliki didalam data. Big data dapat berupa berbagai data. Data sensor, *data log, data storage, web public*, media social, dan aplikasi bisnis memiliki potensi data besar.

Dinamisasi kondisi perekonomian dunia menunjukkan penetrasi pengaruh teknologi informasi yang diterapkan perusahaan dan organisasi. Indikasi dengan adanya fluktuasi nilai tukar mata uang yang sangat cepat dipengaruhi penerapan *e-trading* di pasar uang dan saham. Pola pembayaran elektronik semakin diandalkan dengan kemudahan komunikasi data antar negara dan region yang semakin luas.

Perubahan pola adaptasi dan implementasi teknologi informasi yang semakin dalam bersentuhan dengan bisnis proses organisasi membawa peluang yang potensial dalam pengembangan ide dan inovasi bisnis. Perusahaan semakin dapat

memfokuskan diri pada bisnis inti untuk kelangsungan bisnis dan menggunakan layanan-layanan dukungan untuk menangani operasional dukungan organisasi. Pola layanan berkemabang dari yang bersifat *dedicated* menjadi lebih *sharerable*.

Indonesia memiliki potensi baik sebagai pengguna dan pelaku untuk menempatkan posisinya dalam perekonomian dunia. Penetrasi infrastruktur komunikasi, data dan teknologi sudah dilayani oleh berbagai *provider* telekomunikasi. Pengguna media penyedia data dan informasi, baik berbasis jejaring sosial, *ecommerce*, *personal blog*, dan lainnya sangat tinggi. Hal ini dibuktikan dengan beberapa artikel media yang menempatkan Indonesia sebagai pengguna terbesar kedua untuk jejaring sosial dan empat terbesar untuk media Twiter. Tentu hal ini menunjukkan tingkat signifikansi terutama dari sisi potensi pasar yang terus berkembang. Meskipun belum didukung dengan penelitian ilmiah, dinamisasi perekonomian di Indonesia dan perkembangan pengguna, infrastruktur, dan media teknologi informasi diyakini memiliki peran dan kontribusi, yang kemudian memunculkan istilah *Cloud-nomics*.

*Big Data* dipicu dan ditunjukkan dengan beberapa *market milestone* diatas terjadi karena penyedia layanan-layanan tersebut memanfaatkan teknologi awan (*cloud technology*) sebagai tulang punggung penyelenggaraan layanan. Teknologi awan memungkinkan penyedia layanan untuk memanfaatkan sumber daya eksternal untuk menunjang keberhasilan bisnis yang ingin dicapai.

Potensi teknologi awan di Indonesia pun tidak bisa dipandang sebelah mata. Tahun-tahun terakhir menunjukkan semakin gencarnya dan bermunculan *start up* lokal yang mengusung layanan berbasis teknologi awan ini. Beberapa perusahaan memberanikan diri untuk memanfaatkan layanan yang ditawarkan meskipun masih dalam tahap pengujian dan memberikan keyakinan terhadap kemanfaatan terhadap bisnis intinya.

Perusahaan dan organisasi membutuhkan peta dan inventarisasi potensi teknologi awan di Indonesia. Layanan-layanan yang dapat diidentifikasi akan memberikan kemudahan bagi perusahaan dan organisasi untuk mensinergikan dengan proses bisnis yang dipunyai.

Referensi

- Ahmed, W., 2014, *Mastering Proxmox*, Packt Publishing, Birmingham.
- Arfriandi, A., 2012, Perancangan, Implementasi, dan Analisis Kinerja Virtualisasi Server Menggunakan Proxmox, VMware ESX, dan Openstack, *Tesis*, Fakultas Teknik, Universitas Gadjah Mada, Yogyakarta.
- Budi, S., 2013, Implementasi Cluster pada Web Server Berbasis Cloud Computing, *Skripsi*, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sunan Kalijaga, Yogyakarta.
- Chang, B.R., Tsai, H.F., Chen, C.M., Lin, Z.Y., dan Huang, C.F., 2012, Assessment of Hypervisor and Shared Storage for Cloud Computing Server, *Prosiding International Conference on Innovation in Bio-Inspired Computing and Applications*, Kaohsiung.
- Cheng, S.M.C., 2014, *Proxmox High Availability*, Packt Publishing, Birmingham.
- Fischer, W., dan Mitasch, C., 2006, High Availability Clustering of Virtual Machines – Possibilities and Pitfalls, *Prosiding Linuxtag*, Wiesbaden.
- Garnieri, M., 2010, Desain dan Implementasi Virtualisasi Server di PT Thiess Contractors Indonesia, *Skripsi*, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada, Yogyakarta.
- Geng, H., 2015, *Data Center Handbook*, John Wiley & Sons, New Jersey.
- Golden, B., 2011, *Virtualization for Dummies*, 3rd HP Special Edition, Wiley Publishing, Indianapolis
- Graupner, S., Kotov, V., dan Trinks, H., 2002, Resource-Sharing and Service Deployment in Virtual Data Centers, *Prosiding International Conference on Distributing Computing Systems Workshops*, Vienna.
- Hatta, P., 2014, Analisis dan Implementasi Virtualisasi Data Center di ICT Center FKIP UNS, *Tesis*, Fakultas Teknik, Universitas Gadjah Mada, Yogyakarta.
- Hausman, K.K., dan Cook, S.L., *IT Architecture for Dummies*, Wiley Publishing, Indianapolis.
- Jones, M.T., 2012, Nested Virtualization for the Next-Generation Cloud – An Introduction to Nesting with KVM, <http://www.ibm.com/developerworks/cloud/library/cl-nestedvirtualization/>, diakses tanggal 5 Juni 2015.
- Khanna, G., Beaty, K., Kar, G., dan Kochut, A., Application performance management in virtualized server environments. *Prosiding Network Operations and Management Symposium (NOMS 2006)*, Vancouver.
- Kovari, A., dan Dukan, P., 2012, KVM & OpenVZ Virtualization Based IaaS Open Source Cloud Virtualization Platforms: OpenNode, Proxmox VE, *Prosiding International Symposium on Intelligent Systems and Informatics*, Subotica.
- Malanik, D., dan Jasek, R., 2014, The Performance of the Data-Cluster Based on the CEPH Platform with Geographically Separated Nodes, *Prosiding International*

*Conference on Mathematics and Computers in Sciences and in Industry, Varna.*

Miller, L.C., 2012, *Server Virtualization for Dummies*, Oracle Special Edition, John Wiley & Sons, New Jersey.

Scheffy, C., 2007, *Virtualization for Dummies*, AMD Special Edition, Wiley Publishing, Indianapolis.

Singh, A., Korupolu, M., dan Mohapatra, D., 2008, Server-Storage Virtualization: Integration and Load Balancing in Data Centers, *Prosiding ACM/IEEE Conference on Supercomputing*, Austin