

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1 Data**

Data merupakan sesuatu yang tertulis dari kegiatan operasional sehari-hari atau hasil dari transaksi yang diperoleh suatu organisasi atau perusahaan. Menurut (Bourgeois, 2014) data merupakan bit mentah dan potongan informasi tanpa konteks dan data bisa bersifat kuantitatif atau kualitatif. Data kuantitatif adalah numerik, hasil pengukuran, menghitung.

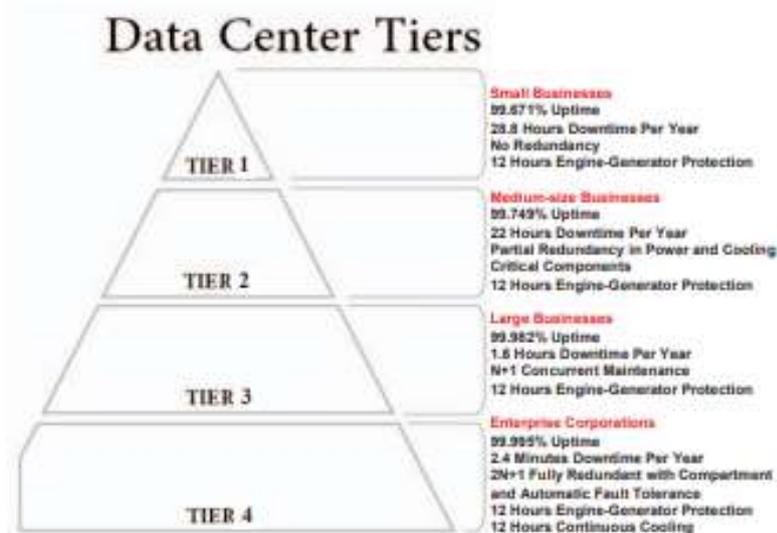
Kumpulan data juga bisa diolah menjadi sebuah informasi dengan mengedepankan proses, pemenuhan kesesuaian data, melakukan komparasi dari beberapa varian dan memberikan saran untuk pengembangan. Kombinasi dari kumpulan data tersebut akan bernilai baik itu untuk perseorangan, organisasi ataupun perusahaan yang memerlukan kumpulan data tersebut (van der Aalst, 2016).

#### **2.2 Data Center**

*Data center* dapat didefinisikan sebagai fasilitas dengan semua sumber daya yang diperlukan untuk penyimpanan dan pemrosesan digital informasi, dan bidang pendukungnya. *Data center* terdiri dari infrastruktur yang diperlukan misalnya distribusi daya, sistem kontrol lingkungan, telekomunikasi, keamanan, perlindungan kebakaran, sistem otomasi dan informasi peralatan teknologi misalnya *server* untuk pemrosesan, data penyimpanan, jaringan / peralatan komunikasi. *Data center* merupakan bagian penting dari perusahaan mengingat

informasi digital yang mereka simpan dalam jumlah besar (Levy & Hallstrom, 2017).

Pada dewasa ini hampir seluruh perusahaan sudah memiliki *data center*, Seluruh kebutuhan dan bisnis sesuai dengan design yang terkini, *environment* pendukung didalam *data center*. Untuk membedakan *data center* tersebut berdasarkan beberapa aspek memiliki klasifikasi ruang *server*, *tier 1*, *tier 2*, *tier 3* maupun *tier 4* yang memiliki infrastruktur *full redundancy* (Wiboonrat, 2016).

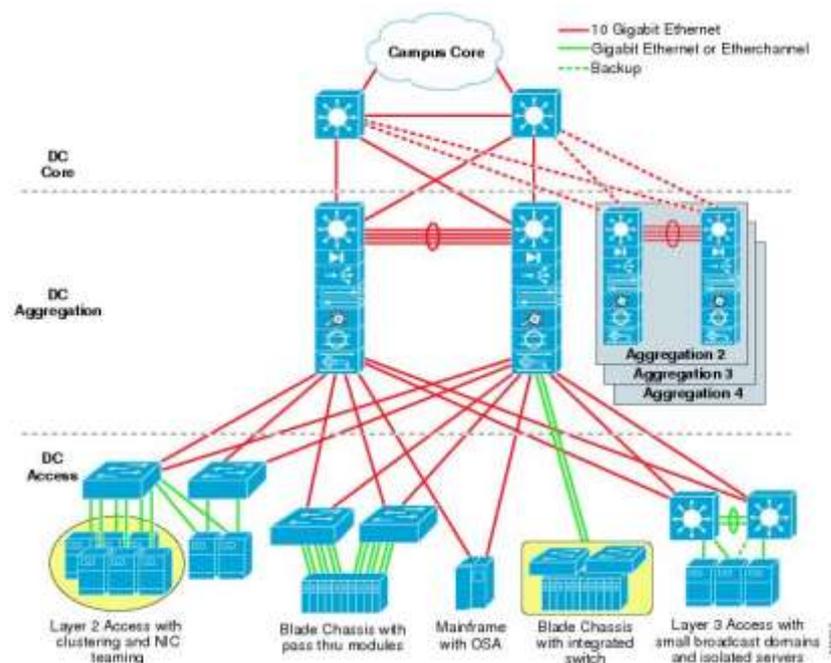


**Gambar 2.1 Data Center Tier**

Merujuk gambar 2.1 diatas, *Data center tier 1* digunakan untuk *small business* pada perusahaan yang tujuannya *data center* sebagai pendukung pekerjaan. Sehingga para *stakeholder* tidak terlalu *concern* terhadap *data center* tersebut, tidak ada *fase* pengembangan hanya untuk berfungsi. Klasifikasi *data center tier 2* para *stakeholder* harus *concern* terhadap aspek *redundancy* walaupun masih hanya sebagian tidak seluruh aspek dilakukan *redundancy* fokus terhadap *power source* dan *cooling system*. Klasifikasi *data center tier 3* untuk perusahaan yang

ukurannya sudah *large business*, dalam aspek ini *stakeholder* memikirkan *aspek redundancy, maintenance, operational* dan *down time*. klasifikasi *data center tier 4* ini merupakan untuk *level enterprise company*. Seluruh perangkat memiliki *backup* dan seluruh system bisa di pindah-pindahkan tanpa menghilangkan data yang ada.

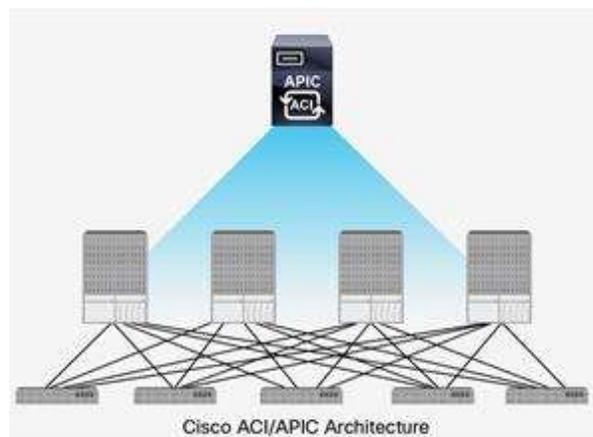
Dalam perkembangan *data center* sekarang sangatlah canggih dan mutakhir. Seperti yang kita ketahui bersama *system data center* yang lama dengan menggunakan teknologi *3 tier layering* pada *segment network* (Sama et al., 2015).



**Gambar 2.2 Three Tier Layer Topology**

Merujuk gambar 2.1 diatas, dalam skema tersebut sudah dipisahkan antara perangkat yang berfungsi sebagai *layer 2* maupun perangkat yang berfungsi sebagai *layer 3*. Sehingga *controlling* perangkat betul-betul menggunakan teknologi *layering*. Dengan berkembangnya teknologi dan kemajuan penelitian ternyata ditemukan teknologi baru yang lebih *agile* untuk pengembangan dan

investasi sehingga tidak fokus terhadap *capex* tetapi mulai labat-laun kearah *opex* perusahaan. Salah satunya adalah *software defined network*. perkembangan teknologi ini diawali oleh perusahaan teknologi informasi terkemuka yaitu *cisco*. Mereka mengembangkan system komunikasi perangkat yang lebih teratur dan tersegmentasi untuk pembagian setiap klasifikasinya (Yu, Yu, Wang, Fan, & Li, 2017).



**Gambar 2.3 Cisco ACI/APIC Architecture**

Merujuk gambar 2.3 diatas, teknologi yang dikembangkan dengan menggunakan *software defined architecture* mengembangkan teknologi *spine* dan *leaf* untuk berkomunikasi. *Spine* sebagai *parent* dari seluruh perangkat yang ada dan *leaf* sebagai jembatan untuk berkomunikasi dengan seluruh perangkat lain baik itu *security*, *server*, *database* maupun *storage*. Seluruh perangkat tersebut di *control* oleh satu *controller* yang disebut dengan *ACI Controller* (Herker, An, Kiess, Beker, & Kirstaedter, 2015).

### **2.3 Disaster Recovery Center**

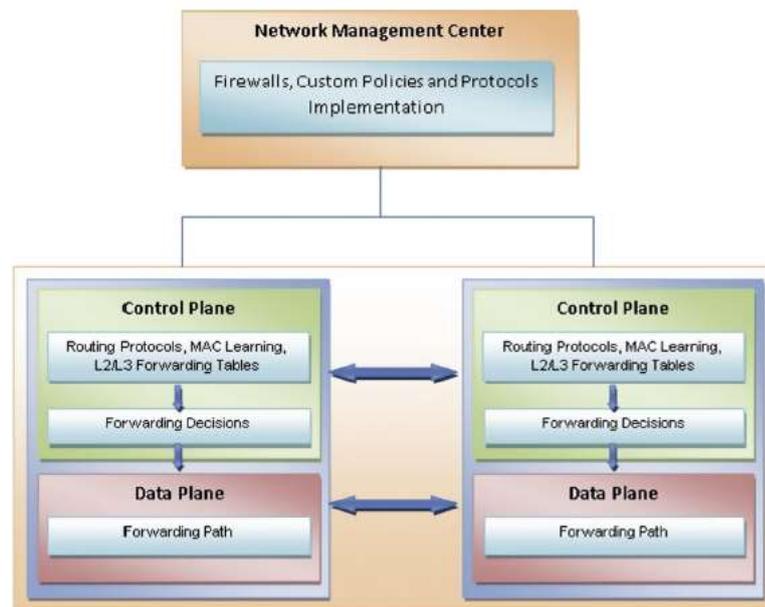
Menurut buku yang berjudul “*The Disaster Recovery Handbook*” (Wallace & Webber, 2017). Pada umumnya seluruh orang ketika berfikir tentang *disaster recovery*, mereka hanya berfokus kepada teknik bagaimana *recovery plan* yang mereka akan lakukan. Meskipun seluruh teknik sudah dijelaskan hingga sangat detail tentang fungsi, hasil, data hingga skema tetapi masih sangat sedikit yang menjelaskan hingga *step by step* untuk *rebuilding* seluruh *server* didalam *data center*. Seluruh *recovery* bertujuan untuk perusahaan agar tetap berjalan ketika terjadi kegagalan *system* yang menyebabkan harus berpindah data dari perangkat utama ke dalam perangkat *backup*. Seluruh skema tersebut harus dijelaskan dan dikategorikan menjadi beberapa perencanaan

- Melakukan identifikasi *Business Impact Analyst (BIA)*
- Melakukan klasifikasi seluruh fungsi bisnis yang kritikal menggunakan BIA
- Melakukan pemindahakan seluruh data ke dalam perangkat DRC
- Memindahkan akses internet ke dalam internet backup
- Melakukan pemindahakan sistem dan aplikasi ke perangkat DRC

### **2.4 Software Defined Networking (SDN)**

Protokol jaringan *transport* dan *control* terdistribusi berjalan didalam *router* dan *switch* yang merupakan kunci utama untuk melakukan *forwarding data*. Meskipun persebaran *router* dan *switch* tersebar luas namun kebanyakan masih diterapkan dengan menggunakan *traditional IP network* yang kompleks dan lebih sulit untuk dipelihara. Untuk melakukan perubahan pada semua perangkat,

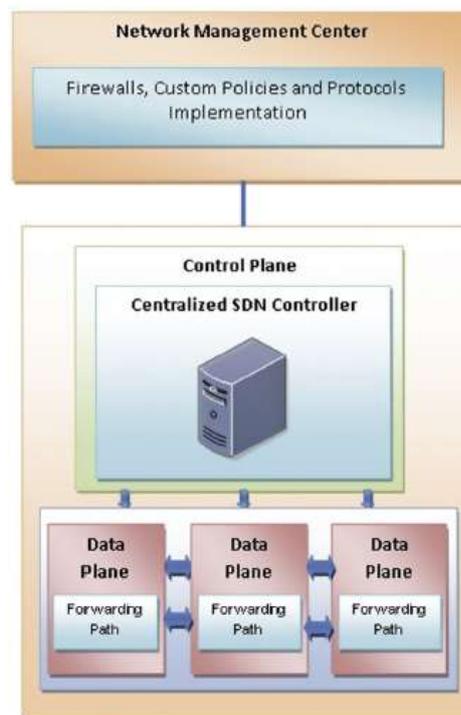
*network administrator* harus melakukan konfigurasi satu-satu pada setiap *site*, terlebih jika terdapat perangkat yang berbeda *merk* biasanya juga menggunakan *command* tersendiri. Sehingga untuk melakukan *automatic reconfiguration* dan *response mechanism* masih sulit diterapkan pada jaringan *IP network* sekarang ini karena *data plane* dan *control plane* masih menjadi satu pada satu perangkat seperti pada gambar 2.4 dibawah ini (Jammal, Singh, Shami, Asal, & Li, 2014).



**Gambar 2.4 Inflexible Infrastructure**

*Software Defined Network* merupakan sebuah paradigma jaringan yang muncul yang memberikan perubahan untuk mengatasi keterbatasan infrastruktur jaringan saat ini. Pertama, SDN akan memisahkan integrasi antara *control plane* dan *data plane*. Kedua, dengan memisahkan antar keduanya perangkat *forwarding (switch, router)* menjadi lebih ringan dalam melakukan transfer data dan *control plane* diimplementasikan secara terpusat, menyederhanakan kebijakan, dan memudahkan untuk *network reconfiguration*.

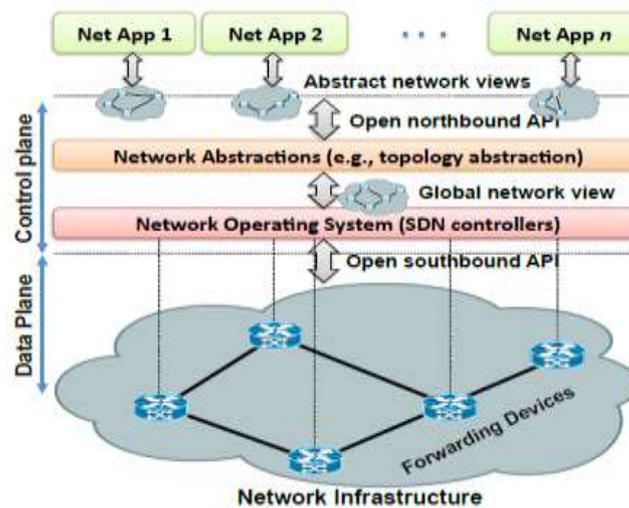
Pada jaringan computer dapat dibagi menjadi tiga fungsionalitas yaitu *data plane*, *control plane* dan *management plane* seperti pada gambar 2.4. *Data plane* berhubungan dengan perangkat jaringan yang bertugas untuk *transfer data*, *control plane* berhubungan dengan *protocol* yang digunakan untuk transfer data yang digunakan oleh *data plane*, *management plane* berhubungan dengan pengelolaan, *remote* dan monitor perangkat. Sebagai contoh *network policy* didefinisikan oleh *management plane*, diatur oleh *control plane* dan dilaksanakan dengan mentransfer data oleh *data plane*.



**Gambar 2.5 Fungsionalitas jaringan komputer**

Merujuk pada gambar 2.5 diatas, *traditional ip network*, *data plane* dan *control plane* terdapat pada satu perangkat yang sama dan keseluruhan struktur *network* terdesentralisasi, sehingga sangat kompleks dalam

melakukan pemeliharaan dan *control*. Untuk itu dikembangkan teknologi yang namanya *Software Defined Network* (SDN). Pada gambar 2.6, SDN terdapat 2 elemen penting yaitu *controller* (*control plane*) dan *forwarding devices* (*data plane*), dimana *controller* merupakan sebuah *software* atau disebut *network brain* yang bertugas untuk mengontrol *forwarding device*, sedangkan *forwarding device* adalah *hardware* maupun *software* yang bertugas untuk melakukan transfer data.



**Gambar 2.6** Arsitektur SDN

SDN dikembangkan sebagai arsitektur jaringan dengan mempunyai empat pilar:

1. *control plane* dan *data plane* dipisahkan, fungsionalitas *control* dihapuskan dari perangkat jaringan dan perangkat jaringan hanya bertugas untuk meneruskan paket
2. *forwarding packet* pada SDN berbasis *flow packet*, bukan berdasarkan *destination network*. Jadi *flow packet* dapat diartikan sebagai satu set paket berdasarkan pencocokan (*filter*) dan instruksi

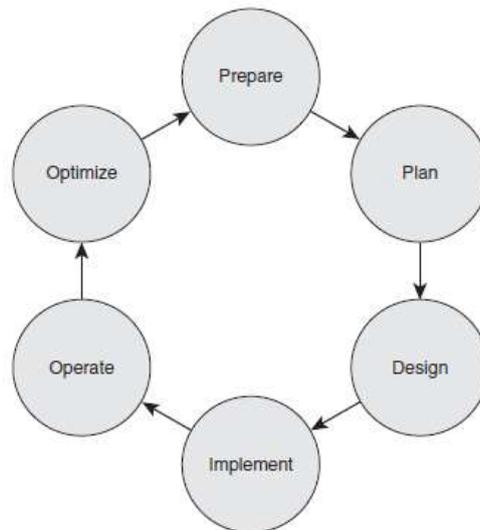
3. *Control plane* dipisahkan ke entitas eksternal yang dinamakan *controller*, *controller* ini berjalan pada teknologi *server* dan memberikan kemudahan dalam melakukan pemrograman terhadap perangkat networking secara terpusat
4. Perangkat network diprogram atau dikonfigurasi dari satu *controller*, ini merupakan karakteristik dari *Software Defined Network*, yang dianggap sebagai nilai utamanya. Dengan *control plane* yang terpusat mempunyai manfaat untuk mengatur semua perangkat jaringan, diantaranya yaitu kemudahan melakukan konfigurasi perangkat yang terhubung dengan SDN.

## 2.5 PPDIOO Life Cycles

Pada penelitian ini menggunakan metodologi PPDIOO. Metodologi ini merupakan metodologi yang menggambarkan elemen utama untuk mendesain sebuah jaringan. Merujuk pada gambar 2.7 dibawah, menurut (Bruno & Jordan, n.d.) Cisco telah menetapkan siklus hidup jaringan menjadi enam tahap, yaitu: *Prepare*, *Plan*, *Design*, *Implement*, *Operate* dan *Optimize* yang lebih dikenal sebagai PPDIOO. PPDIOO *life cycle* mempunyai empat kelebihan:

1. Menurunkan *Total Cost Ownership* (TCO) dengan memvalidasi persyaratan dan perencanaan teknologi untuk perubahan infrastruktur dan kebutuhan sumber daya
2. Meningkatkan ketersediaan jaringan dengan desain jaringan yang tepat

3. Meningkatkan kehandalan bisnis dengan cara menetapkan persyaratan bisnis dan strategi teknologi
4. Meningkatkan kecepatan akses ke aplikasi dengan cara meningkatkan *avaibility, reliability, security, scalability* dan *performance*



**Gambar 2.7 PPDIIO Network Life Cycle**

1. *Prepare*

Pada fase ini digunakan untuk menetapkan kebutuhan organisasi dan bisnis, mengembangkan strategi jaringan yang akan diimplementasi dan mengusulkan konsep *high-level architecture* untuk mendukung strategi yang akan dijalankan.

2. *Plan*

Pada fase ini mengidentifikasi kebutuhan jaringan berdasarkan tujuan dan kebutuhan user dengan melakukan *assessment network* dan sarana pendukung, melakukan terhadap *review* desain arsitektur jaringan sesuai dengan best practice. *Project plan* digunakan untuk mengatur tugas, tanggungjawab masing-masing

bagian, *milestone*, dan sumber daya untuk melakukan desain dan implementasi proyek. *Project plan* harus sesuai dengan ruang lingkup pekerjaan, biaya proyek dan sumber daya yang digunakan untuk memenuhi kebutuhan user

### 3. *Design*

*Network design* dikembangkan berdasarkan kebutuhan bisnis dan kebutuhan teknik yang telah didefinisikan pada fase sebelumnya. *Network design* yang dibuat lebih detail yang mencakup topologi jaringan dan daftar perangkat serta dalam topologi yang diusulkan tidak terlepas dari fungsi *high availability*, *reliability*, *security*, dan *performance*. Setelah *network design* disetujui oleh *user* maka dapat dilanjutkan ke tahap implementasi

### 4. *Implement*

Pada fase ini dilakukan instalasi dan konfigurasi perangkat berdasarkan desain yang telah dibuat pada fase sebelumnya dan berdasarkan waktu implementasi. Fase implementasi ini juga termasuk pengetesan perangkat atau sistem sebelum dioperasikan

### 5. *Operate*

Pada fase ini bagaimana perangkat yang telah diimplementasikan dilakukan monitoring baik dari status perangkat, *performance*, maupun melakukan *corrective maintenance*

### 6. *Optimize*

Fase ini merupakan proses identifikasi dan penyelesaian isu yang muncul untuk menghindari sistem atau *hardware failure* pada

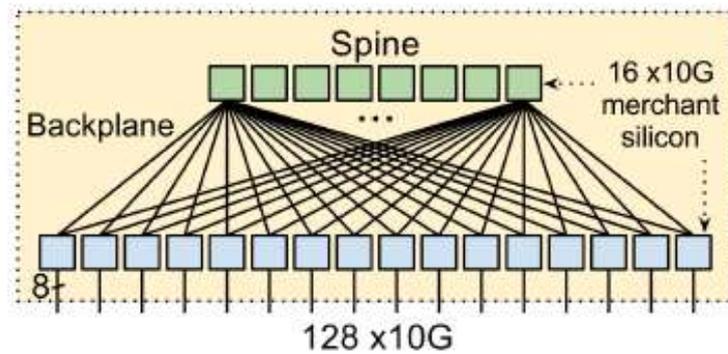
perangkat yang telah diimplementasi. Pada fase ini juga memungkinkan untuk dilakukan pengembangan jaringan berdasarkan isu yang pernah terjadi sebelumnya

## 2.6 Penelitian Sejenis

Sebenarnya konsep dari teknologi *Software Defined Network* (SDN) tidak sepenuhnya baru. Faktanya beberapa dekade yang lalu sudah ada yang menggunakan infrastruktur khusus seperti *cloud computing* untuk memisahkan jaringan sistem operasi (mirip dengan fungsi kontrol di SDN) dengan aplikasi komputasi intensif (mirip dengan *data plane*) (Hu, Hao, & Bao, 2014). Gagasan mengenai *programmable network* baru-baru ini mendapat momentum yang cukup besar dengan kemunculan paradigma *Software Defined Networking* (Nunes, Mendonca, Nguyen, Obraczka, & Turlitti, 2014).

Pada awal tahun 2012, Menurut (Jain et al., 2013) Google mengimplementasikan konsep dan arsitektur SDN pada skala besar untuk menghubungkan antar *data center* nya diseluruh dunia. Pekerjaan ini menyajikan desain yang lebih detail, implementasi, dan evaluasi dari B4 (koneksi WAN untuk menghubungkan *data center* Google seluruh dunia). Pekerjaan ini dinilai merupakan implementasi SDN paling besar didunia. Motivasi Google dalam implementasi ini untuk mewujudkan kustomisasi *routing protocol* dan *traffic engineering*, sehingga dapat mendukung *level of scalability*, *fault tolerance*, *cost efficiency*, dan *control*. Dimana semua aspek tersebut belum dapat terpenuhi jika masih menggunakan *traditional WAN architecture*. Setelah tiga tahun berjalan untuk operasional, B4 terbukti efisien dalam penggunaan link yang utilitasnya mencapai 100% dan telah dibagi menjadi beberapa *link/path* (Nunes et al., 2014),

selain itu membantu perusahaan untuk meningkatkan efisiensi operasional dan secara signifikan mengurangi biaya (Kreutz et al., 2015). Berikut custom built dan topologinya terlihat pada gambar 2.1 berikut ini.



**Gambar 2.8 Custom built dan topologi**

Penelitian terkait tentang teknologi *software defined network* yang pernah dilakukan oleh (Li, Shang, & Chen, 2014) yang berjudul *Software Defined Green Data Center Network with Exclusive Routing* yaitu melakukan penelitian terkait bagaimana untuk mendapatkan hasil *network capacity* yang optimal, banyak arsitektur *data center* yang diusulkan untuk mereplace *traditional tree architecture*, seperti *fat tree*, *Bcube*, dll. karena arsitektur tradisional identik dengan banyak *switch* dan *traffic data* yang bersifat dua arah ketika terjadi pertukaran *traffic data* pada *traffic peak time*. dengan *traditional tree architecture* ada *2 side effect* pada *data center consumption*. yang pertama dapat meningkatkan konsumsi energi listrik karena penggunaan *switch* dalam jumlah yang banyak. kedua menyebabkan konsumsi energi jaringan yang tidak efisien ketika *switch* sedang *low utilization*, karena meskipun *low utilization* energi yang diserap tetap sama. Dengan hasil penelitian yaitu menerapkan metode *flow scheduling* untuk

mengurangi konsumsi energi pada *data center network* yang menerapkan teknologi *software defined network* dengan dan mempersingkat waktu *traffic flow* rata-rata.

Penelitian lainnya yang berhubungan dengan *software defined network* dilakukan oleh (Yu et al., 2017) yaitu dengan menerapkan teknologi SDN pada *data center 1* dan *data center 2* pada kelas *enterprise* berhasil mengintegrasikan sebagai satu *resource pool* dengan melalui L3 *network*. Pada penelitian ini juga diterapkan *distributed logical firewall* dan *load balancer* yang diintegrasikan dengan SDN dan dapat mengurangi *policy* yang diterapkan di *core switch* dari 20 ribu *policy* bisa dijadikan menjadi 3 ribu *policy* dengan menggunakan *distributed firewall*. Dalam hal implementasi sistem baru dapat mengurangi waktu implementasi dari 2 minggu menjadi 1 hari dengan fleksibilitas penambahan perangkat yang diintegrasikan dengan SDN. Dan dari segi pemeliharaan perangkat pada *data center*, itu mempercepat fusi dan revolusi fungsi pekerjaan untuk platform, sistem dan staf jaringan. Hal ini juga mengarah pada kebutuhan yang lebih tinggi bagi para manajer di data center dan efisiensi bisnis yang lebih tinggi. Staf yang bekerja pada pemeliharaan peralatan jaringan dikurangi dari 5 menjadi 2. Personel tambahan dapat ditransfer ke grup lain yang bertanggung jawab atas manajemen atau pemeliharaan.

Penelitian mengenai SDN juga dilakukan oleh (Akyildiz, Lee, Wang, Luo, & Chou, 2016) yang berjudul *Research Challenge for Traffic Engineering in Software Defined Networks*. Dalam penelitian ini membahas mengenai SDN yang mewakili arsitektur baru, yang bersifat fleksibel, dan

terbuka itu memungkinkan pengaturan perilaku jaringan yang dinamis dan tepat waktu di jaringan komputer yang kompleks dan berskala besar. Karena SDN mempercepat inovasi dan evolusi jaringan data modern, ini membutuhkan sistem TE yang sangat skalabel dan cerdas. Artikel ini menyelidiki solusi TE SDN dari perspektif *traffic management, load balancing, fault tolerance, topology update, traffic analysis*. Keadaan saat ini dan tantangan penelitian dari SDN TE disajikan dengan membahas metrik kinerja SDN utama dalam hal *scalability, availability, reability, consistency, accuracy*.

**Tabel 2.1 Penelitian sejenis**

Penulis	Judul Penelitian	Tujuan Penelitian	Hasil
(Jain et al., 2013)	<i>B4: Experience with a Globally-Deployed Software Defined WAN</i>	Tujuan dari penelitian ini adalah membuat desain, evaluasi pada implementasi SDN WAN untuk data center connectivity, dengan pendekatan memisahkan network control plane dan network data plane untuk mewujudkan deployment yang cepat pada jaringan baru	Teknologi SDN yang telah diimplementasikan sudah berjalan hingga saat ini untuk melayani pelanggan dengan konektivitas WAN dan memiliki tingkat pertumbuhan yang lebih tinggi. B4 telah memungkinkan untuk menggunakan bandwidth WAN yang sangat efektif, menjalankan banyak tautan dengan utilisasi mendekati 100% untuk waktu yang lama. Berdasarkan pengalaman, hambatan dalam menjembatani paket protokol dari control plane ke data plane dan overhead dalam pemrograman perangkat keras adalah bidang penting untuk pekerjaan di masa depan.

Penulis	Judul Penelitian	Tujuan Penelitian	Hasil
(Li et al., 2014)	Software Defined Green Data Center Network with Exclusive Routing	Tujuan dari penelitian ini adalah memanfaatkan teknologi software defined network (SDN) dan mengeksplorasi solusi baru untuk aliran traffic data yang bersifat energy-aware. Untuk mendapatkan hasil network capacity yang optimal, banyak arsitekur data center yang diusulkan untuk mereplace traditional tree architecture, seperti fat tree, Bcube, dll.	Pada penelitian ini hasilnya adalah menerapkan metode flow scheduling untuk mengurangi konsumsi energy pada data center network yang menerapkan teknologi software defined network dengan dan mempersingkat waktu traffic flow rata-rata.
(Akyildiz et al., 2016)	Research Challenges for Traffic Engineering in Software Defined Networks	Penelitian ini membahas mengenai SDN yang mewakili arsitektur baru, yang bersifat fleksibel, dan terbuka yang memungkinkan pengaturan perilaku jaringan yang dinamis pada jaringan komputer yang kompleks dan berskala besar.	Dalam perancangannya SDN dapat mempercepat inovasi dan evolusi jaringan data modern, ini membutuhkan sistem TE yang scalable dan cerdas. Adapun perlu dikembangkan solusi TE SDN dari perspektif traffic flow, load balance, high availability, fault tolerance
(Yu et al., 2017)	Application of Software-Defined Network with Software-based Architecture in Enterprise Data Center	Tujuan dilakukannya penelitian ini adalah dengan menerapkan teknologi software defined network untuk menghubungkan data center 1 dan data center 2 serta distributed firewall dan load balancer. serta untuk mengetahui peranan penerapan teknologi SDN pada masa pemeliharaan	Dengan diterapkannya teknologi SDN pada data center 1 dan data center 2 pada kelas enterprise berhasil mengintegrasikan sebagai satu resource pool dengan melalui L3 network. Pada penelitian ini juga diterapkan distributed logical firewall dan load balancer yang diintegrasikan dengan SDN dan dapat mengurangi policy yang diterapkan di core switch

Penulis	Judul Penelitian	Tujuan Penelitian	Hasil
			<p>dari 20 ribu policy bisa dijadikan menjadi 3 ribu policy dengan menggunakan distributed firewall. Dan dari segi pemeliharaan perangkat pada data center, itu mempercepat fusi dan revolusi fungsi pekerjaan untuk platform, sistem dan staf jaringan. Hal ini juga mengarah pada kebutuhan yang lebih tinggi bagi para manajer di data center dan efisiensi bisnis yang lebih tinggi. Staf yang bekerja pada pemeliharaan peralatan jaringan dikurangi dari 5 menjadi 2. Personel tambahan dapat ditransfer ke grup lain yang bertanggung jawab atas manajemen atau pemeliharaan.</p>

