

## SOFTWARE DEFINED EVERYTHING (SDx/SDE) : PARADIGMA AUTOMASI JARINGAN INFRASTRUKTUR TEKNOLOGI INFORMASI

**Sri Wahyuni**

*Prodi Teknologi Informasi Universitas Islam Negeri Ar-Raniry  
Email : Sri.wahyuni@ar-raniry.ac.id*

**Abstract :** Over the past decade, Information Technology (IT) infrastructure has grown rapidly in line with user needs. Radio technologies such as WI-FI 6.5G and developments in optical transmission make connectivity even easier. Internet of Things (IoT) is an example of integrated connectivity systems. IoT changes the paradigm of IT infrastructure which is currently increasingly mobile and processed in the cloud. The widespread adoption of IoT comes easier and more convenient. An architectural solution is needed that makes network management effective. This article describes the results of the Library's study on Software Defined Everything (SDx/SDE) which presents a scalable, secure, and automated network management solution that connects users to applications and data that are scattered across multiple locations. SDx/SDE is represents a shift in controlling access to networks, data centers, and storage. The goal of SDx/SDE is increasing more programmability to the IT infrastructure by separating the management part from the hardware, so that it can be flexibly and automated according to user requirements.

**Keywords :** Software Defined Everything, ICT, Information technology

**Abstrak :** Selama dekade terakhir, infrastruktur Teknologi Informasi (TI) telah berkembang pesat seiring dengan kebutuhan pengguna. Teknologi radio seperti WI-FI 6, 5G dan perkembangan transmisi optik menjadikan konektivitas semakin mudah. *Internet of Things* (IoT) merupakan contoh konektivitas antar sistem yang terintegrasi. IoT mengubah paradigma infrastruktur IT yang saat ini semakin *mobile* dan diproses secara *cloud*. Semakin meluasnya adopsi IoT membuat kehidupan menjadi lebih mudah dan nyaman. Dibutuhkan solusi arsitektur yang memudahkan dalam melakukan manajemen jaringan. Artikel ini memaparkan hasil kajian Pustaka tentang *Software Defined Everything* (SDx/SDE) yang menghadirkan solusi dalam manajemen jaringan yang terukur, aman, dan otomatis menghubungkan pengguna kepada aplikasi dan data yang berpecah di beberapa lokasi. SDx/SDE merupakan suatu wadah yang menunjukkan pergeseran dalam mengontrol akses ke jaringan, pusat data, dan penyimpanan. Tujuan dari SDx/SDE adalah menambahkan lebih banyak programabilitas ke dalam infrastruktur TI dengan memisahkan bagian manajemen dari perangkat keras, sehingga dapat diatur secara fleksibel dan otomatis sesuai kebutuhan pengguna.

**Kata kunci :** Software Defined Everything, ICT, Teknologi Informasi

## 1. PENDAHULUAN

Sebuah teknologi baru diciptakan untuk membuat sesuatu yang rumit menjadikan lebih simple dan ringkas. Kebutuhan akses pengguna internet yang semakin beragam, akan berpengaruh terhadap pengembangan infrastruktur TI. Jumlah pengguna piranti cerdas yang terhubung tumbuh sangat cepat, sudah lebih dari satu milyar perangkat terkoneksi melalui internet (Durga & Reddy, 2015). Di era ICT saat ini masalah utama dari administrator adalah bagaimana membuat sebuah desain dan manajemen pemeliharaan lebih sederhana. Dimana peningkatan fleksibilitas, kehandalan, dan keamanan. Menggunakan perangkat lunak untuk melakukan optimalisasi dalam mengelola dan mengontrol sumber daya bukanlah hal baru. Konsep dari *Software Defined* adalah kemampuan dari perangkat lunak untuk mengontrol semua entitas sumber daya yang mendasari dari berbagai vendor dengan secara fisik mengisolasi keberagaman perangkat keras pada lapisan data (Darabseh et al., 2015).

Pada akhirnya teknologi *Software Defined* bertujuan untuk menyembunyikan semua kompleksitas dari manajemen dan fungsi control sumber daya sistem dari pengguna. Mentransfer konsep-konsep ini agar dapat diterapkan pada sistem bukanlah pekerjaan sederhana. Teknologi ICT konvensional masih sangat kompleks sehingga tantangan utama dari *software defined* adalah terkait pemisahan antara *control plane* dan *data plane* serta pemeliharaan layanan dalam kerangka sistem yang sudah ada.

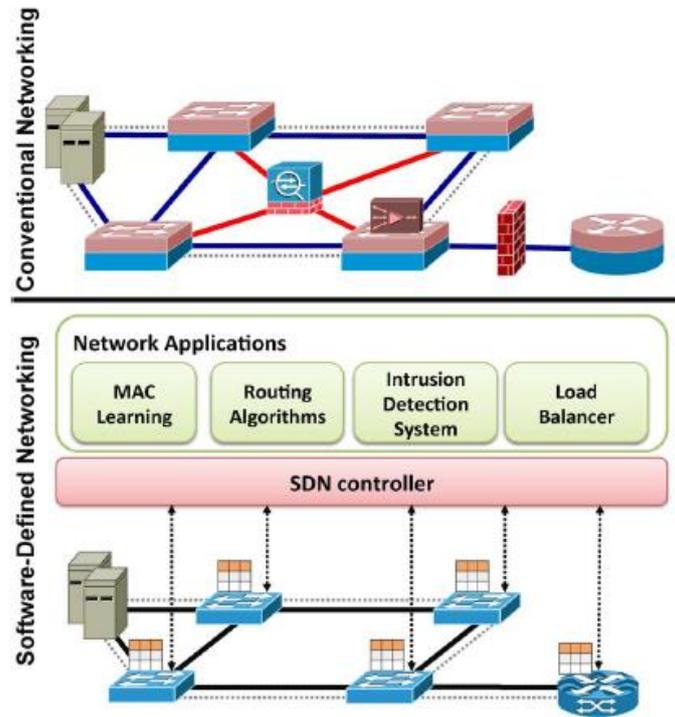
Dalam kajian ini berfokus untuk memperkenalkan macam-macam teknologi *Software Defined* (SD), gambaran, keuntungan, dan hambatan ketika diterapkan dalam dunia ICT berdasarkan literatur yang penulis dapatkan. Topik yang akan dikaji dalam makalah ini antara lain *Software Defined Networking* (SDN), *Software Defined Security* (SDSec), *Software Defined Storage* (SDS), dan *Software Defined Infrastructure* (SDI).

## 2. KAJIAN KEPUSTAKAAN

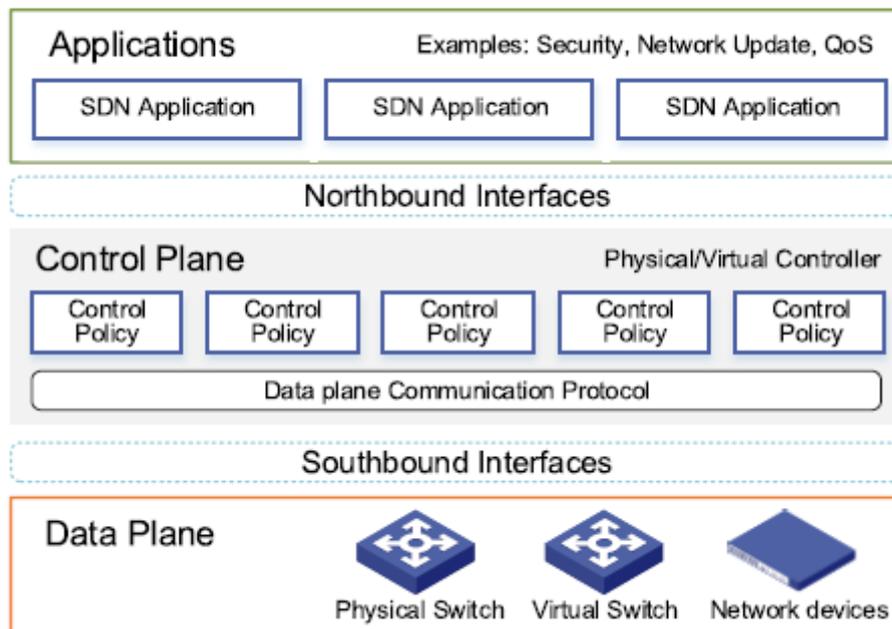
### I. SOFTWARE DEFINED NETWORKING (SDN)

Tren perkembangan ICT terkini yang melibatkan selular, *cloud*, *big data* (Manyika et al., 2011) mengakibatkan jaringan komputer untuk mempersiapkan jaringan dengan bandwidth yang tinggi, akses yang mudah, dan manajemen yang dinamis. Saat ini dimulai era baru yang secara fungsionalitas mengimplementasikan dan mengatur jaringan komputer berbasis perangkat lunak (Rana et al., 2019). *Software Defined Networking* (SDN) adalah arsitektur jaringan yang muncul dimana kontrol jaringan (*control plane*) dipisahkan dengan bidang data (*data plane*) yang dapat diprogram secara langsung (gambar.1). (Olsson et al., 2013) (Nunes et al., 2014). Arsitektur SDN dibagi menjadi empat pilar. Pertama, Pemisahan antara control dan data plane tersebut memudahkan dalam pengembangan protocol dan

aplikasi baru, melakukan visualisasi secara langsung, dan konsolidasi dari berbagai perangkat *middlebox* ke dalam control perangkat lunak. Sehingga terjadi penyerdahanan proses. Kedua, *forwarding* data berdasarkan dari aliran data tersebut, bukan berbasis tujuan (Alvizu et al., 2017). Ketiga, logika control dipindahkan ke entitas eksternal, atau biasa disebut *SDN controller*. Keempat, Aplikasi yang berjalan diatas *SDN controller* mengatur jaringan melalui perangkat lunak yang berinteraksi dengan perangkat data plane.



**Gambar 1.** SDN Pemisahan antara *Control Plane* dan *Data Plane* (Kreutz et al., 2015)



**Gambar 2.** Arsitektur *Software Defined Networking* (SDN) (Li & Chen, 2015)

Keuntungan penggunaan SDN dengan kemampuannya memperoleh status jaringan secara instan, SDN memungkinkan control terpusat secara *realtime* dari jaringan. Hal ini berpengaruh kepada optimalisasi konfigurasi jaringan dan meningkatkan kinerja jaringan. SDN menawarkan platform yang nyaman untuk eksperimen Teknik baru/ dan mendorong desain jaringan baru. Selain itu terdapat beberapa keuntungan dari SDN diantaranya :

- a. Meningkatkan efektifitas jaringan : sebagai contoh Ketika ada perangkat baru ditambahkan ke dalam jaringan, diperlukan konfigurasi untuk itu. Namun, prosedur konfigurasi manual tersebut rawan dengan kesalahan. Dengan SDN prosedur tersebut dapat dilakukan secara otomatisasi
- b. Meningkatkan performa : Dengan memisahkan control plane menjadi tersentralisasi performa akan menjadi meningkat dengan algoritma pemrograman yang sesuai dapat dengan mudah diatur. Metode ini akan banyak menyelesaikan masalah seperti pengaturan trafik (Al-Fares et al., 2010), mengontrol kepadatan trafik (Ghobadi et al., n.d.), *load balancing* trafik (Handigol & Seetharaman, 2010), dan Quality of Services (QoS) dapat tertangani dengan efektif.
- c. Mendorong inovasi : otomatisasi pada SDN membuat ide-ide baru didalam pengembangannya. Akan banyak melibatkan transisi dari fase eksperimental ke fase operasional, sehingga banyak inovasi yang muncul didalam prosesnya.

- d. Mengurangi biaya perangkat keras : konsep dari SDN membangun jaringan menggunakan optimalisasi perangkat lunak dengan kebutuhan perangkat keras yang minimal.(Rana et al., 2019)(Heller et al., 2010)

Teknologi jaringan ICT akan mengadopsi SDN sebagai solusi atas semakin tingginya permintaan dari pengguna. Inovasi dari SDN akan menyelesaikan beberapa masalah dari teknologi jaringan konvensional (tabel.1) . SDN meningkatkan dengan signifikan performa, skalabilitas, keamanan, dan ketahanan sistem. Terkait itu akan terjadi perubahan *controller* dan desain jaringan. Nantinya semakin banyak pemetaan ke model virtualisasi untuk mendukung proses migrasi teknologi SDN yang berbasis perangkat lunak.

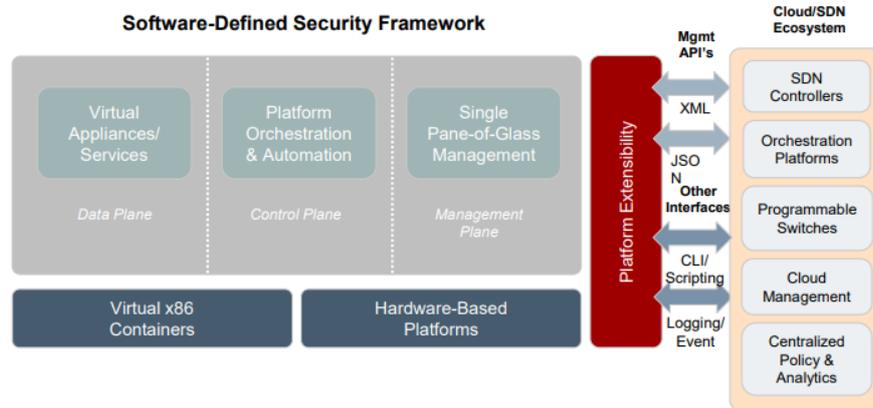
**Tabel 1.** Perbedaan antara SDN dan jaringan konvensional(Xia et al., 2015)

	SDN	Conventional Networking
Features	decoupled data and control plane, and programmability	a new protocol per problem, complex network control [49]
Configuration	automated configuration with centralized validation	error prone manual configuration
Performance	dynamic global control with cross layer information	limited information, and relatively static configuration
Innovation	easy software implementation for new ideas, sufficient test environment with isolation, and quick deployment using software upgrade	difficult hardware implementation for new ideas, limited testing environment, long standardization process

## II. SOFTWARE DEFINED SECURITY (SDSec)

Software Defined Security (SDSec) dapat didefinisikan sebagai jenis model keamanan di mana keamanan dalam lingkungan komputasi diimplementasikan, dikendalikan, dan dikelola oleh perangkat lunak keamanan (*What is Software-Defined Security (SDS)? - Definition from Techopedia, n.d.*). Ini adalah keamanan yang dikelola perangkat lunak, digerakkan oleh kebijakan, dan diatur yang dapat terdiri dari firewall; deteksi dan pencegahan intrusi; segmentasi jaringan; pemindaian aktif dan pasif; identifikasi pengguna dan perangkat; kontrol aplikasi; pencegahan ancaman; dekripsi; dan banyak lagi (*What Is Software-Defined Security? / Arrow ECS NA, n.d.*).

Arsitektur keamanan teknologi informasi tradisional untuk pusat data secara fisik gagal memenuhi ekspektasi kebutuhan bisnis digital yang berkembang pesat. SDDSec memungkinkan penerapan control keamanan yang canggih dan otomatis untuk mengamankan data dan sumber daya jaringan di pusat data yang ditentukan perangkat lunak. SDDSec diadopsi sebagai keamanan jaringan generasi berikutnya untuk pusat data virtual. Perlindungan yang dimungkinkan oleh SDDSec bersifat adaptif dan tidak bergantung pada server dan perangkat keamanan individu apa pun.



**Gambar 3.** *Software Defined Security (SDSec) Framework*

SDSec sering diimplementasikan di lingkungan TI yang memiliki infrastruktur cloud atau virtualisasi. Setiap kali perangkat baru dimasukkan ke dalam lingkungan, perangkat tersebut otomatis tercakup dan dikontrol oleh kebijakan keamanan yang telah ditetapkan. Ini juga berarti bahwa lingkungan dapat dimigrasi ke pusat data lain tanpa memengaruhi kebijakan dan kontrol keamanan yang ada.

Dengan SDSec, kebijakan keamanan dapat mengikuti pengguna alih-alih perangkat - baik itu laptop, ponsel cerdas, tablet, atau desktop virtual. Jika seorang karyawan meninggalkan kantor dengan salah satu perangkat ini, kebijakan atau profil keamanan yang sama berlaku seolah-olah mereka berada di kantor. Karyawan tidak harus bergantung pada kafe atau jaringan hotel untuk memberikan kontrol keamanan. Persyaratan untuk SDSec meliputi:

- a. Arsitekturnya harus sederhana untuk dipasang dan dikelola dalam lingkungan yang sangat dinamis.
- b. SDSec harus hemat biaya untuk memastikan keamanan dapat dipasang di mana saja.
- c. Itu harus aman untuk melindungi secara efektif dari ancaman tingkat lanjut. (Dijiang Huang et al., n.d.)

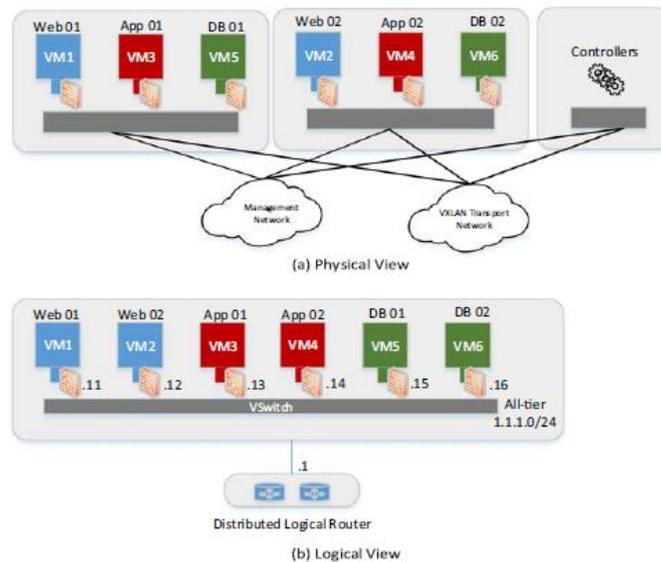
Dengan menggunakan SDSec Administrator pada penyedia jasa layanan atau administrator penyewa dapat mengatur kebijakan firewall di pusat data melalui *network controller* dan *northbound Application Programming Interface (API)*. Pusat layanan firewall menawarkan beberapa keuntungan bagi penyedia layanan diantaranya.

- a. Solusi firewall berbasis perangkat lunak yang sangat skalabel, dapat dikelola, dan dapat didiagnosis yang dapat ditawarkan kepada penyewa
- b. Kebebasan untuk memindahkan mesin virtual penyewa ke host komputasi yang berbeda tanpa melanggar kebijakan firewall penyewa

- c. Menawarkan perlindungan untuk mesin virtual penyewa yang independen dari sistem operasi tamu penyewa.

Sedangkan pusat data firewall menawarkan beberapa keuntungan berikut untuk penyewa :

- a. Kemampuan untuk menentukan aturan firewall untuk membantu melindungi beban kerja yang dihadapi Internet di jaringan virtual.
- b. Kemampuan untuk menentukan aturan firewall untuk membantu melindungi lalu lintas antar virtual mesin pada subnet virtual L2 yang sama serta antara virtual mesin pada subnet virtual L2 yang berbeda
- c. Kemampuan untuk menentukan aturan firewall untuk membantu melindungi dan mengisolasi jaringan lalu lintas antara penyewa di jaringan premis dan pada jaringan virtual mereka di penyedia layanan.



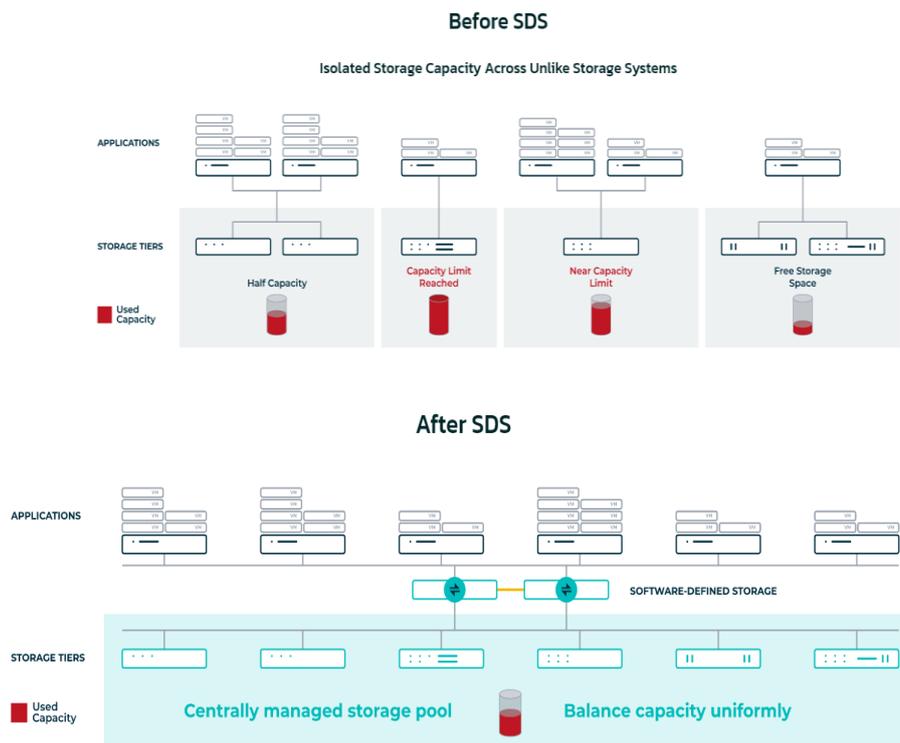
**Gambar 4.** Abstraksi Topologi Fisik SDSec

### III. SOFTWARE DEFINED STORAGE (SDS)

Perkembangan infrastruktur jaringan mengakibatkan perubahan perubahan tren dunia terhadap digitalisasi. Ahli Statistik mengatakan setiap harinya, kira-kira 15 petabyte data baru sedang dihasilkan di seluruh dunia dan jumlah total digital data dua kali lipat setiap dua tahun. Pertumbuhan tersebut mengharuskan pusat data cepat dalam berinovasi, untuk memenuhi kebutuhan pengguna. Kebanyakan pusat data saat ini, masih menggunakan mekanisme penyimpanan secara tradisional, dimana infrastruktur tersebut didesain secara spesifik untuk mendukung beban kerja dan pengguna tertentu. Sistem seperti ini mengakibatkan akses menjadi lebih

tertutup dan tidak fleksibel dalam kebutuhan pengguna yang lebih heterogen, karena akan menjadi lebih sulit dikelola. Bertambah pesatnya perusahaan dan pelanggan membutuhkan akses penyimpanan yang memudahkan dalam mengakses, berinovasi, serta lebih dinamis dalam pengaturannya, berkurangnya biaya dalam pengembangan dan pemeliharaan, dan dibutuhkan peningkatan skalabilitas.

*Software-defined storage* (SDS) adalah teknologi yang digunakan dalam manajemen penyimpanan data yang bertanggung jawab untuk memisahkan fungsi-fungsi dalam hal menyediakan kapasitas, melindungi data, dan mengontrol penempatan data dari perangkat keras fisik tempat data disimpan. Sehingga dengan menggunakan SDS dapat mengizinkan penyimpanan perangkat keras yang dengan mudah untuk diganti, ditingkatkan, dan diperluas tanpa perlu mencabut prosedur operasional atau membuang investasi perangkat lunak yang berharga. (*Software-Defined Storage: The Definitive Guide* | DataCore, n.d.). Dengan melakukan migrasi ke sistem SDS aspek tersebut akan terpenuhi seutuhnya. SDS memberikan pendekatan perangkat lunak dari infrastruktur tradisional ke model lingkungan virtualisasi yang meliputi perubahan dalam hal komputasi, jaringan, keamanan, ketersediaan server, dan media penyimpanan



**Gambar 5.** Perbedaan Topologi Sebelum dan Sesudah SDS

Perbandingan antara SDS dengan teknologi penyimpanan tradisional, dimana pada sistem tradisional yaitu desain perangkat keras sangat mengikat erat pada operasi penyimpanan ke perangkat atau pabrikan tertentu. Setiap merek dan model

menjalankan fungsi yang serupa namun menerapkan proses secara berbeda sehingga menyebabkan tidak kompatibel satu sama lain. Ketidakcocokan tersebut mengubah penyegaran perangkat keras yang kecil menjadi perbaikan operasional besar yang menjadi tertunda karena proses migrasi yang menyulitkan. Dalam bentuknya yang paling serbaguna, solusi SDS menyembunyikan keistimewaan perangkat keras berpemilik melalui lapisan perangkat lunak virtualisasi. Tidak seperti hypervisor yang membuat satu server tampak seperti banyak mesin virtual, SDS menggabungkan berbagai perangkat penyimpanan ke dalam kumpulan yang dikelola secara terpusat. Dengan memisahkan jenis penyimpanan kelas perusahaan yang menggunakan berbagai komoditas. Penggunaan penyimpanan virtual menjadikan lebih dinamis dalam pengoperasiannya. Sumber daya penyimpanan yang transparan memungkinkan pengguna untuk memaksimalkan ruang penyimpanan yang tidak terpakai untuk melayani permintaan lainnya. Pemanfaatan Infrastruktur yang efisien dan fleksibel menyebabkan biaya penyimpanan akan turun. Virtualisasi penyimpanan menghasilkan skalabilitas penyimpanan, penggantian, substitusi, dan pengelolaan. Penerapan SDS yang memisahkan perangkat lunak dari perangkat keras memungkinkan perusahaan melakukan pembelian, penerapan, dan pengoperasian perangkat keras secara independent tanpa perlu khawatir terhadap isu kompatibilitas maupun interoperabilitas penyimpanan sumber daya. (Raj & Raman, n.d.)

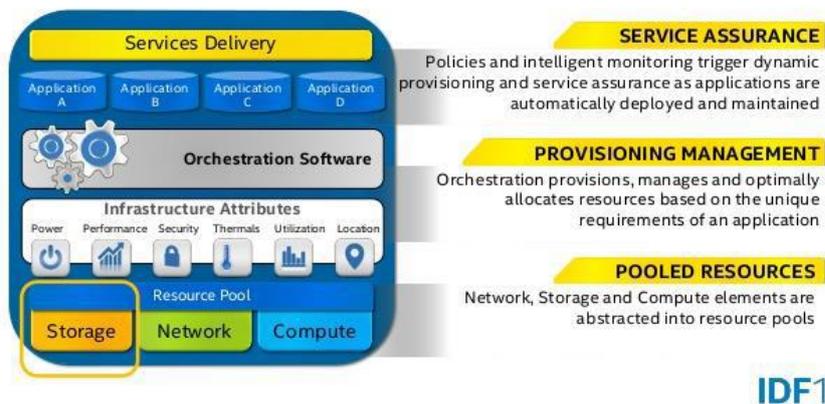
#### IV. SOFTWARE DEFINED INFRASTRUCTURE (SDI)

Perkembangan IT yang sangat cepat menyebabkan Infrastruktur IT turut berubah. Banyak inovasi yang memicu menjadikan komoditas. Perencanaan model infrastruktur, kerangka kerja, dan proses operasi dalam suatu Infrastruktur harus dipikirkan dengan detail yang dapat mengikuti dengan perkembangan teknologi dan inovasi terkini. Desain model infrastruktur nantinya akan mempengaruhi perubahan ekonomi dan model bisnis suatu perusahaan. Dibutuhkan pengembangan yang mendukung fleksibilitas, personalisasi, keamanan, dan efisiensi. Hal itu tidak dapat terjadi dengan mengikuti infrastruktur tradisional dimana model tersebut sangat bergantung kepada perangkat keras, sehingga hampir tidak mungkin untuk diaplikasikan.

*Software Defined Infrastructure* (SDI) adalah teknologi infrastruktur komputasi yang sepenuhnya dikontrol oleh perangkat lunak tanpa campur tangan operator atau manusia. SDI memungkinkan untuk melakukan virtualisasi hampir disemua aspek lingkungan seperti : sumber daya komputasi, jaringan, dan penyimpanan. Biasanya berupa kombinasi teknologi yang diatur perangkat lunak. Misalkan *Software Defined Storage* (SDS) dengan *Software Defined Networking* (SDN). (*Software-defined infrastructure - IBM Documentation*, n.d.) (*Software-defined infrastructure - Wikipedia*, n.d.). Implementasi SDI harus integrasi dengan mulus sehingga

pengguna dan pembeli tidak mengetahui dan tidak peduli apakah produk atau layanan tersebut berasal dari komputasi, jaringan, ataupun penyimpanan.

## Software Defined Infrastructure (SDI)



**Gambar 6.** *Software Defined Infrastructure* (Cohen & States, 2017)

Dalam pengembangannya, SDI memiliki beberapa atribut yang harus dimiliki infrastruktur umum untuk pengguna akhir dan vendor :

- Inovasi di perangkat lunak yang siap pakai didalam perangkat keras : SDI harus berpusat kepada perangkat lunak yang lebih mudah dalam memenuhi persyaratan dalam hal kelincahan, otomatisasi, dan efisiensi. Vendor tidak perlu mengembangkan solusi berbasis perangkat keras dalam memecahkan masalah infrastruktur. Vendor hanya cukup focus untuk mempersiapkan perangkat keras komersial yang siap pakai daripada kemampuan dalam hal komputasi maupun penanganan data.
- Ketergantungan terhadap virtualisasi : Dengan memisahkan aplikasi dari perangkat keras memudahkan dalam portabilitas, yang meningkatkan kelincahan dan pilihan dalam pemeliharaan. (Cohen & States, 2017)
- Sistem diatur menurut kebijakan dan kebutuhan : Infrastruktur ini dapat terus dipantau jika terjadi suatu peristiwa dan perubahan maka akan secara otomatis bereaksi dan memperingatkan administrator secara *real time* untuk memastikan kebijakan yang dimaksudkan untuk diterapkan. Perangkat lunak harus memiliki kecerdasan tertanam agar dapat memproses sesuai dengan hasil yang diharapkan. Peran administrator membuat barisan algoritma skrip untuk melakukan otomatisasi yang kompleks.
- Berbasis konteks : SDI menambahkan fungsi otomatisasi berdasarkan konteks, misalkan pembuatan lingkungan cloud pribadi dan public, maka dibutuhkan mesin virtual, computer, penyimpanan, dan sumber daya jaringan untuk memaksimalkan kinerja, meminimalkan latensi, atau memenuhi tujuan bisnis tersebut.

- e. Arsitektur modular : konsep SDI harus dapat menyelesaikan masalah dengan fleksibel, praktis, dan cepat. Perangkat lunak yang diimplementasikan dalam sistem bersifat modular, fungsi-fungsi yang biasanya berupa template didalam perangkat lunak dapat ditukar dengan komponen lain yang serupa untuk menyelesaikan suatu masalah. Dengan model seperti itu sistem tidak terpengaruh ataupun tidak mengalami gangguan secara menyeluruh yang berarti tidak mengubah solusi arsitektur sepenuhnya.
- f. Dapat diperluas oleh pengguna akhir (*end user*) : sistem pada SDI fleksibel digunakan oleh pengguna akhir. Perangkat lunak mudah untuk ditambahkan fitur dan kemampuan baru pada aplikasi, yang mungkin memerlukan fungsionalitas baru di tingkat infrastruktur. Sering kali dalam pemeliharaan jaringan terdapat masalah baru yang diperlukan penanganan dengan cepat yang memerlukan kecerdasan pada perangkat lunak untuk melakukannya secara otomatisasi. Karena vendor tidak memungkinkan untuk melakukan kustomisasi yang berbeda pada masing-masing pengguna akhir. Pemanfaatan ekstensibilitas elemen SDI yang dapat diprogram mempermudah pengguna dalam membangun fungsionalitas baru.
- g. Dirancang untuk keamanan : kemudahan dalam mengatur sistem harus mengutamakan keamanan didalamnya. Aplikasi SDI memiliki akses ke data sangat sensitif yang dijadikan incaran oleh peretas. SDI memerlukan kemampuan dalam membatasi kontrol akses, penanganan data yang aman dan terenkripsi, ketahanan terhadap serangan, dan autentikasi yang kuat diantara komponen yang berinteraksi. (*Software Defined Everything Part 6: Infrastructure - SDxCentral*, n.d.)

SDI membawa peranan penting untuk menyukseskan dalam hal mengintegrasikan antar perangkat lunak yang menjadikan sebuah lingkungan infrastruktur IT yang dinamis. Namun dalam implementasi semua tergantung kebutuhan industri yang terkadang masih membutuhkan infrasturktur model lama.

### 3. KESIMPULAN

Pada artikel ini menjelaskan gambaran arsitektur dari berbagai macam *Software Defined*. Teknologi dengan pemrograman berbasis perangkat lunak kedepannya akan dipergunakan luas. Banyak keunggulan yang dapat menyelesaikan masalah dari sistem arsitektur tradisional. Pada masa depan penelitian dan pengembangan lebih berfokus pada inovasi dari optimalisasi dari sistem otomatisasi perangkat lunak agar manajemen menjadi lebih mudah, stabil, dan dinamis.

### Daftar Kepustakaan

Al-Fares, M., Radhakrishnan, S., Raghavan, B., Huang, N., & Vahdat, A. (2010). Hedera: Dynamic flow scheduling for data center networks. *Proceedings of*

---

- 
- NSDI 2010: 7th USENIX Symposium on Networked Systems Design and Implementation*, 281–295.
- Alvizu, R., Maier, G., Kukreja, N., Pattavina, A., Morro, R., Capello, A., & Cavazzoni, C. (2017). Comprehensive Survey on T-SDN: Software-Defined Networking for Transport Networks. *IEEE Communications Surveys and Tutorials*, 19(4), 2232–2283. <https://doi.org/10.1109/COMST.2017.2715220>
- Cohen, R. B., & States, U. (2017). *How Software is Transforming the U . S . Economy* (Issue July). <https://doi.org/10.13140/RG.2.2.13507.91686>
- Darabseh, A., Al-Ayyoub, M., Jararweh, Y., Benkhelifa, E., Vouk, M., & Rindos, A. (2015). SDStorage: A software defined storage experimental framework. *Proceedings - 2015 IEEE International Conference on Cloud Engineering, IC2E 2015*, 341–346. <https://doi.org/10.1109/IC2E.2015.60>
- Dijiang Huang, Ankur Chowdhary, & Sandeep Pisharody. (n.d.). *Software-Defined Networking and Security From Theory to Practice*. CRC Press.
- Durga, L. N., & Reddy, P. (2015). *Implementing Software - Defined Networking ( SDN ) in a Campus Environment*. 3(18), 1–6.
- Ghobadi, M., Yeganeh, S. H., & Ganjali, Y. (n.d.). *Rethinking End-to-End Congestion Control in Software-Defined Networks*.
- Handigol, N., & Seetharaman, S. (2010). Aster\* x: Load-balancing as a network primitive. *9th GENI Engineering Conference, June 2015*, 1–2.
- Heller, B., Seetharaman, S., Mahadevan, P., Yiakoumis, Y., Sharma, P., Banerjee, S., & McKeown, N. (2010). Elastictree: Saving energy in data center networks. *Proceedings of NSDI 2010: 7th USENIX Symposium on Networked Systems Design and Implementation*, 249–264.
- Kreutz, D., Ramos, F. M. V., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1), 14–76. <https://doi.org/10.1109/JPROC.2014.2371999>
- Li, Y., & Chen, M. (2015). Software-defined network function virtualization: A survey. *IEEE Access*, 3, 2542–2553. <https://doi.org/10.1109/ACCESS.2015.2499271>
- Manyika, J., Chui Brown, M., B. J., B., Dobbs, R., Roxburgh, C., & Hung Byers, A. (2011). Big data: The next frontier for innovation, competition and productivity. *McKinsey Global Institute, June*, 156.
- Nunes, B. A. A., Mendonca, M., Nguyen, X. N., Obraczka, K., & Turetletti, T. (2014). A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys and Tutorials*, 16(3), 1617–1634. <https://doi.org/10.1109/SURV.2014.012214.00180>
- Olsson, M., Sultana, S., Rommer, S., Frid, L., & Mulligan, C. (2013). Architecture

- Overview. *EPC and 4G Packet Networks*, 17–64.  
<https://doi.org/10.1016/b978-0-12-394595-2.00002-5>
- Raj, P., & Raman, A. (n.d.). *Computer Communications and Networks Software-Defined Cloud Centers Operational and Management Technologies and Tools*.
- Rana, D. S., Dhondiyal, S. A., & Chamoli, S. K. (2019). Software Defined Networking (SDN) Challenges, issues and Solution. *International Journal of Computer Sciences and Engineering*, 7(1), 884–889.  
<https://doi.org/10.26438/ijcse/v7i1.884889>
- Software-defined infrastructure - IBM Documentation*. (n.d.).
- Software-defined infrastructure - Wikipedia*. (n.d.).
- Software-Defined Storage: The Definitive Guide | DataCore*. (n.d.).
- Software Defined Everything Part 6: Infrastructure - SDxCentral*. (n.d.).
- What Is Software-Defined Security? | Arrow ECS NA*. (n.d.).
- What is Software-Defined Security (SDS)? - Definition from Techopedia*. (n.d.).
- Xia, W., Wen, Y., Foh, C. H., Niyato, D., & Xie, H. (2015). A Survey on Software-Defined Networking. *IEEE Communications Surveys and Tutorials*, 17(1), 27–51. <https://doi.org/10.1109/COMST.2014.2330903>