

## BAB II

### TINJAUAN PUSTAKA DAN LANDASAN TEORI

#### 2.1. Tinjauan Pustaka

Virtualisasi aplikasi merupakan salah satu dari delapan jenis teknik virtualisasi (Murphy, 2016). Teknik virtualisasi yang berpusat pada sebuah *server* dapat dimanfaatkan oleh pengguna untuk menjalankan beberapa aplikasi secara bersamaan (Afriandi, 2012). Inti dari virtualisasi ini adalah mendistribusikan aplikasi-aplikasi pada *server* agar dapat juga dijalankan pada *PC client* yang belum ataupun tidak memiliki aplikasi tertentu.

Beberapa penelitian yang berkaitan dengan virtualisasi aplikasi telah dilakukan sebelumnya. Salah satu penelitian tersebut dilakukan di perusahaan yang bergerak di bidang industri (Coto, 2016). Tujuan penelitian ini adalah untuk menyediakan kemudahan *maintenance* dan perizinan untuk *software* yang berlisensi. Virtualisasi aplikasi dibangun dengan sistem operasi *Windows Server 2008 R2*. Sementara itu, *tools* yang digunakan adalah *Citrix XenApp 6.5*. Hasil penelitian menunjukkan bahwa virtualisasi aplikasi dapat berjalan secara fleksibel, memungkinkan untuk dijalankan di berbagai *platform* seperti *Personal Computer (PC)* dan *smartphone*.

Penelitian mengenai virtualisasi juga pernah dilakukan dalam sebuah tesis (Larson, Nugroho, Pramana, & Tanto, 2015). Pada penelitian ini aplikasi diimplementasikan oleh *server* pada *Windows Server 2012 R2* menggunakan *platform Hyper-V*. Dari hasil penelitian mereka dapat disimpulkan bahwa proses penyimpanan data dapat terekam dengan cukup baik dan mengurangi resiko kehilangan data. Selain itu, sistem yang digunakan dalam

penyimpanan data-data membuat data saling terintegrasi sehingga dapat diakses dimana saja dengan cepat.

Dari beberapa penelitian tersebut, para peneliti menggunakan virtualisasi aplikasi untuk mendistribusikan aplikasi pada *server*. Pada penelitian-penelitian sebelumnya implementasi virtualisasi aplikasi dilakukan pada sistem operasi *server* yaitu *Windows Server*, baik itu versi 2008 R2 atau 2012 R2. Kesamaan lainnya adalah virtualisasi menyediakan kemudahan *maintenance* dan perizinan untuk *software* yang berlisensi. Hal itu juga yang mendorong penulis untuk melakukan penelitian dalam perancangan virtualisasi aplikasi.

Perbedaan dari penelitian-penelitian sebelumnya, penulis mengimplementasikan virtualisasi aplikasi dengan *RemoteApp* yang tersedia pada *Windows Server 2012 R2*. Berbeda dengan penelitian sebelumnya yang menggunakan *Citrix XenApp 6.5*. Virtualisasi aplikasi yang dibangun hanya ditujukan agar berjalan pada *Personal Computer (PC)*.

Oleh karena itu, penulis tertarik merancang virtualisasi aplikasi menggunakan *RemoteApp*, program yang belum digunakan pada penelitian-penelitian sebelumnya. Penelitian ini diharapkan dapat menjadi solusi alternatif bagi sebuah organisasi dalam mengoptimalkan sistem teknologi informasi di lingkungannya.

## **2.2. Landasan Teori**

### **2.2.1. Virtualisasi**

Virtualisasi adalah sebuah teknik untuk menyembunyikan karakteristik fisik dari sumber daya komputer dari bagaimana cara sistem lain, aplikasi atau pengguna berinteraksi dengan sumber daya tersebut (Buller, Cerling, Enstall, & Ruiz, 2010). Virtualisasi

menciptakan versi virtual, bukan fisik, pada sistem operasi komputer, sumber daya jaringan komputer, atau perangkat penyimpanan. Dengan virtualisasi, pengguna dapat menggunakan perangkat lunak untuk mengubah satu komputer fisik menjadi beberapa komputer virtual yang lebih bermanfaat.

Menurut Alan Murphy (2016) dalam papernya *Virtualization Defined – Eight Different Ways*, menyebutkan setidaknya terdapat delapan istilah dalam penerapan virtualisasi. Macam-macam virtualisasi itu adalah *operating system virtualization, application server virtualization, application virtualization, management virtualization, network virtualization, hardware virtualization, storage virtualization dan service virtualization*.

Salah satu manfaat utama virtualisasi adalah bahwa fitur ini dapat memangkas biaya TI perusahaan (Buller, Cerling, Enstall, & Ruiz, 2010). Misalnya, perusahaan yang memiliki beberapa *server* dapat menggunakan virtualisasi *server* untuk mengurangi jumlah *server* fisik, sehingga mengurangi biaya pemeliharaan dan keamanan. Fitur ini juga memungkinkan perusahaan mengoptimalkan investasi perangkat keras yang ada daripada membeli *PC*, laptop, maupun *server* baru.

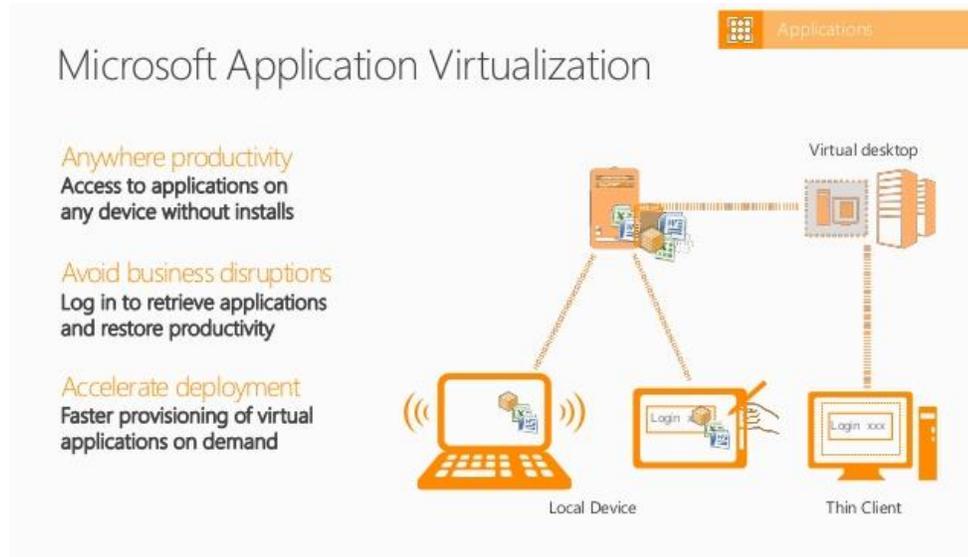
Pengguna pun dapat meningkatkan pemanfaatan perangkat keras yang telah dimiliki (Buller, Cerling, Enstall, & Ruiz, 2010). Pengguna juga dapat dengan mudah mengatur sistem untuk pemulihan gangguan, tanpa harus menambah perangkat keras lainnya. Selain itu, keuntungan yang dapat diperoleh antara lain terjadinya kesinambungan jalannya organisasi/perusahaan, fleksibilitas, kesederhanaan, konsolidasi *server*, dan *recovery* yang cepat.

### 2.2.2. Virtualisasi Aplikasi

Virtualisasi ini memungkinkan aplikasi virtual berjalan secara lokal walaupun pada kenyataannya aplikasi hanya berada di *server* utama (Murphy, 2016). Tipe lain dari virtualisasi aplikasi meliputi *Microsoft Terminal Services* dan *browser-based applications* (Microsoft, Overview of Remote Desktop Web Access (RD Web Access), 2016). Untuk penelitian ini, penulis menggunakan teknologi dari *Microsoft* yaitu *App-V*.

*Microsoft* menyediakan *App-V* sebagai *tools* untuk menjalankan aplikasi virtual. Dengan *App-V*, aplikasi virtual berjalan secara mandiri di lingkungan virtual mereka sendiri pada *PC* pengguna. Hal ini menghilangkan konflik aplikasi dan memungkinkan suatu organisasi/perusahaan untuk mengurangi kompatibilitas aplikasi waktu pengujian, sehingga penyebaran aplikasi lebih cepat dan *update*. Aplikasi-aplikasi virtual dan pengaturan pengguna disimpan ketika pengguna *online* atau *offline*. *App-V* akan memberikan pengalaman yang konsisten dan akses yang dapat diandalkan untuk aplikasi dan data bisnis, terlepas dari lokasi pengguna (Microsoft, Microsoft Desktop Virtualization Overview, 2016).

Pada gambar 2.1 merupakan konsep virtualisasi aplikasi yang dikembangkan oleh *Microsoft*. Dari gambar tersebut dapat dijelaskan bahwa *local device* atau *PC client* akan terhubung dengan *server* pusat yang telah berisi berbagai macam aplikasi. Perangkat yang terhubung dan memiliki akses ke *server* harus melakukan *login* sebagai *user* terlebih dahulu sebelum dapat menampilkan aplikasi di *desktop* mereka secara virtual. Konsep virtualisasi inilah yang diimplementasikan dalam penelitian ini.



**Gambar 2. 1** Konsep Microsoft Application Virtualization (Slideshare, 2016)

### 2.2.3. *Windows Server 2012 R2*

*Windows Server 2012 R2* merupakan sistem operasi terbaru yang dirilis oleh *Microsoft* setelah *Windows Server 2008* dan *Windows Server 2012* (*Microsoft, Windows Server 2012 R2, 2016*). *Windows Server 2012* hanya diproduksi dalam arsitektur 64-bit. *Windows Server 2012* terdiri dari 4 edisi, yaitu *Windows Server 2012 Foundation*, *Windows Server 2012 Essential*, *Windows Server 2012 Standard* dan *Windows Server 2012 Datacenter*. Perbedaan antara satu edisi dengan edisi lainnya terletak pada lisensi, fitur dan harga.

Penelitian ini menggunakan *Windows Server 2012 R2 Datacenter* karena memiliki lisensi dan fitur paling lengkap dibandingkan versi lainnya. *Windows Server 2012 R2* ini akan berperan sebagai sistem operasi pada *server* pusat. Selain itu, sistem operasi ini juga digunakan pada *PC domain controller*.

Sistem operasi *Windows Server 2012 R2* tentu saja dapat melakukan berbagai fungsi-fungsi atau *task* untuk melayani *client* karena peran utamanya sebagai *server*. Fungsi-fungsi itu disebut dengan *roles*. *Roles* merupakan kombinasi dari beragam layanan yang dapat diinstal dan dijalankan pengguna menggunakan *Server Manager* ataupun *Windows Powershell*. *Roles* yang terdapat pada *Windows Server 2012 R2* berfungsi maksimal jika disandingkan dengan *Windows 8* dan *8.1*.

#### **2.2.4. *Hyper-V* atau *Hypervisor***

*Hyper-V* atau *Hypervisor* adalah salah satu *role* yang terdapat pada *Windows Server 2012 R2*. *Hypervisor* memungkinkan sebuah *server*/komputer fisik menjalankan beberapa sistem operasi secara bersamaan. Hal tersebut menunjukkan seolah-olah *user* memiliki banyak komputer/*server* yang bisa dijalankan sekaligus dalam *virtual machine*. *Hyper-V* disebut virtualisasi berbasis *Hypervisor*. *Hypervisor* bisa disebut perangkat lunak atau *firmware* yang membuat mesin virtual (Booth, et al., 2014).

Sebenarnya *Hyper-V* bukanlah *role* baru pada *Windows Server 2012 R2*, tapi sudah hadir sejak *Windows Server 2008*, *Windows Server 2008 R2* dan *Windows Server 2012*. Hingga saat ini, *Hyper-V* juga terdapat pada *Windows 8 Pro* dan *Windows 8 Enterprise*. *User* membutuhkan *processor* yang telah mendukung teknologi virtualisasi, yaitu *Intel-VT* atau *AMD-V*, agar dapat menjalankan *role* ini. *Hyper-V Manager* adalah *interface* yang bisa *user* gunakan untuk management *virtual machine* (Microsoft, *Hyper-V Overview*, 2016).

Penulis membutuhkan *Hyper-V* sebagai *virtual machine* untuk menjalankan *Windows Server 2012 R2*. Dalam prosesnya penulis menambahkan fitur *RemoteFX* untuk meningkatkan pengalaman pengguna dari sisi grafis. *RemoteApp* akan berjalan dengan

bantuan *Hyper-V* karena *RemoteFX* tidak dapat dibangun secara langsung pada sistem operasi *server* utama.

### **2.2.5. Active Directory Domain Services (AD DS)**

*Active Directory Domain Services* merupakan salah satu *role* utama yang ada pada *Windows Server 2012 R2*. *Role* ini merupakan ciri khas dari *Windows Server* yang tidak dimiliki oleh sistem operasi lain. *Active Directory Domain Services (AD DS)* adalah layanan yang terintegrasi ke dalam sistem operasi *Windows Server 2012 R2* tetapi tidak secara otomatis diinstal sebagai *default*. Pengguna dapat menjalankan dan memberhentikan *Active Directory* karena layanan ini berjalan sebagai *Windows* pada *background* (Booth, et al., 2014).

Jika *Windows Server 2012 R2* akan dijalankan dengan *domain controller*, baik pada *domain* yang sudah ada atau benar-benar baru, pengguna harus membuat *AD DS*, *Active Directory database* di *server*, dan komponen-komponen lainnya yang diperlukan agar layanan ini berfungsi dengan baik. *Domain* merupakan sekumpulan objek yang berbagi pada *database* sejenis. Sebagai contoh, *admin* akan membuat ‘Joe’ pada *Active Directory database* pusat yang terkoneksi dengan *workgroup* komputer 1 dan 2 pada *database domain*. Jika semua objek dikelola secara terpusat, *admin* tidak perlu terkoneksi atau berjalan ke setiap komputer untuk mengubah *password* pengguna (Booth, et al., 2014).

*AD DS* digunakan penulis untuk memberikan akses bagi pengguna untuk menjalankan aplikasi pada *RemoteApp*. Penulis juga bertindak sebagai *admin*. Hal ini memungkinkan penulis untuk membatasi perilaku pengguna ketika terhubung dengan *RemoteApp*. Pada tahap awal, *admin* memberikan *username* dan *password* kepada setiap

pengguna yang ingin menjalankan aplikasi virtual. Pengguna bertindak sebagai *user* dimana hanya dapat menggunakan aplikasi-aplikasi yang disediakan oleh *admin* tanpa bisa merubah hal tersebut.

#### **2.2.6. Remote Desktop Services (RDS)**

*Remote Desktop Services* merupakan salah satu *role* utama pada *Windows Server 2012 R2*. Untuk penelitian ini, *RDS* akan berperan sebagai penghubung antara *server* dan *PC-client*. *RDS* yang biasa disebut *Terminal Services*, akan membuat pengguna seolah-olah merasa dan melihat aplikasi berjalan pada sistem lokal atau komputer mereka sendiri. Saat pengguna menjalankan sistem, pergerakan kursor dan tombol akan dikirim ke *server*, kemudian barulah ditampilkan kembali ke *desktop* mereka (Booth, et al., 2014).

*Remote Desktop Services* di *Windows Server 2012 R2* menyediakan lingkungan eksekusi dimana beberapa *distributed user* dapat mengakses *Windows desktop environment* atau aplikasi yang berjalan pada *server* terpusat atau kelompok *server* (Buller, Cerling, Enstall, & Ruiz, 2010). Masing-masing sesi *desktop* pengguna atau aplikasi menggunakan *CPU* bersama dan memori dari *server*. Kemudian, dengan menggunakan *Remote Desktop Protocol (RDP)*, hasil grafis sesi tersebut ditransfer pada perangkat *client*.

Pada penelitian ini layanan *Remote Desktop* yang digunakan adalah *Remote Desktop Web Access*. Dengan layanan ini, sebelumnya disebut *Terminal Services Web Access (TS Web Access)*, memungkinkan *user* untuk mengakses *RemoteApp and Desktop Connections* melalui menu *Start* pada komputer yang berjalan di sistem operasi *Windows* atau *web browser* (Microsoft, Overview of Remote Desktop Web Access (RD Web Access), 2016). *RemoteApp and Desktop Connections* digunakan untuk menyediakan kustomisasi tampilan

program *RemoteApp* dan *virtual desktop* pada penelitian ini. Selain itu, *RD WA* yang juga termasuk *Remote Desktop Web Connection*, karena memungkinkan pengguna untuk terhubung secara *remote* dari *web browser* ke desktop dari beberapa komputer dimana mereka memiliki akses *Remote Desktop*.

### **2.2.7. *RemoteApp***

*RemoteApp* adalah cara untuk melakukan virtualisasi dan menyediakan aplikasi yang terpusat pada *host* (Microsoft, Microsoft Desktop Virtualization Overview, 2016). *RemoteApp* memungkinkan pengguna untuk mengakses program-program secara *remote* melalui *Remote Desktop Services* yang memunculkan mereka seolah-olah berjalan pada komputer *client* itu sendiri (Microsoft, Overview of RemoteApp, 2016). Program ini telah mengalami peningkatan pada *Windows Server 2008 R2* dari versi sebelumnya yang dinamai *Terminal Services RemoteApp* (Booth, et al., 2014).

Program *RemoteApp* berjalan pada *windows* mereka sendiri, dapat dimunculkan diantara beberapa monitor dan memiliki ikonnya sendiri pada menu *start* ataupun *taskbar* (Microsoft, Overview of RemoteApp, 2016). Berikut ini adalah beberapa cara untuk mengakses *RemoteApp*, yaitu:

1. Mengakses sebuah *link* ke program melalui *RemoteApp and Desktop Connections* menggunakan *Remote Desktop Web Access (RD Web Access)*.
2. Klik 2x pada *Remote Desktop Protocol (.rdp) file* yang telah dibuat dan didistribusikan oleh administrator.
3. Klik 2x ikon program pada *desktop* atau menu *Start* yang telah dibuat dan didistribusikan oleh admin dengan *Windows Installer (.msi)*.

4. Klik 2x pada *file* dimana *file name extension* telah dihubungkan dengan *RemoteApp*.

Program ini memerlukan persyaratan agar dapat dijalankan secara optimal yaitu *.NET Framework 4* dan sistem operasi *Windows*. Namun, tidak semua versi atau edisi *Windows* dapat menjalankan *RemoteApp* (Kimknight, 2016). Penulis memaparkan sistem-sistem operasi yang mendukung program *RemoteApp* pada tabel 2.1.

**Tabel 2. 1** Daftar Sistem Operasi yang Mendukung *RemoteApp* (Kimknight, 2016)

| <b>Sistem Operasi</b> | <b>Versi/Edisi</b> |
|-----------------------|--------------------|
| Windows XP            | Professional SP 3  |
| Windows 7             | Ultimate           |
|                       | Enterprise         |
|                       | Education          |
| Windows 8             | Ultimate           |
|                       | Enterprise         |
|                       | Education          |
| Windows 10            | Ultimate           |
|                       | Enterprise         |
|                       | Education          |
| Windows Server        | 2008 (R2)          |
|                       | 2012 (R2)          |

Dari keempat cara akses *RemoteApp* tersebut, penulis memastikan bahwa aplikasi dapat diakses hanya menggunakan *Remote Desktop Web Access* pada penelitian kali ini.

Hal ini disebabkan oleh *PC client* yang masih menggunakan *Windows XP Professional Service Pack 3*. Sistem operasi tersebut tidak memiliki fitur *RemoteApp and Desktop Connection* layaknya *Windows 7, 8, 8.1, dan 10*. Oleh karena itu, program *RemoteApp* hanya dapat diakses melalui *RD Web Access* untuk penelitian ini.

#### **2.2.8. RemoteFX**

*RemoteFX* merupakan bagian dari protokol untuk *Remote Desktop Protocol (RDP)* yang digunakan untuk menyediakan *virtual desktop* secara *remote* dalam jaringan LAN. Dengan fitur ini pengguna dapat memanfaatkan virtual grafis pada kualitas tertinggi (Rouse, 2016). *RemoteFX* berfungsi secara mandiri dan tidak tergantung pada *graphic stack* dan mampu melayani apapun yang ada di monitor. *RemoteFX* bekerja sebagai *virtualized graphic resources*, memungkinkan kita untuk membangun konten pada *host* berbasis *thin client* maupun *thick client* (Microsoft, Microsoft RemoteFX, 2016).

Dari beberapa penjelasan sebelumnya, *RemoteFX* cenderung lebih bermanfaat ketika digunakan pada *desktop*. Namun, pada penelitian ini *RemoteFX* juga memberikan pengaruh pada virtual aplikasi. Dengan fitur ini, aplikasi yang berjalan pada *RemoteApp* akan terasa lebih lancar ketika digunakan oleh pengguna. Hal tersebut karena *RemoteFX* berperan sebagai virtual *GPU* pada *server* (Microsoft, Microsoft RemoteFX, 2016). Oleh sebab itu, penulis menambahkan fitur ini guna memperkaya pengalaman pengguna dalam menggunakan aplikasi dengan grafis terbaik.