

BAB II LANDASAN TEORI

2.1 Tinjauan Pustaka

Untuk mendukung penelitian ini, Penulis menggunakan beberapa *Literature* yang berkaitan dengan judul dan pokok bahasan pada penelitian. Adapun *Literatur* yang dipergunakan dapat ditinjau pada Tabel 2.1.

Tabel 2. 1 Literature Review

No. <i>Literature</i>	Penulis, Tahun	Judul
<i>Literature 01</i>	(Afrizal, H., Prihanto, A. 2022)	Analisis Kebutuhan <i>Resource</i> Dan <i>Independensi</i> Antara Teknologi Single Server, Virtualisasi Dan <i>Container</i>
<i>Literature 02</i>	(Suryanto, S. A. 2018).	Pengembangan Mekanisme Akses <i>E-Learning</i> Berbasis Linux <i>Container</i>
<i>Literature 03</i>	(Prasetyo, S. E., & Benny, B. 2021)	Analisis Perbandingan Performa Virtualisasi Berbasis <i>Container</i> Dengan Virtualisasi Berbasis <i>Hypervisor</i>
<i>Literature 04</i>	(Ariadi, F. S., Iswahyudi, C., & Nurnawati, E. K. 2020)	Penerapan <i>Docker Container</i> Sebagai Teknologi Ramah <i>Skalabilitas</i> Dibanding Teknik Virtualisasi Untuk Membangun Website di <i>Ubuntu</i>
<i>Literature 05</i>	(Mubarok, H., & Saptono, H. 2019)	Analisis dan Perancangan Platform Virtualisasi Berbasis <i>Docker</i> .

2.1.1 Tinjauan Terhadap Literatur 01

Berdasarkan penelitian yang telah dilakukan oleh (Afrizal, H.,Prihanto, A.2022) dengan judul Analisis Kebutuhan *Resource* Dan *Independensi* Antara Teknologi *Single Server*, *Virtualisasi* Dan *Container*. Penulis mengangkat permasalahan mengenai menempatkan layanan di lingkungan terpisah yang disebut

wadah, sehingga dapat menggabungkan layanan dengan perpustakaan atau perangkat lunak lain yang diperlukan. Docker memiliki sifat *containerization* yang ringan dengan berbagai komponen dan fitur yang memudahkan pengembang untuk mengembangkan dan memantau kinerja aplikasi yang dibangunnya, memungkinkan mereka untuk mendistribusikan dan mengembangkan aplikasi dengan lebih cepat.

Docker dapat dikonfigurasi untuk membuat objek. Objek yang disertakan dalam *Docker* termasuk gambar, wadah, dan layanan. Untuk menyebarkan aplikasi Anda ke *platform Docker*, cukup buat file instruksi (bernama *docker file*) tanpa membuat perubahan yang tidak perlu pada infrastruktur aplikasi Anda. *Klien Docker* dan *daemon Docker* dapat berjalan di *host* yang sama dan berkomunikasi melalui permintaan API, soket, atau jaringan. Fungsi dari *klien Docker* adalah mengambil sekumpulan perintah dari pengguna dan mengirimkannya untuk diproses oleh *daemon Docker*.

2.1.2 Tinjauan Terhadap Literature 02

Pada penelitian yang dilakukan oleh (Suryanto, S. A. 2018) dengan judul, Pengembangan Mekanisme Akses *E-Learning* Berbasis *Linux Container*. Penulis pada mengangkat permasalahan menjalankan beberapa sistem operasi dan layanan lainnya secara bersamaan, dengan tujuan untuk mengoptimalkan penggunaan prosesor dan menghindari pemborosan energi listrik dan biaya *server* yang berlebihan. Teknologi virtualisasi memudahkan dalam mengelola dan mengamankan *server* fisik jika rusak, sehingga cepat dan tidak perlu diinstal ulang. Keuntungan lain dari teknologi *virtualisasi* adalah teknologi yang sangat kuat yang digunakan untuk menyederhanakan pengembangan dan pengujian perangkat lunak dan memungkinkan *integrasi server*, sebagai pemisahan perangkat lunak dan perangkat keras, telah terbukti dapat digunakan. *Virtualisasi* mesin memiliki beberapa kelemahan, termasuk pengelompokan *server web* yang lambat. Ini telah menciptakan berbagai teknik dan model untuk meningkatkan *efisiensi, skalabilitas, dan ketahanan*.

2.1.3 Tinjauan Terhadap Literature 03

Pada penelitian yang dilakukan oleh (Prasetyo, S. E., & Benny, B. 2021) yang berjudul Analisis Perbandingan *Performa Virtualisasi* Berbasis *Container*

Dengan *Virtualisasi* Berbasis *Hypervisor*. Penulis mengangkat masalah tentang dalam membangun server-server yang berbeda. Diantara virtualisasi yang ada, terdapat dua jenis virtualisasi yang mendominasi, yaitu *Hypervisor* dan *container*. *Hypervisor* adalah virtualisasi memungkinkan suatu mesin untuk menjalankan beberapa mesin virtual dengan sistem operasi yang berbeda, dan *container* adalah virtualisasi yang mengisolasi pada level *sistem operasi*. Terdiri dari arsitektur yang berbeda, performa dari kedua *virtualisasi* juga dianggap berbeda.

Untuk menemukan perbandingan performa dari kedua *virtualisasi* maka dibangun dua mesin berbasis *cloud* yang memiliki spesifikasi yang sama kemudian dijalankan *performance*. Test pada kedua mesin tersebut dengan menggunakan *Apache JMeter*. Variabel yang menjadi acuan pada *performance test* ini adalah *response time*, *deviation*, *throughput*, *error* dan *CPU utilization*, dengan dua jenis pengujian yang pertama dengan 200 user dan yang kedua dengan 500 user. Mesin yang menggunakan virtualisasi berbasis *Hypervisor* menghasilkan performa yang sama dengan mesin yang menggunakan virtualisasi berbasis *container* pada 200 *user*, namun memiliki penurunan performa yang signifikan ketika dihadapi dengan 500 user dibandingkan dengan mesin yang menggunakan *virtualisasi* berbasis *container*.

2.1.4 Tinjauan Terhadap Literature 04

Pada penelitian yang dilakukan oleh (Ariadi, F. S., Iswahyudi, C., & Nurnawati, E. K. 2020) yang berjudul Penerapan Docker Container Sebagai Teknologi Ramah Skalabilitas Dibanding Teknik Virtualisasi Untuk Membangun Website di Ubuntu. Penulis mengangkat masalah tentang Penelitian tentang bagaimana menerapkan Docker Container untuk mengatasi kendala pengembangan aplikasi, Docker diterapkan untuk membangun Aplikasi Chatting Berbasis Web. Sistem informasi dalam sebuah perusahaan bersifat dinamis (sesuai perkembangan bisnisnya), sehingga menjadi masalah bagi *developer* dan *administrator* sebuah sistem aplikasi, terutama aplikasi berbasis web. *Developer* harus mempunyai aplikasi yang mampu berjalan dalam lingkungan yang berbeda untuk menjawab permasalahan seperti saat melakukan development dan testing pada environment yang berbeda atau *migrasi server*.

Kendala yang sering terjadi adalah *conflicting dependencies*, *missing dependencies* dan perbedaan *platform server* itu sendiri. Maka dari itu digunakan lah teknologi *Docker Container* dalam penelitian tersebut, dimana cara kerja dari *Docker Container* mencakup penempatan aplikasi dalam sebuah *Container* di lingkungan operasinya sendiri, berbeda dengan virtual mesin yang diterapkan di *server* pada umumnya. dilakukan implementasi & analisis performansi layanan web pada platform berbasis *Docker*, karena dalam proses pengembangan layanan web dibutuhkan beberapa aplikasi yang saling terintegrasi (*front-end* dan *back-end development*). Pada arsitektur monolitik, penyedia layanan membangun sebuah aplikasi yang seluruh komponennya terbuat menjadi sebuah kesatuan lalu disimpan dalam server.

Jika penyedia layanan web ingin melakukan *scaling system* maka harus melakukan perombakan sistem secara menyeluruh karena penambahan line coding sekecil apapun akan memengaruhi keseluruhan aplikasi sehingga proses development sangat beresiko. Maka solusi dari arsitektur monolitik yang kurang efektif tersebut yaitu dengan melakukan transisi ke arsitektur *microservice*, aplikasi dipecah menjadi beberapa komponen yang dikembangkan dan berjalan secara independen. Untuk mengembangkan *microservice application* terdapat sebuah platform yang mendukung yaitu *Docker*, dimana satu *server* bisa digunakan untuk menjalankan banyak aplikasi tanpa mengintervensi kinerja aplikasi lainnya.

2.1.5 Tinjauan Terhadap Literature 05

Pada penelitian yang di lakukan oleh (Oktivasari, Habibullah, 2017) penulis membahas tentang Analisis dan Perancangan *Platform Virtualisasi* Berbasis *Docker*, maksud dari penelitian ini adalah untuk kebutuhan pengembangan aplikasi. Yang membutuhkan servis yang beragam dan berbeda-beda dengan lingkungan yang berbeda juga, maka dibutuhkan adanya virtualisasi dengan berbasis *Docker* agar aplikasi dapat dijalankan di berbagai *platform* yang berbeda sehingga dapat memudahkan pengembang aplikasi dan *sistem administrator*. Dalam membangun dan menjalankan aplikasi yang telah dirancang, yaitu dengan menjalankan *servis* yang dibutuhkan dalam sebuah *kontainer*.

Penulisan ilmiah ini bermaksud menganalisis dan merancang sebuah platform virtualisasi berbasis *Docker* yang dipasang di *operasi system Linux Ubuntu 17.10*, meliputi proses instalasi, konfigurasi, dan menjalankan aplikasi *Docker*. *Docker* adalah sebuah projek terbuka yang berfungsi membantu cara kerja

untuk *developer* atau sistem administrator untuk membangun, mengemas dan menjalankan aplikasi dimanapun dalam sebuah kontainer. Arsitektur menggunakan metode *klien server*, *Docker klien*, mengirimkan permintaan ke *Docker daemon* untuk membangun, mendistribusikan dan menjalankan kontainer *Docker*.

2.1 Keaslian Peneliti

Adapun beberapa hal yang menjadi pembeda antara penelitian yang dilakukan penulis dengan penelitian yang sudah dilakukan sebelumnya sebagaimana terlampir di tabel tinjauan pustaka, antara lain adalah:

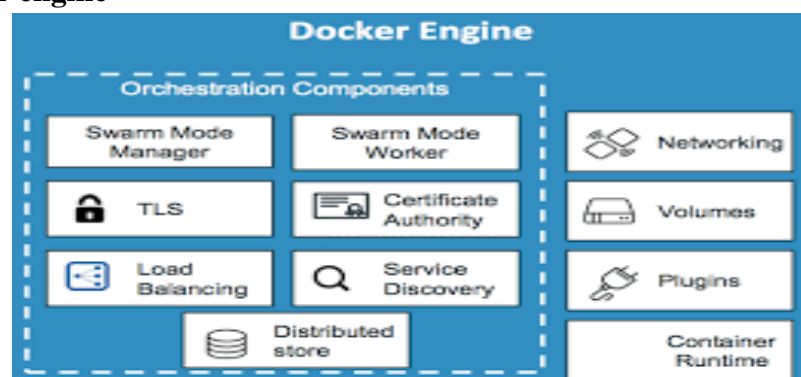
- 1) Penelitian ini berfokus pada analisis perbandingan performa Virtualisasi antara *Docker* dan *VirtualBox* pada website *Learning Management System*.
- 2) Penelitian ini pertama kali dilakukan dengan membandingkan performa antara *Docker* dan *VirtualBox* pada website *LMS Moodle*.

2.2 Docker

Docker adalah layanan yang menyediakan kemampuan untuk mengemas dan menjalankan sebuah aplikasi dalam sebuah lingkungan terisolasi yang disebut dengan *container*. Dengan adanya isolasi dan keamanan yang memadai memungkinkan kamu untuk menjalankan banyak *container* di waktu yang bersamaan pada *host* tertentu. (Adiputra, F. 2015).

Berikut adalah fitur-fitur dari *Docker* :

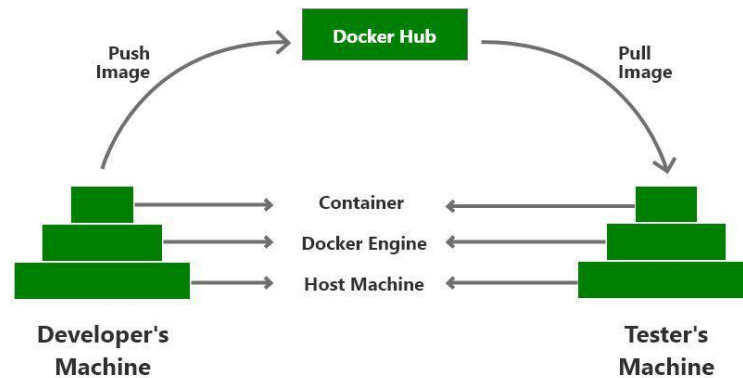
- Docker engine



Gambar 2. 1 *Docker engine*

Docker engine digunakan untuk membuat image dan container. diinstal pada setiap *server* ingin menjalankan *kontainer* dan menyediakan sekumpulan perintah sederhana yang dapat digunakan untuk membuat, memulai, atau menghentikan *kontainer*.

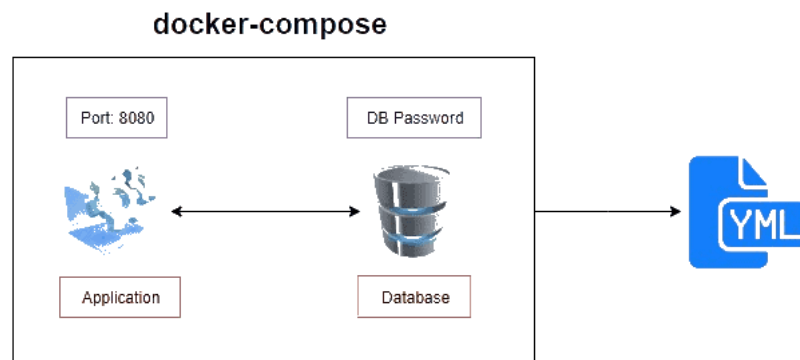
- Docker Hub



Gambar 2. 2 docker Hub

Docker Hub adalah *registry* yang berisikan kumpulan dari image-image. Dengan menggunakan *docker hub* ini dapat mengumpulkan *image*. *Hub* ini berbeda dengan *docker engine* yang hanya membuat image.

- Docker Compose



Gambar 2. 3 Docker Compose

Docker compose adalah salah satu fitur unggulan yang berfungsi untuk menjalankan beberapa *container* atau biasa disebut *multi-container* sehingga dapat menghemat banyak waktu

2.3 VirtualBox



Gambar 2. 4 VirtualBox

VirtualBox merupakan perangkat lunak virtualisasi untuk menginstal sistem operasi “*Operating System*”. Kata virtualisasi yaitu mengubah atau mengkonversi sesuatu menjadi bentuk simulasi dari bentuk real atau nyata. Misalnya, jika seseorang telah *menginstal sistem operasi Windows* di komputer mereka, orang ini juga dapat menjalankan sistem operasi lain yang diinginkan dalam sistem operasi Windows. Bagi Anda yang ingin mencoba berlatih menginstal sistem operasi, tidak perlu menginstal ulang PC / laptop. Hanya memerlukan perangkat lunak *VirtualBox* ini untuk mencoba atau belajar menginstal sistem operasi. (Sriyanta, S. 2018).

2.4 LMS Moodle



Gambar 2. 5 LMS MOODLE

Moodle adalah platform bersifat web-based yang biasa digunakan untuk keperluan e-learning. Dengan kata lain, Moodle dibuat khusus sebagai sebuah sistem aplikasi pembelajaran online yang efektif. Kepanjangan dari *Moodle* adalah *Modular Object-Oriented Dynamic Learning Environment*. Artinya, platform ini dikhususkan untuk membuat aplikasi belajar *online* yang dinamis. (Samsudin 2019).

2.5 Ngrok



Gambar 2. 6 NGROK

Ngrok adalah aplikasi lintas platform yang dapat membuat URL peneropongan atau penerusan, sehingga permintaan internet menjangkau komputer lokal. Gunakan *ngrok* untuk meneruskan pesan dari saluran eksternal di web langsung ke komputer lokal. (suryani 2019).

2.6 Metode Experiment

Metode eksperimen memberikan kesempatan kepada individu dan kelompok untuk secara sadar merancang dan merencanakan eksperimen untuk membuktikan kebenaran teori dengan mengikuti dan menerapkan rute yang teratur dan sistematis. (Rismawati, Ratman dan Dewi, 2006). Penelitian eksperimen dilakukan untuk mengetahui pengaruh pemberian suatu treatment atau perlakuan terhadap subjek penelitian (Cahya, 2013). Selain itu, metode eksperimen dilakukan untuk menghasilkan suatu data yang diperlukan dengan parameter yang sudah ditentukan. Metode eksperimen dilakukan dengan cara melakukan pengetestan langsung pada media (Hasim dan Riadi, 2013).

Karakteristik dari metode eksperimen menurut (Rismawati, Ratman dan Dewi, 2006) sebagai berikut :

- 1) Metode untuk melakukan percobaan, pengamatan dan penarikan kesimpulan terhadap sesuatu yang sedang diuji.
- 2) Metode yang dirancang untuk mengembangkan pengetahuan.
- 3) Metode yang membantu dalam pemrosesan informasi yang aktif.
- 4) Metode yang mengarahkan untuk mempelajari lingkungan belajar sebagai suatu teknologi.
- 5) Metode yang digunakan untuk memecahkan masalah yang bersifat ilmiah.

2.7 Metode Pengujian

Pada tahap ini, Pengujian yang dilakukan pada penelitian ini menggunakan teknik kuantitatif berdasarkan parameter *QoS*. Penelitian kuantitatif adalah penelitian yang membutuhkan penggunaan nilai numerik, mulai dari pengumpulan data dan interpretasi data hingga penyajian hasil. Demikian pula, di akhir pelajaran, Anda mungkin ingin melampirkan foto, tabel, atau gambar lainnya. (Panjaitan, 2019).

