BAB 2

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Dalam mengimplementasikan LAMP *stack* dan email server dengan menggunakan Docker, sebagai pedoman dan pembanding maka digunakan beberapa pustaka yang berkaitan dengan *Docker*. Pustaka yang digunakan sebagai rujukan antara lain:

Tabel 2.1: Tabel Pustaka

Penulis	**Topik**	**Hasil Penelitian**	**Tekno-logi yang diguna-kan**
Wijayanto (2018)	'Tehnologi Platfrom Virtualisasi Untuk Aplikasi Lowongan Kerja'	Menghasilkan aplikasi lowongan kerja yang dibuat menjadi docker images yang dapat berjalan di container docker sehingga dapat dimanfaatkan di penyedia *platform as a service* (PAAS).	Docker
Dwiyatno (2020)	'Implementasi virtualisasi server berbasis docker container'	Mengimplementasikan sistem virtualisasi server berbasis docker container di SMK Negeri 1 Rangkasbitung dan pengujian penggunaan sumber daya dari penggunaan Docker sebagai sistemt virtualisasi	Docker, Nginx, Apache Web Server, Apache Jmeter
Megantara et al. (2021)	'IMPLEMENTASI DOCKER DALAM MEMBANTU VIRTUAL LAB PADA UNIVERSITAS'	Menghasilkan sistem untuk membantu mahasiswa melakukan praktek pemrograman pada saat pembelajaran jarak jauh. Dengan sistem ini mahasiswa dapat mengerjakan tugas di beragam perangkat dan memudahkan dosen memeriksa hasil praktek.	Docker, Jupyter Notebook
Prasetyo dan Wijaya (2021)	'Analisa dan Implementasi Microservice pada Container Menggunakan Docker'	Analisa performa microservice yang diimplementasikan di docker container. Dalam penelitian ini disimpulkan docker memiliki efisiensi yang lebih baik dibanding virtualisasi hypervisor.	Docker, Apache Jmeter

Penulis	Topik	Hasil Penelitian	Teknologi yang digunakan
Furnama (2022)	'Implementasi Arsitektur Microservices Pada Sistem Backend Pembayaran Terintegrasi Menggunakan Docker'	Menghasilkan sistem *backend* pembayaran terintegrasi Midtrans dengan arsitektur microservice yang berjalan di atas docker.	Docker, Midtrans
Saputra dan Rismayadi (2021)	'IMPLEMENTASI CLOUD STORAGE MENGGUNAKAN OWNCLUOD DAN DOCKER'	Penelitian ini menghasilkan sistem cloud storage menggunakan OwnCloud dan Docker. Dengan menggunakan Docker, pemasangan OwnCloud tidak mempengaruhi sistem *host*, selain itu Docker juga menghemat penggunaan resource.	Docker, OwnCloud
Kristiawan, Wahanani dan Idhom (2022)	'Implementasi Reverse Proxy Pada Hosting Web Server di Docker Container'	Pada penelitian ini menghasilkan sistem web hosting dengan menggunakan Wordpress sebagai CMS yang memanfaatkan nginx sebagai web servernya, sedangkan untuk reverse proxy digunakan Apache Web Server.	Docker, Wordpress, Nginx
Aldzikri Dwijayanto Prathama (Diusulkan)	Implementasi Lamp Stack Dan Email Server Menggunakan Docker di PT.Pundi Mas Berjaya (Magang Bersertifikat Kampus Merdeka)	Pada penelitian ini adalah membahas cara membangun sebuah server LAMP dan email, dengan metode *containerization* menggunakan Docker. Sehingga memudahkan manajemen, dan *deployment* aplikasi.	Docker, LAMPstack, Postfix, Dovecot

2.2 Dasar Teori

2.2.1 PT.Pundi Mas Berjaya

PT.Pundi Mas Berjaya adalah salah satu penyedia solusi perangkat lunak di pasar global yang memberikan solusi bisnis mengadopsi teknologi informasi terkini. Solusi berbasis layanan untuk para pelanggan yang tersebar di beberapa negara dan memiliki model pengembangan on-site dan off-site. Sejak berdiri pada tahun 2014, Perusahaan telah merancang, mengembangkan dan digunakan banyak solusi di bidang properti, otomotif, transportasi, pengirimanan makanan, pengiriman barang dan ecommerce.

PT.Pundi Mas Berjaya memiliki tim yang kuat dengan ukuran pelaksanaan proyek besar dan mengerahkan seluruh kemampuannya untuk memberikan pelayanan yang berkualitas tinggi kepada pelanggan. Salah satu kunci keberhasilan PT.Pundi Mas Berjaya adalah adaptasi perusahaan tersebut terhadap berbagai macam dan aneka kebutuhan akan sistem informasi komputer sehingga menghasilkan Manajemen Kualitas, Manajemen Proyek, kebutuhan Infrastruktur, dan lainnya dengan sempurna untuk memberikan kepuasan kepada para pelanggan PT.Pundi Mas Berjaya. Pelayanan PT.Pundi Mas Berjaya sifatnya terpadu menggunakan media responsif website, mobile website, mobile apps untuk mendukung aktifitas yang lebih optimal.

PT.Pundi Mas Berjaya memiliki tim berpengalaman di bidang:

1. Manajemen teknis website
2. Manajemen teknis mobile application
3. Manajemen bisnis untuk aplikasi yang sedang dibangun
4. Manajemen kendali mutu untuk aplikasi yang dijalankan

Sedangkan fokus bisnis dari perusahaan ini antara lain:

1. Aplikasi properti online kolaborasi dalam dan luar negeri
2. Aplikasi transportasi berbasis aplikasi
3. Aplikasi bank data untuk pebisnis
4. Aplikasi pelayanan pengantaran makanan
5. Aplikasi pelayanan pengiriman bunga/parcel
6. Aplikasi Deal untuk aneka voucher
7. Aplikasi e-commerce

2.2.2 Container

Menurut Shah dan Dubaria (2019) *Containerization* adalah cara untuk menjalankan beberapa perangkat lunak yang berbeda di satu mesin. Yang masing-masingnya berjalan di dalam lingkungan yang terisolasi yang disebut sebagai *container*. *Container* adalah lingkungan yang terisosali untuk perangkat lunak. Container mengikat semua berkas dan *library* yang dibutuhkan oleh aplikasi untuk berfungsi dengan baik.

2.2.3 Docker

Docker adalah sebuah project *open-source* yang menyediakan platform terbuka untuk developer maupun sysadmin untuk dapat membangun, mengemas, dan menjalankan aplikasi dimanapun di dalam sebuah container(Docker 2023).

Docker menggunakan arsitektur client-server. Dimana client dan docker berkomunikasi dengan daemon docker, yang melakukan suatu tindakan untuk membangun, menjalankan, dan mendistribusikan container docker. Client docker dan daemon dapat berjalan pada sistem yang sama. Client docker dan daemon berkomunikasi menggunakan REST API, melalui soket UNIX atau antarmuka jaringan(Docker 2023).

2.2.4 Docker Compose

Docker Compose adalah alat untuk mendefinisikan dan menjalankan aplikasi Docker multi-kontainer. Dengan Docker Compose, file YAML bisa digunakan untuk mengonfigurasi layanan aplikasi. Kemudian, hanya dengan satu perintah, Docker akan membuat dan memulai semua layanan dari konfigurasi YAML tersebut.

2.2.5 LAMP *stack*

LAMP *stack* adalah gabungan dari empat teknologi perangkat lunak berbeda yang digunakan oleh developer untuk membangun situs web dan aplikasi web. LAMP adalah singkatan dari sistem operasi Linux; server web Apache; server basis data

MySQL; dan bahasa pemrograman PHP. Keempat teknologi ini bersifat *open-source*, yang berarti LAMP dikelola komunitas dan tersedia secara bebas untuk digunakan oleh siapa saja. Developer menggunakan LAMP *stack* untuk membuat, meng-host, dan memelihara konten web. LAMP *stack* adalah solusi populer yang mendukung banyak situs web yang biasa gunakan saat ini(Amazon 2023).

2.2.6 Email Server

Email server adalah perangkat lunak yang berfungsi untuk mengirim dan menerima email. Email server biasanya menjadi seebutan untuk *Mail Transfer Agent* (MTA) dan *Mail Delivery Agent* (MDA), yang keduanya memiliki fungsi yang berbeda(Cloudflare 2023).

Pesan email dikirim dan diterima dengan menggunakan dua jenis email server: *outgoing mail server* atau *Mail Transfer Agent*, dan *incoming mail server* atau *Mail Delivery Agent*. MTA akan menerima pesan yang keluar dari *email client* pengirim, kemudian mengirimkannya ke MDA, yang mana bertanggung jawab untuk menyimpan sementara dan menyampaikan pesan ke *email client* penerima(Cloudflare 2023).

2.2.7 DNS TXT Record

TXT Record, Merupakan sebuah record DNS yang digunakan untuk menyimpan informasi berupa text pada domain. Teks pada TXT Records harus diformat sesuai dengan teknologi yang akan gunakan(Rosenbaum 1993).

2.2.8 DKIM

Merujuk pada dokumen RFC 6376, DKIM *(DomainKeys Identified Mail)* adalah teknik untuk memverifikasi keaslian pesan email dengan memeriksa tanda tangan digital yang disematkan di header pesan. Tanda tangan ini dihasilkan dengan

mengenkripsi hash konten pesan dengan *private key*, yang dapat diverifikasi oleh penerima menggunakan *public key* yang diterbitkan dalam data DNS TXT domain pengirim(Kucherawy, Crocker dan Hansen 2011).

Dengan DKIM email server penerima dapat memverifikasi bahwa pesan yang masuk tidak diubah saat transit dan berasal dari domain yang digunakan. DKIM *signature* dapat dibuat dengan menambahkan header ke pesan yang berisi informasi tentang pesan dan tanda tangan kriptografi dari isi pesan dan beberapa header. Server email penerima kemudian dapat menggunakan informasi di header DKIM dan kunci publik yang dipublikasikan di DNS domain pengirim untuk memverifikasi keaslian dan integritas pesan(Kucherawy, Crocker dan Hansen 2011).

2.2.9 SPF

Menurut dokumen RFC 7208, SPF *(Sender Policy Framework)* adalah protokol autentikasi email yang dirancang untuk mendeteksi dan mencegah spoofing email dengan cara memverifikasi identitas domain yang diklaim berasal dari email. SPF bekerja dengan memberikan daftar alamat IP atau nama host pengiriman resmi untuk domain di DNS, dan server email penerima dapat menggunakan informasi tersebut untuk memvalidasi keaslian pesan masuk(Kitterman 2014).

Saat email diterima, server email penerima memeriksa data SPF untuk domain pada header ”From” pesan untuk menentukan apakah pengirim diizinkan untuk mengirim email atas nama domain yang digunakan. Jika server pengirim tidak tercantum dalam catatan SPF, pesan tersebut mungkin akan ditandai sebagai spam atau bahkan langsung ditolak(Kitterman 2014).

2.2.10 DMARC

Menurut dokumen RFC 7489, DMARC *(Domain-based Message Authentication, Reporting & Conformance)* merupakan metode autentikasi, pengaturan dan pelaporan yang menggunakan kombinasi metode autentikasi SPF (Sender Policy Framework) dan DKIM (DomainKeys Identified Mail) untuk menentukan apakah pesan email itu sah. Dengan menggunakan DMARC, email server penerima dapat mengambil tindakan yang sesuai, seperti menolak atau mengkarantina pesan, berdasarkan peraturan yang dipublikasikan pemilik domain di DMARC-nya(Kucherawy dan Zwicky 2015).

DMARC juga memiliki mekanisme pelaporan bagi pemilik domain untuk menerima umpan balik tentang pesan yang lulus atau gagal dalam evaluasi DMARC, dan membantu pemilik domain memantau pengiriman email dan mendeteksi potensi terjadinya spoofing email(Kucherawy dan Zwicky 2015).