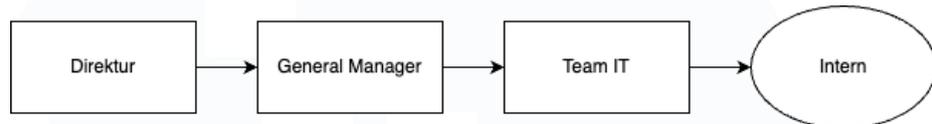


BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

3.1.1 Kedudukan

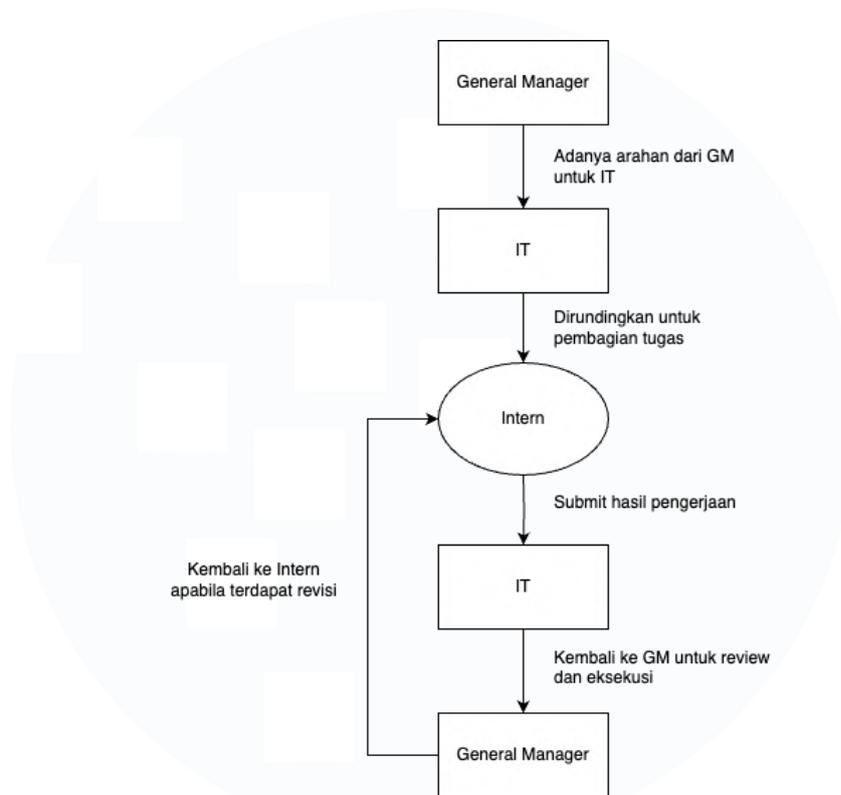


Gambar 3.1 Kedudukan pada PT Maro Anugrah Jaya

Dalam struktur organisasi PT Maro Anugrah Jaya, posisi pekerja magang (Intern) berada pada tingkat paling kanan atau di bawah divisi IT. Struktur hierarki ini menunjukkan jalur pelaporan formal dalam organisasi, dimulai dari Direktur sebagai pemegang otoritas tertinggi, turun melalui General Manager, Manager, dan divisi IT, hingga ke posisi Intern.

Dalam hierarki ini, divisi IT berperan sebagai kelompok pembimbing utama yang memberikan pengarahan serta menerima pelaporan aktivitas dari pekerja magang. Setiap permasalahan atau tugas yang dihadapi oleh pekerja magang akan dikoordinasikan terlebih dahulu dengan divisi IT untuk mendapatkan solusi atau arahan yang sesuai.

3.1.2 Alur Koordinasi



Gambar 3.2 Alur Koordinasi Pengerjaan Penugasan

Proses koordinasi berlangsung sebagai berikut:

1. Inisiasi Tugas

General Manager menginisiasi proyek atau pekerjaan yang perlu dilakukan. Pada tahap ini, General Manager memberikan arahan dan ekspektasi dari tugas yang akan dikerjakan.

2. Delegasi ke Tim IT

Tim IT menerima arahan dari General Manager dan menganalisis *requirement* teknis dari tugas tersebut. Arahan tersebut kemudian di-*breakdown* menjadi tahapan-tahapan yang lebih spesifik.

3. Penugasan ke *Intern*

Tim IT mendelegasikan tugas kepada *Intern* disertai dengan penjelasan teknis yang diperlukan. *Intern* menerima brief detail mengenai *scope* pekerjaan dan target yang harus dicapai.

4. Pengerjaan oleh *Intern*

Intern mengerjakan tugas sesuai dengan arahan yang diberikan. Selama proses pengerjaan, *Intern* dapat berkonsultasi dengan tim IT untuk mendapatkan bantuan teknis atau klarifikasi.

5. Review oleh Tim IT

Hasil pekerjaan *Intern* di-review oleh tim IT untuk memastikan kualitas teknis dan kesesuaian dengan *requirement* yang diberikan. Tim IT dapat memberikan *feedback* untuk perbaikan jika diperlukan.

6. *Final Review* oleh *General Manager*

Setelah melewati *review* tim IT, hasil pekerjaan diserahkan kepada *General Manager* untuk final review. *General Manager* akan mengevaluasi apakah hasil pekerjaan sudah sesuai dengan ekspektasi awal.

7. Siklus Umpan Balik

Jika hasil *review* memerlukan revisi, proses akan kembali ke tahap pengerjaan oleh *Intern* dengan *feedback* yang lebih spesifik.

3.2 Tugas dan Uraian Kerja Magang

Tabel 3.1 Tugas dan Uraian Kerja Magang Minggu 1-2

Minggu ke-	Pekerjaan
1	Perkenalan dengan tim IT dan lingkungan kerja infrastruktur. Mempelajari topologi jaringan yang ada di perusahaan dan menganalisis kebutuhan server. Melakukan studi literatur mengenai virtualisasi menggunakan Proxmox VE dan Docker. Membuat dokumentasi awal tentang kebutuhan infrastruktur dan mengidentifikasi layanan yang akan dimigrasi.
2	Melakukan analisis kebutuhan perangkat keras untuk server Proxmox. Membantu dengan perancangan proposal awal infrastruktur khususnya spesifikasi komponen. Merencanakan tata letak ruang server dan kebutuhan <i>networking</i> , termasuk konfigurasi IP statis dan persiapan jaringan untuk server baru.

Tabel 3.2 Tugas dan Uraian Kerja Magang Minggu 3-8

Minggu ke-	Pekerjaan
3	Perakitan fisik server Proxmox dan instalasi sistem dasar. Konfigurasi Proxmox VE termasuk setting RAID, <i>networking</i> , dan <i>storage pools</i> . Implementasi layanan infrastruktur utama seperti Cloudflare Tunneling, AdGuard DNS, dan sistem <i>monitoring</i> (Prometheus, Grafana). Konfigurasi awal kontainer database (MariaDB, Redis) untuk mendukung aplikasi yang akan di- <i>deploy</i> .
4	Konfigurasi kontainer LXC untuk berbagai layanan. Implementasi keamanan dasar pada server Proxmox termasuk <i>firewall</i> dan <i>secure shell</i> (SSH) <i>hardening</i> . Migrasi website perusahaan dari <i>hosting</i> eksternal ke infrastruktur server baru. Konfigurasi sistem pencadangan otomatis untuk kontainer dan <i>virtual machine</i> .
5	Implementasi dan konfigurasi Nextcloud sebagai solusi file sharing perusahaan. Membangun environment pengembangan untuk aplikasi ERPNext di server Proxmox. Optimalisasi performa dan resource allocation pada kontainer dan <i>virtual machine</i> . Konfigurasi monitoring detail untuk memantau kinerja server.
6	Implementasi solusi manajemen aset IT menggunakan Snipe-IT pada kontainer khusus. Konfigurasi ZFS <i>storage pools</i> untuk pengelolaan data yang lebih efisien. Optimalisasi alokasi IP statis untuk seluruh kontainer dan VM. Dokumentasi lengkap infrastruktur server untuk keperluan manajemen dan <i>transfer knowledge</i> .
7	Deployment dan konfigurasi ERPNext untuk mendukung operasional perusahaan. Implementasi solusi database redundan dengan replikasi MariaDB untuk <i>high availability</i> . Konfigurasi GPU <i>passthrough</i> untuk virtual machine Windows yang akan digunakan untuk <i>rendering</i> . Manajemen <i>user</i> dan <i>access control</i> di seluruh layanan infrastruktur.
8	Implementasi server printing menggunakan CUPS dan Samba pada infrastruktur Proxmox. Konfigurasi lanjutan untuk keamanan server termasuk implementasi <i>Time-based One-Time Password</i> (TOTP). Konfigurasi detail dashboard monitoring Grafana untuk pemantauan performa server dan aplikasi.

Tabel 3.3 Tugas dan Uraian Kerja Magang Minggu 9-14

Minggu ke-	Pekerjaan
9	Optimalisasi performa ERPNext pada infrastruktur server. Penyempurnaan sistem <i>caching</i> menggunakan Redis untuk meningkatkan responsivitas aplikasi.
10	Implementasi keamanan lanjutan pada server Proxmox dengan Fail2Ban dan sistem autentikasi dua faktor (2FA). Konfigurasi VPN server untuk akses aman ke infrastruktur dari luar jaringan. Optimalisasi performa database dengan <i>fine-tuning parameter</i> MariaDB. Implementasi sistem monitoring lanjutan dengan <i>alert notification</i> melalui email dan Telegram.
11	Pengembangan layanan microservice dalam container. Optimalisasi CPU dan <i>memory resource limits</i> pada kontainer untuk mencegah <i>resource starvation</i> .
12	Implementasi infrastruktur testing untuk continuous integration aplikasi <i>microservice</i> pada server Proxmox. Optimalisasi storage dengan implementasi <i>cache tier</i> menggunakan SSD. Pembuatan dokumentasi teknis lengkap untuk manajemen dan troubleshooting infrastruktur.
13	Optimalisasi infrastruktur server untuk mendukung workload rendering dengan konfigurasi lanjutan GPU passthrough. Implementasi solusi monitoring tambahan untuk memantau performa GPU pada virtual machine. Konfigurasi backup offsite menggunakan <i>remote storage</i> . Implementasi <i>disaster recovery plan</i> dan pengujian prosedur <i>restore</i> dari <i>backup</i> .
14	Perluasan kapasitas infrastruktur server dengan penambahan storage pool untuk kebutuhan aplikasi baru. Migrasi dan konfigurasi layanan tambahan ke infrastruktur Proxmox. Implementasi container orchestration untuk manajemen <i>microservices</i> yang lebih efisien. Konfigurasi SSL/TLS untuk semua <i>endpoint</i> aplikasi dengan implementasi <i>auto-renewal certificate</i> .

Tabel 3.4 Tugas dan Uraian Kerja Magang Minggu 15

Minggu ke-	Pekerjaan
15	Deployment aplikasi web menggunakan NextJs dan Framer Motion pada infrastruktur Proxmox. Konfigurasi reverse proxy dan SSL untuk aplikasi web. Implementasi integrasi dengan layanan eksternal melalui API yang aman. Finalisasi dokumentasi teknis infrastruktur server dan knowledge transfer kepada tim IT internal. Evaluasi performa keseluruhan infrastruktur dan penyusunan rekomendasi untuk pengembangan server di masa mendatang.

Implementasi dan pengembangan infrastruktur server menggunakan Proxmox dilaksanakan secara intensif mulai minggu ke-3 hingga minggu ke-10 masa kerja magang. Proxmox VE (Virtual Environment) adalah platform virtualisasi *open-source* berbasis Debian Linux yang mengintegrasikan Kernel-Based Virtual Machine (KVM) *hypervisor* dan Linux Container (LXC), memungkinkan manajemen terpusat untuk virtualisasi server, penyimpanan, dan jaringan melalui antarmuka web [1]. Tahap awal dimulai dengan perakitan server fisik dan instalasi Proxmox sebagai platform virtualisasi utama, dilanjutkan dengan konfigurasi *Redundant Array of Independent Disks* (RAID) tipe 1. RAID 1 mengimplementasikan pencerminan *disk* untuk redundansi data [2]. Dalam aspek keamanan dan *networking*, dilakukan implementasi kontainer Cloudflare untuk *tunneling* dan Adguard sebagai *DNS filtering*, serta perancangan *static IP* untuk memudahkan manajemen jaringan. Cloudflare Tunnel menciptakan koneksi terenkripsi yang aman antara server dan jaringan Cloudflare. Layanan yang memungkinkan ekspos aplikasi ke internet secara aman tanpa memerlukan IP publik atau *port forwarding*, menggunakan koneksi *outbound* terenkripsi melalui jaringan Cloudflare [3].

Pengembangan infrastruktur berlanjut dengan pemasangan sistem monitoring menggunakan Prometheus dan Grafana untuk pemantauan performa server. Prometheus adalah sistem *monitoring* dan *alerting toolkit open-source* yang dirancang untuk keandalan tinggi [4], sedangkan Grafana

menyediakan platform visualisasi dan analitik yang memungkinkan query, visualisasi, dan pemahaman metrik [5]. Implementasi *database layer* dilakukan melalui kontainer Redis dan MySQL, disertai dengan konfigurasi *backup job* untuk replikasi database. Redis adalah *in-memory data structure store* yang dapat digunakan sebagai *database, cache, message broker, dan queue* [6]. Untuk mendukung operasional perusahaan, dilakukan instalasi berbagai aplikasi pendukung seperti Nextcloud untuk *file sharing*, menghadirkan platform kolaborasi dan berbagi file yang aman dengan kontrol penuh atas data dan *mail client* [7], serta Frappe Apps yang mencakup ERPNext, HRMS, dan *Customer Relationship Management (CRM)*. ERPNext merupakan sistem *Enterprise Resource Planning (ERP) open-source* yang menyediakan berbagai modul terintegrasi untuk manajemen bisnis [8].

Aspek keamanan server diperkuat dengan implementasi Fail2Ban pada minggu ke-10 untuk mencegah serangan *brute force*. Sepanjang periode implementasi, dilakukan juga migrasi berbagai aplikasi web dari *hosting* eksternal ke server Proxmox, termasuk *landing page* Bowbo Dimsum dan aplikasi pendukung lainnya. Proses pemeliharaan dan *troubleshooting* juga menjadi bagian penting, seperti yang dilakukan pada minggu ke-10 saat menangani server Proxmox yang mengalami *downtime*.

Keseluruhan implementasi Proxmox ini mencakup aspek-aspek penting dalam pengelolaan infrastruktur IT perusahaan, mulai dari virtualisasi, keamanan, pencadangan, pemantauan, hingga *deployment* aplikasi. Setiap tahapan implementasi dilakukan dengan mempertimbangkan kebutuhan operasional perusahaan dan mengikuti *best practices* dalam pengelolaan infrastruktur server.

3.2.1 Perencanaan dan Analisis Kebutuhan

3.2.1.1 Identifikasi Masalah dan Kebutuhan Perusahaan

Penulis bersama tim IT melakukan observasi dan diskusi intensif dengan berbagai divisi untuk mengidentifikasi permasalahan spesifik terkait infrastruktur teknologi yang ada. Tim IT kemudian mengambil aspek teknis yang memerlukan penanganan strategis untuk mendukung perkembangan bisnis perusahaan.

1. Penggunaan beberapa layanan hosting yang terus bertambah seiring berkembangnya perusahaan menghabiskan biaya yang cukup tinggi.
2. Tim kreatif menghadapi tantangan signifikan dalam proses *rendering* konten multimedia. Proses *rendering* yang memakan waktu lama pada masing-masing workstation individu mengurangi produktivitas tim dan menghambat penyelesaian proyek tepat waktu.
3. Penyimpanan data sensitif pada berbagai hosting dengan ukuran penyimpanan yang terbatas meningkatkan resiko keamanan dan mengurangi kontrol perusahaan terhadap aset digital penting.
4. Ketika terjadi gangguan layanan, perusahaan harus menunggu respons dari penyedia layanan, yang seringkali tidak sesuai dengan tingkat urgensi operasional bisnis.
5. Berbagi berkas dalam ukuran besar antar tim menghadapi kendala. Ketidakadaanya sistem *file sharing* terpusat memperlambat proses transfer data, terutama untuk file multimedia yang sering mencapai ukuran hingga 100 GB per sesi shooting, sehingga memperpanjang siklus produksi konten.

Berdasarkan identifikasi masalah tersebut, kebutuhan infrastruktur yang dirumuskan meliputi:

1. Sistem server yang dapat mengkonsolidasikan seluruh layanan digital perusahaan dalam satu *platform* terkelola.
2. Solusi penyimpanan dengan kapasitas yang cukup untuk menampung data operasional, arsip, dan kebutuhan produksi konten multimedia.
3. Spesifikasi performa yang dapat mendukung virtualisasi, *rendering*, dan beban kerja aplikasi bisnis secara simultan.
4. Mekanisme *backup* dan *failover* untuk memastikan ketersediaan layanan dan keamanan data.
5. Arsitektur yang memungkinkan penambahan kapasitas dan fungsionalitas sesuai dengan perkembangan kebutuhan bisnis.

3.2.1.2 Penyusunan Proposal Infrastruktur

Berdasarkan hasil analisis kebutuhan, tim IT menyusun proposal infrastruktur untuk diajukan kepada manajemen. Proposal ini mencakup beberapa komponen utama:

1. Rancangan infrastruktur server yang menggunakan Proxmox VE sebagai platform virtualisasi utama.
2. Rekomendasi komponen perangkat keras yang memenuhi kebutuhan performa, termasuk prosesor, RAM, penyimpanan redundan, dan akselerator grafis untuk mendukung beban kerja *rendering*.
3. Desain konfigurasi jaringan yang mengoptimalkan konektivitas internal dan eksternal, serta aspek keamanan dan efisiensi.

4. Rencana bertahap untuk migrasi dari infrastruktur yang ada ke sistem baru, termasuk jadwal, alokasi sumber daya, dan mitigasi resiko.
5. Protokol untuk pemeliharaan rutin, pemantauan kinerja, dan prosedur penanganan insiden.

Proposal kemudian dipresentasikan dalam format yang terstruktur untuk memudahkan pemahaman bagi pemangku kepentingan non-teknis, dengan penekanan pada manfaat operasional dan strategis bagi perusahaan.

3.2.1.3 Analisis Biaya dan Penghematan Jangka Panjang

Analisis finansial dilakukan untuk mengevaluasi kelayakan ekonomis dari implementasi infrastruktur baru dengan membandingkan dua skenario utama:

1. Struktur Biaya Hosting Model Tradisional
 - Biaya berlangganan *hosting* untuk website dari PT Maro Anugrah Jaya dan merk-merk dagang masing-masing perusahaan.
 - Biaya penyimpanan menggunakan '*cloud*' untuk data operasional dan multimedia.
 - Investasi pada komputer dengan performa tinggi untuk tim kreatif.
 - Biaya *bandwidth* untuk transfer data antar lokasi.
2. Struktur Biaya Infrastruktur Server Internal
 - Investasi awal untuk pengadaan perangkat keras.
 - Biaya operasional (listrik dan pemeliharaan).
 - Biaya domain dan konektivitas internet.
 - Peningkatan kapasitas bertahap sesuai kebutuhan.

Hasil analisis menunjukkan bahwa terdapat investasi awal yang signifikan, implementasi infrastruktur server internal memberikan penghematan jangka panjang yang lebih besar. Dalam proyeksi lima tahun, model server internal menunjukkan *total cost of ownership* yang lebih rendah dibandingkan dengan model *hosting* tradisional.

Faktor-faktor yang berkontribusi terhadap penghematan meliputi:

1. Penghapusan biaya *hosting* untuk semua *website* dan layanan.
2. Pemanfaatan bersama kapasitas komputasi dan penyimpanan antar departemen.
3. Kemampuan *rendering* terpusat mengurangi kebutuhan investasi pada komputer performa tinggi.
4. Peningkatan produktivitas melalui akses lebih cepat ke data dan aplikasi.

3.2.2 Pengadaan dan Perakitan Infrastruktur

3.2.2.1 Pemilihan Komponen Hardware

Pemilihan komponen perangkat keras untuk infrastruktur server Proxmox dilakukan melalui analisis kebutuhan yang komprehensif dengan mempertimbangkan beban kerja saat ini dan proyeksi pertumbuhan di masa mendatang. Proses seleksi komponen dilaksanakan dengan memperhatikan aspek performa, keandalan, dan nilai ekonomis jangka panjang.

a. Spesifikasi Teknis

Tabel 3.5 Spesifikasi Teknis Server

Komponen	Jumlah	Merk	Model	Justifikasi
CPU	1	AMD	Ryzen 9 7950X	Prosesor 16 inti untuk mendukung virtualisasi dan beban kerja simultan
RAM	2	Klevv	CRAS V RGB DDR5	Memori dengan kapasitas 64GB untuk menangani VM dan <i>kontainer</i> .
Motherboard	1	Gigabyte	B650E Aorus Elite X AX Ice	Mendukung PCIe 4.0 dan fitur virtualisasi.
GPU	1	Gigabyte	RTX 4090 Aorus Xtreme	Mendukung <i>rendering</i> video.
Penyimpanan Sistem	1	ADATA	LEGEND 900 1TB	SSD NVMe 1TB untuk sistem operasi dan <i>cache</i> dengan kecepatan baca/tulis tinggi.
Penyimpanan Data	2	Seagate	EXOS X12 Enterprise 12TB	Hard drive <i>enterprise-grade</i> untuk penyimpanan data dengan konfigurasi RAID1.

b. Justifikasi Kebutuhan CPU (AMD Ryzen 9 7950X):

Perusahaan membutuhkan prosesor berperforma tinggi dengan 16 core karena:

- PT Maro Anugrah Jaya perlu menjalankan minimal 10-12 kontainer terpisah secara bersamaan untuk berbagai layanan (database, web server, file sharing, ERP, dll).
- Tim kreatif perusahaan memerlukan daya komputasi tinggi untuk proses rendering dan editing konten multimedia.
- Aplikasi ERPNext yang menjadi tulang punggung operasional perusahaan membutuhkan minimal 4 core untuk performa optimal, terutama saat menangani transaksi simultan dari berbagai divisi.
- Analisis beban kerja menunjukkan bahwa dengan pertumbuhan bisnis, perusahaan akan memerlukan minimal 12-14 core dalam 2 tahun ke depan.

c. Justifikasi Kebutuhan RAM (64GB DDR5):

Analisis kebutuhan memori menunjukkan bahwa perusahaan membutuhkan 64GB RAM karena:

- Database MariaDB untuk ERPNext dan aplikasi perusahaan memerlukan alokasi 16GB untuk optimal.
- *File sharing* Nextcloud membutuhkan minimal 8GB untuk menangani transfer file simultan dari 30+ pengguna.
- Kontainer aplikasi web dan backend APIs memerlukan total 16GB.
- *Virtual machine* Windows untuk rendering membutuhkan alokasi 16GB agar dapat menjalankan

software Adobe Creative Suite dan Blender secara optimal.

- *Monitoring stack* dan layanan jaringan memerlukan 4-6GB tambahan.

d. Justifikasi Kebutuhan GPU (RTX 4090):

Tim kreatif PT Maro Anugrah Jaya membutuhkan GPU performa tinggi karena:

- Produksi konten marketing untuk 5 sub-brand perusahaan (Bowbo Xpress, Bowbo Munchies, dll.) memerlukan *rendering* video 1080p secara reguler.
- Waktu *rendering* dengan *workstation* yang ada mencapai 30 menit sampai 1 jam per video, menghambat produktivitas tim.
- Dengan RTX 4090, waktu *rendering* dapat diturunkan hingga 80%, memungkinkan iterasi lebih cepat. Dan tim dapat langsung mengerjakan proyek selanjutnya tanpa terhambat komputasi yang ada.
- Tim marketing membutuhkan minimal 3-4 video promosi per minggu, dengan GPU *passthrough* dapat dilakukan secara paralel.
- Analisis biaya menunjukkan bahwa satu GPU RTX 4090 shared lebih ekonomis dibanding membeli 3-4 workstation terpisah dengan GPU *mid-range*.

e. Justifikasi Kebutuhan Penyimpanan Data (2x 12TB HDD):

Perusahaan membutuhkan kapasitas penyimpanan 12TB efektif (24TB *raw* dalam RAID1) berdasarkan:

- Volume Data Tim Kreatif: Tim kreatif PT Maro Anugrah Jaya secara rutin menangani proyek video dengan total *raw footage* mencapai 100GB per sesi

shooting. Dengan 5 sesi *shooting* per bulan, akumulasi data mencapai 6TB per tahun.

- Arsip Operasional Perusahaan: Data operasional dari 5 sub-brand (Bowbo Xpress, Bowbo Munchies, Bowbo Dimsum, dll) mencakup transaksi harian, laporan keuangan, dan dokumentasi yang menghasilkan sekitar 200GB data per bulan atau 2.4TB per tahun.
- Backup Website dan Aplikasi: Total 8 website dan aplikasi perusahaan memerlukan pencadangan reguler dengan retensi, menghabiskan sekitar 1.5TB.
- Dokumen Kolaboratif dan *File Sharing*: Implementasi Nextcloud untuk kurang lebih 50 karyawan dengan alokasi 10GB per pengguna membutuhkan minimal 500GB, dengan proyeksi pertumbuhan hingga 1TB dalam setahun.
- Proyeksi Pertumbuhan: Dengan rencana ekspansi bisnis ke bidang perhotelan, layanan gas, dan manajemen kos, volume data diproyeksikan tumbuh 40% per tahun.
- Retensi Data: Kebijakan retensi data perusahaan mengharuskan penyimpanan data operasional minimum 3 tahun, dan data finansial hingga 7 tahun.

Konfigurasi RAID 1 (*mirroring*) dipilih untuk memastikan ketersediaan data kritis perusahaan. Analisis menunjukkan kapasitas efektif 12TB akan memadai untuk 2-3 tahun operasional, dengan mempertimbangkan pertumbuhan bisnis dan kebutuhan penyimpanan data yang semakin meningkat.

3.2.2.2 Proses Pengadaan Perangkat Keras

Proses pengadaan perangkat keras dilaksanakan melalui beberapa tahapan sistematis untuk memastikan kesesuaian dengan spesifikasi yang dibutuhkan dan efisiensi anggaran. Tahapan pengadaan meliputi:

1. Penyusunan daftar kebutuhan komponen berdasarkan spesifikasi teknis pada tabel 3.5.
2. Penelusuran vendor dan *supplier* yang menyediakan komponen dengan kualitas dan garansi yang memadai.
3. Perbandingan harga dan ketersediaan dari minimal tiga vendor berbeda untuk setiap komponen.
4. Pengajuan proposal pengadaan kepada manajemen dengan rincian spesifikasi, harga, dan justifikasi teknis.
5. Setelah mendapatkan persetujuan, dilakukan pemesanan komponen.
6. Verifikasi komponen yang diterima secara langsung bersama vendor dan *supplier*, termasuk pemeriksaan fisik dan kelengkapan dokumentasi garansi.

III. ANALISIS KEBUTUHAN

Jumlah Website	5	Bowbo Xpress, Bowbo Munchies, PT Maro Anugrah Jaya (Administrasi & Lapangan), PT Wadah Berkah Nusantara Jaya, dan mendatang (Kos, Gas, Perhotelan)
Kapasitas Penyimpanan	-	10TB (10.000 GB)*
Kebutuhan Kinerja	-	CPU dengan > 10 Core** RAM 64GB** GPU***
Keamanan	Ya	Pentingnya penyimpanan tersendiri mengingat pentingnya data konsumen dan administrasi perusahaan.
Redudansi	Ya	Pentingnya penyimpanan yang <i>reliable</i> (ter-backup 1:1)

* Menimbang hasil survey tim kreatif yang memerlukan 100GB untuk 5 hari *shooting*.

* Menimbang keperluan QC rumah, penyimpanan berkas konsumen, bank, dan administrasi kantor secara digital dan disentralisasi melalui website PT Maro Anugrah Jaya.

** Menimbang keperluan website dan *rendering* tim kreatif.

*** Menimbang keperluan *rendering* tim kreatif (opsional)

IV. SPESIFIKASI TEKNIS

Type	Nama	Jumlah	Harga Satuan	Total
CPU	AMD Ryzen 9 7950X (16 Core)	1	Rp 6.700.000	Rp 6.700.000
RAM	KLEVV CRAS V RGB DDR5 PC51200 6400MHz 64GB (2x32GB)	1	Rp 2.000.000	Rp 2.000.000
Motherboard	GIGABYTE B650 Aorus Elite AX	1	Rp 4.000.000	Rp 4.000.000
GPU	MSI RTX 3090 SUPRIM X 24GB GDDR6X	1	Rp 11.000.000	Rp 11.000.000
HDD	Seagate Ironwolf NAS 10TB	2	Rp 5.500.000	Rp 11.000.000
SSD	Team MP33 512GB M.2 NVME	1	Rp 1.000.000	Rp 1.000.000
Case	Deepcool Macube 310P	1	Rp 300.000	Rp 300.000
PSU	Seasonic Focus Gold GX-850	1	Rp 2.000.000	Rp 2.000.000
TOTAL				Rp 40.000.000

Note: Spesifikasi masih dapat menyesuaikan anggaran yang diberikan.

Gambar 3.3 Bahasan analisis kebutuhan dan spesifikasi teknis

V. ANALISIS BIAYA

Untuk membahas poin penghematan biaya, berikut merupakan rincian biaya yang seharusnya keluar apabila tidak melakukan pengadaan server:

Pengeluaran		Biaya	Biaya 5 Tahun
Cloud Hosting Website*	PT Maro Anugerah Jaya (Administrasi)	Rp. 100.000 perbulan	Rp. 6.000.000
	PT Maro Anugerah Jaya (Lapangan)	Rp. 200.000 perbulan	Rp. 12.000.000
	PT Wadah Berkah Nusantara	Rp. 200.000 perbulan	Rp. 12.000.000
	Bowbo Xpress	Rp. 200.000 perbulan	Rp. 12.000.000
	Bowbo Munchies	Rp. 200.000 perbulan	Rp. 12.000.000
	PT Kuda Mas Perkasa	Rp. 200.000 perbulan	Rp. 12.000.000
	Kos	Rp. 200.000 perbulan	Rp. 12.000.000
	Perhotelan	Rp. 200.000 perbulan	Rp. 12.000.000
Biaya Perakitan PC Editing**		Rp. 10.000.000	Rp. 50.000.000***
TOTAL			Rp. 118.000.000

* Biaya diambil dari DomaiNesia dan Hostinger (provider hosting sekarang).

** Biaya diambil dari perakitan PC Editing sebelumnya yang memerlukan GPU dengan spesifikasi yang mumpuni.

*** Biaya yang keluar apabila merakit 5 buah PC Editing.

Potensi penghematan apabila melakukan pengadaan server:

Pengeluaran		Biaya	Biaya 5 Tahun
Cloud Hosting Website*	PT Maro Anugerah Jaya (Administrasi)	Rp. 100.000 perbulan	Rp. 6.000.000
	PT Maro Anugerah Jaya (Lapangan)	Rp. 200.000 perbulan	Rp. 12.000.000
	PT Wadah Berkah Nusantara	Rp. 200.000 perbulan	Rp. 12.000.000
	Bowbo Xpress	Rp. 200.000 perbulan	Rp. 12.000.000
	Bowbo Munchies	Rp. 200.000 perbulan	Rp. 12.000.000
	PT Kuda Mas Perkasa	Rp. 200.000 perbulan	Rp. 12.000.000
	Kos	Rp. 200.000 perbulan	Rp. 12.000.000
	Perhotelan	Rp. 200.000 perbulan	Rp. 12.000.000
Biaya Perakitan PC Editing**		Rp. 10.000.000	Rp. 50.000.000***
TOTAL			Rp. 118.000.000

* Biaya Rp 20.000 dan 125.000 merupakan biaya *domain* website.

** Biaya diambil dari perakitan PC Editing yang spesifikasinya diturunkan dari perakitan awal (karena sudah di *handle* oleh server).

*** Biaya yang keluar apabila merakit 5 buah PC Editing.

Gambar 3.4 Bahasan analisis biaya pada proposal

Pengadaan dilakukan secara bertahap dengan prioritas pada komponen utama seperti *motherboard*, CPU, dan RAM yang menjadi pondasi sistem. Pendekatan ini memungkinkan kompatibilitas awal sebelum melanjutkan ke pengadaan komponen lainnya.

Tantangan utama dalam proses pengadaan adalah ketersediaan komponen tertentu, khususnya GPU NVIDIA RTX 4090 yang masih mengalami fluktuasi harga dan ketersediaan di pasaran. Untuk mengatasi hal ini, dilakukan pemesanan awal dan koordinasi intensif dengan *supplier* untuk memastikan ketersediaan pada waktu yang dibutuhkan.

3.2.2.3 Perakitan Fisik Server

Perakitan fisik server dilaksanakan di lokasi Bowbo Xpress *outlet* Melati Mas yang telah dipersiapkan sebagai ruang server. Proses perakitan dilakukan dengan memperhatikan aspek keamanan komponen dan optimalisasi aliran udara untuk pendinginan yang efektif. Tahapan perakitan meliputi:

1. Persiapan area kerja dengan memastikan kebersihan dan bebas dari listrik statis yang dapat merusak komponen elektronik.
2. Instalasi prosesor AMD Ryzen 9 7950X pada soket *motherboard* dengan memperhatikan orientasi yang benar dan aplikasi *thermal paste* secukupnya.
3. Pemasangan sistem pendingin air dengan konfigurasi radiator pada bagian atas *casing* untuk pembuangan panas yang optimal.
4. Instalasi modul RAM pada slot yang sesuai dengan konfigurasi *dual-channel* untuk memaksimalkan *bandwidth* memori.
5. Pemasangan *motherboard* pada *casing*.

6. Instalasi SSD NVMe pada slot M.2 *motherboard* sebagai *drive* sistem utama.
7. Pemasangan *hard drive* Seagate EXOS pada bay penyimpanan dengan konfigurasi kabel yang rapi untuk memaksimalkan aliran udara.
8. Instalasi GPU pada slot PCIe utama.
9. Pemasangan *power supply* dan pengaturan kabel yang sistematis untuk memudahkan pengelolaan dan optimalisasi aliran udara.
10. Konfigurasi koneksi *front panel* dan pengaturan kabel internal untuk memastikan kerapihan dan aksesibilitas.

Setelah perakitan fisik selesai, dilakukan pengujian POST (*Power-On Self Test*) untuk verifikasi bahwa semua komponen terdeteksi dengan benar oleh sistem. Pengujian ini meliputi verifikasi deteksi CPU, RAM, penyimpanan, dan komponen lainnya melalui BIOS/UEFI.

Tantangan yang dihadapi selama proses perakitan adalah pengaturan kabel yang optimal untuk memastikan aliran udara yang baik dalam *casing*, terutama dengan adanya komponen berukuran besar, seperti GPU RTX 4090 dan radiator pendingin cairan. Hal ini diatasi dengan perencanaan rute kabel yang sistematis dan penggunaan *cable management* yang tersedia pada *casing*.

Setelah perakitan fisik dan pengujian awal selesai, server siap untuk tahap berikutnya, yaitu instalasi dan konfigurasi sistem operasi Proxmox VE sebagai platform virtualisasi.

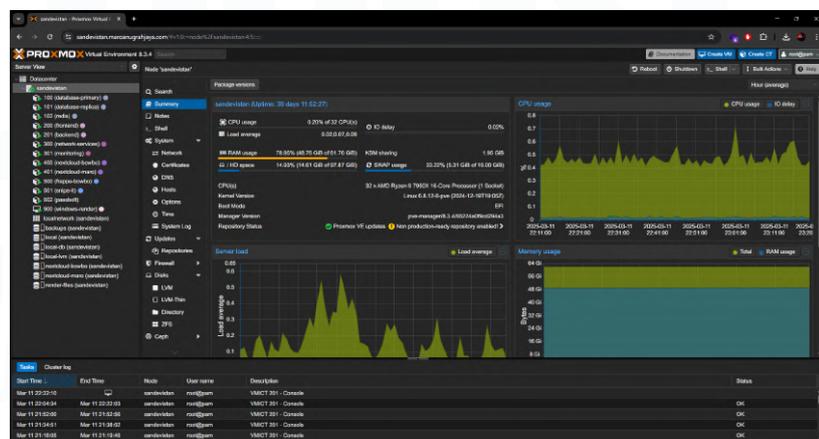
3.2.3 Instalasi dan Konfigurasi Dasar Proxmox VE

3.2.3.1 Instalasi Sistem Operasi Proxmox VE

Proxmox VE sebagai platform virtualisasi dipilih karena merupakan solusi virtualisasi *open-source* yang menawarkan kemampuan manajemen *virtual machine* berbasis KVM dan *kontainer* berbasis LXC dalam satu platform terintegrasi.

Proses instalasi Proxmox VE dilakukan dengan langkah-langkah berikut:

1. Pembuatan media instalasi *bootable* menggunakan image ISO Proxmox versi 8.3.1.
2. Konfigurasi *Basic Input/Output System* (BIOS) server untuk dapat melakukan *booting* dari media instalasi.
3. Eksekusi proses instalasi dengan target SSD.
4. Konfigurasi partisi sistem dengan alokasi yang ada pada tabel 3.7 untuk sistem operasi dan *swap*.
5. Penetapan *hostname* untuk server.
6. Konfigurasi awal jaringan dengan alamat DHCP untuk mendapatkan alamat IP awal.
7. Penetapan kredensial administrator untuk akses ke platform manajemen Proxmox.



Gambar 3.5 Hasil tangkapan layar pada Proxmox

Gambar 3.5 *Dashboard* Proxmox VE menampilkan status dan metrik sistem termasuk penggunaan CPU, memori, dan penyimpanan dari berbagai kontainer LXC dan *virtual machine* yang berjalan. Panel sebelah kiri menunjukkan hierarki node dan kontainer, sementara grafik di bagian kanan menampilkan statistik performa *real-time*.

3.2.3.2 Konfigurasi Jaringan dan IP Statis

Konfigurasi jaringan merupakan aspek yang penting dalam implementasi infrastruktur server. Untuk memastikan konsistensi akses dan memudahkan manajemen, diterapkan skema IP statis untuk host Proxmox dan seluruh kontainer serta *virtual machine* yang akan diimplementasikan.

Tabel 3.6 Konfigurasi alamat IP lokal jaringan

Host	IP Address	Purpose
Proxmox	192.168.0.2	Host System
LXC100	192.168.0.10	Primary Database Kontainer
LXC101	192.168.0.11	Replica Database Kontainer
LXC102	192.168.0.12	Redis Cache
LXC200	192.168.0.20	Frontend Kontainer
LXC201	192.168.0.21	Backend Kontainer
LXC300	192.168.0.30	Network Services
LXC301	192.168.0.31	Monitoring Kontainer
LXC400	192.168.0.40	Nextcloud Bowbo
LXC401	192.168.0.41	Nextcloud Maro
LXC500	192.168.0.50	Frappe
LXC501	192.168.0.51	Snipe IT
LXC502	192.168.0.52	Passbolt
VM900	192.168.0.14 (DHCP)	Windows Render

Konfigurasi jaringan dilakukan dengan pendekatan segmentasi berdasarkan fungsi, dimana:

- Jarak alamat IP 10-19 dialokasikan untuk layanan *database* dan *cache*.
- Jarak alamat IP 20-29 dialokasikan untuk layanan *frontend* dan *backend* website.
- Jarak alamat IP 30-39 dialokasikan untuk layanan jaringan dan pemantauan.
- Jarak alamat IP 40-49 dialokasikan untuk layanan *file sharing*.
- Jarak alamat IP 50-59 dialokasikan untuk aplikasi bisnis.

Implementasi konfigurasi jaringan dilakukan melalui file konfigurasi ``etc/network/interfaces`` pada host Proxmox dengan menentukan parameter berikut:

- *Interface* jaringan utama
- Alamat IP statis
- *Subnet mask*
- *Gateway*
- DNS Server

Berikut konfigurasi pada file ``etc/network/interfaces``:

```

auto lo
iface lo inet loopback

iface enp8s0 inet manual

auto vmbr0
iface vmbr0 inet static
    address 192.168.0.2/24
    gateway 192.168.0.1
    bridge-ports enp8s0
    bridge-stp off
    bridge-fd 0

iface wlp7s0 inet manual

```

```
source /etc/network/interfaces.d/*
```

Untuk meningkatkan keamanan jaringan, diterapkan konfigurasi *firewall* pada level *host* Proxmox yang membatasi akses hanya pada port-port yang diperlukan untuk operasional sistem dan aplikasi.

3.2.3.3 Alokasi dan Manajemen Penyimpanan

Manajemen penyimpanan pada infrastruktur Proxmox dirancang untuk mengoptimalkan performa, keandalan, dan fleksibilitas. Implementasi menggunakan kombinasi beberapa teknologi penyimpanan yang disesuaikan dengan karakteristik beban kerja yang akan ditangani.

Tabel 3.7 Tata letak penyimpanan server

Nama Storage	Tipe	Ukuran	Kegunaan
local	Direktori	100GB	Sistem dan file tempo
local-db	Direktori	400GB	Penyimpanan database
local-lvm	LVM Thin	400GB	Kontainer dan penyimpanan sistem
backups	Direktori	1.5TB	Backup sistem dan data
nextcloud-bowbo	ZFS Pool	4.8TB	File sharing Bowbo Group
nextcloud-mar0	ZFS Pool	4.8TB	File sharing Mar0 Anugrah Jaya
render-files	ZFS Pool	500GB	File rendering untuk tim kreatif

Konfigurasi penyimpanan mengimplementasikan beberapa strategi untuk mengoptimalkan penggunaan:

1. Redundansi data, implementasi RAID 1 pada level perangkat keras untuk *hard drive* Seagate EXOS, memberikan perlindungan terhadap kegagalan disk tunggal.
2. Penerapan konsep *tiered storage* dengan SSD NVMe untuk sistem operasi dan aplikasi yang membutuhkan performa tinggi. Lalu terdapat *hard drive* untuk penyimpanan data dengan volume besar.
3. ZFS *storage pools*, implementasi ZFS untuk penyimpanan *file sharing* dan *rendering* yang menawarkan integritas data melalui *checksumming*, kompresi data untuk pengoptimalan ruang, serta *snapshot* untuk *point-in-time recovery*.
4. *Backup storage*, alokasi *storage pool* khusus untuk *backup* dan kapasitas yang cukup untuk menyimpan *snapshot* sistem dan data penting.

Manajemen penyimpanan dilakukan *interface* Proxmox yang menyediakan visibilitas terhadap penggunaan ruang dan performa. Pemantauan rutin dilakukan untuk mengidentifikasi tren penggunaan dan mengantisipasi kebutuhan ekspansi di masa mendatang.

3.2.3.4 Alokasi Daya Komputasi

Alokasi daya komputasi pada infrastruktur Proxmox dirancang untuk memastikan distribusi sumber daya yang optimal sesuai dengan prioritas dan kebutuhan masing-masing layanan. Pendekatan ini memaksimalkan efisiensi penggunaan perangkat keras dan meminimalisir resiko *resource contention*.

Tabel 3.8 Tata letak konfigurasi kontainer dan *virtual machine*

Kontainer VM ID	CPU Cores	Memory	Prioritas	Beban Kerja
LXC100 (database-primary)	4	8 GB	Tinggi	<i>Database</i> utama untuk keseluruhan layanan
LXC101 (database-replica)	2	4 GB	Menengah	Replika <i>database</i> untuk redundansi
LXC102 (redis)	2	4 GB	Tinggi	Cache untuk meningkatkan performa layanan
LXC200 (frontend)	4	8 GB	Menengah	Server web dan aplikasi <i>frontend</i>
LXC201 (backend)	4	8 GB	Menengah	API dan layanan <i>backend</i>
LXC300 (network-services)	2	2 GB	Tinggi	DNS, <i>tunnelling</i> , dan layanan jaringan
LXC301 (monitoring)	2	4 GB	Rendah	Prometheus, Grafana, untuk pemantauan
LXC400 (nextcloud-bowbo)	4	8 GB	Menengah	<i>File sharing</i> untuk Bowbo Group
LXC401 (nextcloud-mar0)	4	8 GB	Menengah	<i>File sharing</i> untuk Maro Anugrah Jaya
LXC500 (frappe-bowbo)	6	4 GB	Tinggi	ERPNext dan aplikasi bisnis dari Frappe
LXC501 (snipe-it)	2	4 GB	Rendah	Manajemen inventaris IT
LXC502 (passbolt)	2	2 GB	Rendah	Manajemen password
VM900 (windows-render)	8	16 GB	Rendah	<i>Rendering</i> video dan print server

Strategi alokasi daya komputasi diimplementasikan dengan mempertimbangkan beberapa faktor:

1. Layanan kritis seperti *database* dan *cache* mendapatkan prioritas tinggi dengan alokasi CPU dan memori yang besar.
2. Layanan *database* dialokasikan dengan CPU dan memori yang cukup. Layanan jaringan seperti *frontend* dan *backend* mendapatkan alokasi seimbang. Layanan *rendering* mendapatkan alokasi CPU tinggi dan akses ke GPU.
3. Implementasi limitasi CPU, untuk mencegah kontainer mengkonsumsi seluruh sumber daya. Berlaku hal yang sama kepada memori.
4. Penerapan *overcommitment* rasio 1:2 untuk CPU, memungkinkan alokasi virtual CPU lebih banyak dari CPU fisik.

Konfigurasi alokasi daya komputasi diimplementasikan melalui file konfigurasi kontainer dan VM, serta dipantau secara berkala untuk memastikan distribusi sumber daya yang optimal. Penyesuaian dilakukan berdasarkan analisis performa dan kebutuhan yang berkembang.

3.2.3.5 Implementasi Keamanan Dasar

Keamanan merupakan aspek kritis dalam implementasi infrastruktur server, terutama untuk sistem yang mengelola data sensitif perusahaan. Pada infrastruktur Proxmox VE, diterapkan beberapa lapisan keamanan dasar untuk melindungi sistem dari berbagai ancaman potensial.

1. Implementasi Fail2Ban

Fail2Ban diimplementasikan sebagai mekanisme pertahanan utama terhadap serangan *brute force*. Aplikasi ini memindai *file log* seperti ``/var/log/auth.log`` dan secara otomatis

memblokir alamat IP yang melakukan terlalu banyak percobaan login yang gagal dengan memperbarui aturan *firewall* sistem.

Konfigurasi Fail2Ban pada server Proxmox mencakup:

- Proteksi SSH, konfigurasi untuk memblokir IP yang melakukan percobaan login SSH yang gagal berulang kali.
- Proteksi Proxmox Web UI, konfigurasi untuk memblokir percobaan login yang gagal pada antarmuka web Proxmox.

2. Implementasi *Time-based One-Time Password* (TOTP)

Untuk meningkatkan keamanan akses ke sistem Proxmox, diimplementasikan autentikasi dua faktor menggunakan TOTP sesuai dengan RFC 6238. Implementasi ini memberikan lapisan keamanan tambahan di luar kata sandi tradisional.

Konfigurasi TOTP pada Proxmox VE meliputi:

- Instalasi modul PAM untuk TOTP: ``apt-get install libpam-oauth``
- Konfigurasi file PAM untuk mengintegrasikan TOTP dengan sistem autentikasi Proxmox.
- Pembuatan kunci rahasia untuk setiap pengguna administrator.
- Melakukan konfigurasi aplikasi autentikator pada perangkat mobile administrator (seperti Google Authenticator, Authy, atau aplikasi serupa).

Dengan implementasi TOTP, akses ke sistem Proxmox memerlukan kombinasi dari sesuatu yang diketahui oleh pengguna (kata sandi) dan sesuatu yang dimiliki oleh pengguna (perangkat *mobile* dengan aplikasi autentikator).

Pendekatan ini secara signifikan mengurangi resiko akses tidak sah meskipun kata sandi pengguna terkompromisasi.

3. Konfigurasi Firewall Proxmox

Proxmox VE menyediakan *firewall* yang dimanfaatkan untuk membatasi akses ke sistem, konfigurasi *firewall* yang diterapkan meliputi:

- Aturan *default*, melakukan larangan terhadap semua *inbound traffic* dan mengizinkan semua *traffic* untuk tipe *outbound*.
- *Port filtering*, hanya port yang diperlukan untuk operasional yang dibuka, seperti SSH (port 22), Proxmox Web UI (port 8006), dan port spesifik untuk layanan tertentu.

4. *Hardening Secure Shell Protocol (SSH)*

Membatasi akses SSH ke server Proxmox dengan konfigurasi sebagai berikut:

- Penonaktifan login `'root'` saat menggunakan SSH.
- Implementasi autentikasi berbasis file kunci dan penonaktifan autentikasi kata sandi.
- Melakukan konfigurasi *timeout* apabila administrator *idle* terlalu lama.
- Pembatasan algoritma enkripsi dan kriptografi yang digunakan.

5. Isolasi Kontainer dan VM

Implementasi isolasi yang ketat bahwa kontainer dan virtual machine untuk menjadi masalah serangan `'escape'`, yang membuat serangan dalam satu kontainer menyebar hingga beberapa kontainer lain, hingga *host* dari kontainer tersebut bisa mengetahuinya.

Implementasi keamanan dasar ini memberikan pondasi yang kuat untuk melindungi infrastruktur Proxmox VE dari berbagai ancaman. Pendekatan yang diterapkan memastikan bahwa meskipun satu lapisan tertembus, lapisan lain tetap dapat memberikan perlindungan. Strategi keamanan ini akan terus dievaluasi dan ditingkatkan seiring dengan evolusi ancaman dan kebutuhan bisnis.

3.2.4 Implementasi Kontainer dan Virtual Machine

Setelah konfigurasi dasar Proxmox VE dan implementasi keamanan, tahap selanjutnya adalah *deployment* kontainer dan *virtual machine* yang akan menjadi fondasi untuk berbagai layanan yang dibutuhkan perusahaan. Implementasi ini mengadopsi pendekatan *hybrid* dengan memanfaatkan teknologi kontainer *Linux Container* (LXC) untuk efisiensi sumber daya dan *virtual machine* KVM untuk kasus penggunaan khusus.

3.2.4.1 Perancangan dan Deployment LXC Kontainer

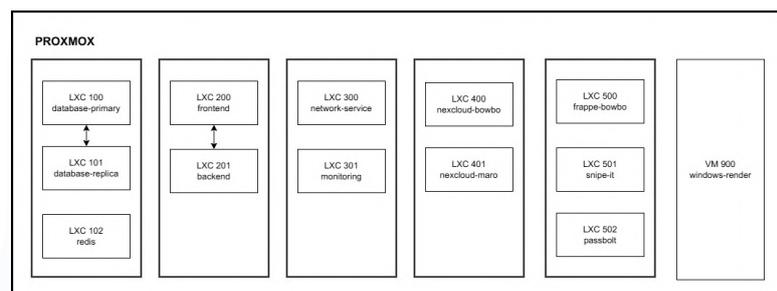
Perancangan kontainer LXC dilakukan dengan mempertimbangkan prinsip *separation of concerns*, di mana setiap *kontainer* memiliki fungsi spesifik dan terisolasi. Pendekatan ini meningkatkan keamanan, memudahkan pemeliharaan, dan memungkinkan alokasi sumber daya yang optimal sesuai kebutuhan masing-masing layanan.

a. Strategi Perancangan *Kontainer*

Strategi perancangan *kontainer* didasarkan pada beberapa prinsip utama:

1. *Kontainer* dikelompokkan berdasarkan fungsi utamanya:
 - Database dan Cache (LXC100-102)
 - Frontend dan Backend (LXC200-201)
 - Network Services (LXC300-301)

- File Sharing (LXC400-401)
 - Aplikasi Bisnis (LXC500-502)
2. Penggunaan template standar sebagai basis untuk semua kontainer:
- Ubuntu 22.04 LTS untuk kontainer aplikasi dan layanan web.
 - Debian TurnKey Nextcloud 18.1 untuk *kontainer* aplikasi *file sharing*.
 - Debian TurnKey Snipe IT 18.0 untuk *kontainer* aplikasi manajemen asset IT.
3. Alokasi sumber daya berdasarkan karakteristik beban kerja:
- *Kontainer database* mendapatkan prioritas CPU dan memori yang lebih tinggi.
 - *Kontainer* aplikasi web mendapatkan alokasi CPU yang seimbang.
 - *Kontainer* pemantauan dan utilitas mendapatkan alokasi minimal namun cukup.



Gambar 3.6 Struktur kontainer pada Proxmox

Gambar 3.6 Struktur kontainer pada Proxmox VE yang menunjukkan pembagian layanan ke dalam beberapa grup kontainer LXC dan satu VM. Grup pertama (LXC 100-102) menangani database dan *caching*, grup kedua (LXC 200-201) untuk aplikasi web, grup ketiga (LXC 300-301) untuk layanan jaringan dan pemantauan, grup keempat (LXC 400-

401) untuk layanan *file sharing*, grup kelima (LXC 500-502) untuk manajemen aset dan kredensial, serta VM 900 yang didedikasikan untuk *rendering* dengan GPU *passthrough*.

b. Proses *Deployment Kontainer*

1. Pembuatan *Kontainer* Dasar

```
pct create 100 local:vztmpl/debian-11-standard_11.3-1_amd64.tar.zst --hostname database-primary --cores 4 --memory 8192 --swap 4096 --rootfs local-lvm:175 --net0 name=eth0,bridge=vbr0,ip=192.168.0.10/24,gw=192.168.0.1
```

Perintah serupa digunakan untuk membuat seluruh kontainer dengan parameter yang disesuaikan untuk masing-masing fungsi. Setiap *kontainer* dikonfigurasi dengan:

- *Template* sistem operasi yang sesuai
- ID dan *hostname* deskriptif
- Alokasi CPU dan memori
- Penyimpanan untuk sistem dan data
- Konfigurasi jaringan dengan IP statis

2. *Deployment* Layanan Spesifik

Setiap *kontainer* kemudian dikonfigurasi dengan layanan spesifik sesuai dengan fungsinya:

- Database Kontainer (LXC100-101)
- Redis Kontainer (LXC102)
- Frontend Kontainer (LXC200)
- Backend Kontainer (LXC201)
- Network Services Kontainer (LXC300)
- Monitoring Kontainer (LXC301)
- File Sharing Kontainer (LXC400-401)
- Aplikasi Kontainer (LXC500-502)

3.2.4.2 Konfigurasi Windows VM dengan GPU *Passthrough*

Untuk mendukung kebutuhan tim kreatif dalam melakukan *rendering* video dan pengolahan grafis, diimplementasikan *virtual machine* Windows dengan konfigurasi GPU *passthrough*. Implementasi ini memungkinkan VM Windows mengakses langsung perangkat keras GPU, memberikan performa yang mendekati *bare-metal* untuk aplikasi yang intensif grafis.

a. Persiapan Host untuk GPU *Passthrough*

Persiapan host Proxmox untuk GPU *passthrough* dilakukan melalui serangkaian konfigurasi yang memungkinkan isolasi GPU dari *host* dan mempersiapkannya untuk digunakan secara eksklusif oleh *virtual machine*.

Proses ini meliputi beberapa tahapan konfigurasi:

1. Konfigurasi GRUB untuk aktivasi IOMMU

Input-Output Memory Management Unit (IOMMU) merupakan komponen penting yang memungkinkan perangkat PCI seperti GPU dapat diisolasi dan diakses langsung oleh *virtual machine* [9]. Aktivasi IOMMU dilakukan melalui konfigurasi GRUB:

```
# Edit file konfigurasi GRUB
nano /etc/default/grub

# Ubah baris berikut
GRUB_CMDLINE_LINUX_DEFAULT="quiet"
# Menjadi
GRUB_CMDLINE_LINUX_DEFAULT="quiet
intel_iommu=on iommu=pt
pcie_acs_override=downstream,multifunction nofb nomodeset
video=vesafb:off,efifb:off"
```

```
# Simpan file dan update GRUB
update-grub
```

Parameter ``amd_iommu=on`` mengaktifkan IOMMU untuk prosesor AMD, sementara ``iommu=pt`` mengaktifkan mode *passthrough*. Parameter tambahan seperti ``nofb nomodeset video=vesafb:off,efifb:off`` mencegah *host* menggunakan *framebuffer* yang dapat mengganggu proses *passthrough*.

2. Konfigurasi Modul Kernel untuk VFIO

Virtual Function I/O (VFIO) adalah *framework* yang memungkinkan akses langsung ke perangkat PCI dari *userspace*, yang diperlukan untuk GPU *passthrough*. Modul kernel VFIO perlu dimuat saat *boot*:

```
# Edit file modules
nano /etc/modules

# Tambahkan baris berikut
vfio
vfio_iommu_type1
vfio_pci
vfio_virqfd
```

3. Konfigurasi IOMMU dan KVM

Konfigurasi tambahan diperlukan untuk memastikan IOMMU dan KVM berfungsi dengan baik untuk *passthrough*.

```
# Konfigurasi untuk mengizinkan interupsi yang tidak aman
nano /etc/modprobe.d/iommu_unsafe_interrupts.conf

# Tambahkan baris berikut
options vfio_iommu_type1 allow_unsafe_interrupts=1
```

```
# Konfigurasi KVM untuk mengabaikan MSRs yang tidak
didukung
nano /etc/modprobe.d/kvm.conf

# Tambahkan baris berikut
options kvm ignore_msrs=1
```

Parameter ``allow_unsafe_interrupts=1`` memungkinkan kompatibilitas dengan beberapa GPU. Sementara ``ignore_msrs=1`` mencegah KVM *crash* ketika VM mencoba mengakses *Model-Specific Registers* yang tidak didukung.

4. *Blacklisting* Driver GPU

Untuk mencegah *host* menggunakan GPU yang akan di-*passthrough*, driver GPU perlu di-*blacklist*:

```
# Edit file blacklist
nano /etc/modprobe.d/blacklist.conf

# Tambahkan baris berikut
blacklist radeon
blacklist nouveau
blacklist nvidia
blacklist nvidiafb
```

5. Finalisasi Konfigurasi

Langkah terakhir adalah mengidentifikasi ID vendor dan *device* dari GPU, kemudian mengkonfigurasi VFIO untuk menggunakan GPU tersebut:

```
# Identifikasi alamat PCI dari GPU
lspci -v

# Dapatkan ID vendor dan device dari GPU
lspci -n -s [alamat_PCI_GPU]
```

```
# Konfigurasi VFIO untuk menggunakan GPU
nano /etc/modprobe.d/vfio.conf

# Tambahkan baris berikut dengan ID vendor dan device yang
sesuai
options vfio-pci ids=[ID_GPU],[ID_Audio_GPU]
disable_vga=1

# Update initramfs dan restart sistem
update-initramfs -u
reboot
```

Setelah *restart*, GPU akan diisolasi dari *host* dan siap digunakan untuk *passthrough* ke *virtual machine*.

b. Pembuatan dan Konfigurasi Windows VM

Setelah *host* siap, dilakukan pembuatan dan konfigurasi VM Windows:

1. Pembuatan VM dasar:

```
qm create 900 --name windows-render --memory 16384 --cores
8 --sockets 1 --numa 1 --ostype win10 --scsihw virtio-scsi-pci --
net0 model=virtio,bridge=vibr0
```

2. Penambahan penyimpanan:

```
qm set 900 --scsi0 local-lvm:vm-900-disk-0,size=100G,ssd=1
```

3. Konfigurasi GPU *Passthrough*:

```
qm set 900 --hostpci0 0000:01:00.0,pcie=1
```

4. Instalasi Windows dan Driver:

- Instalasi Windows 11 Pro.
- Instalasi `virtio drivers` untuk penyimpanan dan jaringan.
- Instalasi driver NVIDIA terbaru.
- Konfigurasi *remote desktop* untuk akses jarak jauh.

Hasil implementasi VM Windows dengan GPU *passthrough* adalah *workstation* virtual yang memberikan performa rendering yang signifikan lebih baik dibandingkan dengan solusi virtualisasi konvensional. Tim kreatif dapat melakukan *rendering* video dan pengolahan grafis dengan kecepatan yang mendekati *workstation* fisik, namun dengan keuntungan manajemen terpusat dan integrasi dengan infrastruktur server yang ada.

3.2.4.3 Implementasi Sistem Backup dan Redundansi

Untuk memastikan ketersediaan layanan dan keamanan data, diimplementasikan sistem pencadangan dan redundansi yang komprehensif. Sistem dirancang untuk memberikan perlindungan terhadap berbagai skenario kegagalan, dari kegagalan perangkat keras hingga kesalahan pengguna.

a. Strategi Pencadangan

Implementasi strategi pencadangan yang mengadopsi pendekatan yang mencakup keseluruhan level dari *kontainer*:

1. Menggunakan fitur bawaan Proxmox VE untuk melakukan *snapshot* dan pencadangan berkala terhadap seluruh kontainer dan *virtual machine*. Jadwal pencadangan dikonfigurasi sebagai berikut:
 - *Snapshot* harian untuk keseluruhan kontainer dan VM setiap pukul 02.00 WIB.
 - Pencadangan mingguan.
 - Retensi pencadangan selama 30 hari.
2. Implementasi mekanisme pencadangan khusus untuk *database* dengan:
 - Melakukan *dump* otomatis pada keseluruhan database MySQL setiap 6 jam.

- Melakukan *backup* transaction log untuk melihat waktu transaksi *query* tepat pada jamnya.
- Replikasi database dari LXC100 ke LXC101 untuk memberlakukan *high availability*.

b. Implementasi Redundansi

Redundansi diimplementasikan pada beberapa level untuk meminimalisir kegagalan pada titik tertentu:

1. Implementasi RAID 1 pada level perangkat keras untuk memberikan perlindungan terhadap kegagalan *hard drive*. Lalu terdapat konfigurasi ZFS pada *storage pool* yang sehingga dapat melakukan pengecekan integritas data lewat *checksumming*.
2. Layanan kritis seperti *database* dikonfigurasi dengan mekanisme *failover* dengan sistem *database* utama dan replika.

Implementasi sistem pencadangan dan redundansi yang komprehensif ini memberikan perlindungan berlapis terhadap berbagai skenario kegagalan. Pendekatan ini memastikan bahwa data perusahaan terlindungi dan layanan kritis dapat pulih dengan cepat jika terjadi gangguan, meminimalkan dampak terhadap operasional bisnis.

3.2.5 Deployment Layanan dan Aplikasi

Setelah infrastruktur dasar Proxmox VE terkonfigurasi, tahap selanjutnya adalah melakukan *deployment* layanan dan aplikasi yang dibutuhkan untuk mendukung operasional perusahaan. Proses *deployment* dilakukan secara bertahap dengan memperhatikan dependensi antar layanan dan prioritas kebutuhan bisnis.

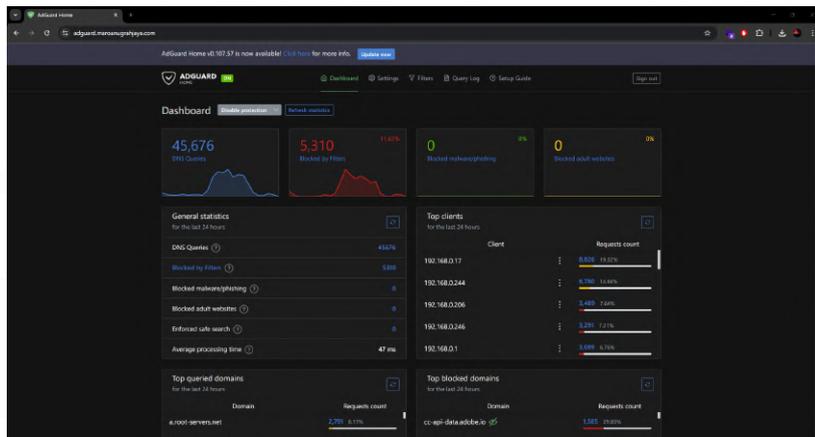
3.2.5.1 Implementasi Layanan Jaringan AdGuard DNS

Implementasi AdGuard DNS pada kontainer LXC 300 dilakukan sebagai komponen penting dalam infrastruktur jaringan untuk meningkatkan keamanan, privasi, dan performa *browsing* di seluruh jaringan kantor perusahaan. AdGuard DNS berfungsi sebagai DNS *resolver* yang menyediakan fitur pemblokiran iklan, *tracker*, dan domain berbahaya pada level jaringan [11].

Proses implementasi AdGuard DNS meliputi:

1. Instalasi dan konfigurasi dasar:
 - Melakukan instalasi AdGuard Home pada kontainer LXC300 menggunakan *script* instalasi resmi.
- ```
curl -s -S -L
https://raw.githubusercontent.com/AdguardTeam/AdGuardHome/master/scripts/install.sh | sh -s -- -v
```
- Konfigurasi untuk mendengarkan port 53 (DNS) dan 3000 (untuk *web interface*)
  2. Konfigurasi *filtering* dengan melakukan aktivasi *blocklist* standar untuk pemblokiran iklan dan *tracker*.
  3. Integrasi dengan infrastruktur, seperti menambahkan implementasi sistem pemantauan menggunakan Prometheus dan Grafana.

Implementasi AdGuard DNS memberikan manfaat signifikan, termasuk peningkatan keamanan jaringan melalui pemblokiran domain berbahaya, peningkatan performa *browsing* dengan mengurangi *loading* iklan, dan visibilitas terhadap *traffic* DNS untuk keperluan analisis dan *troubleshooting*.



Gambar 3.7 Hasil tangkapan layar dari AdGuard Home

Gambar 3.7 Dashboard AdGuard Home menampilkan statistik DNS dan filtering dalam 24 jam terakhir, termasuk jumlah total query DNS, permintaan yang diblokir, serta daftar klien dan domain teratas. Panel statistik juga menunjukkan efektivitas pemblokiran *malware/phishing* dan konten dewasa, dengan waktu pemrosesan rata-rata 72 milidetik per *query*.

### 3.2.5.2 Deployment Database Layer

*Database layer* merupakan komponen kritis yang bisa menjadi fondasi bagi seluruh aplikasi dan layanan. Implementasi *database layer* dirancang dengan mempertimbangkan aspek performa, keandalan, dan skalabilitas. Implementasi yang digunakan untuk server merupakan MariaDB. MariaDB merupakan fork dari MySQL yang menawarkan fitur tambahan dan peningkatan performa [10].

#### 1. Implementasi MariaDB Primary dan Replika

- Instalasi MariaDB versi 10.6 pada kontainer LXC100 dan LXC101 sebagai database utama.
- Konfigurasi kontainer LXC101 sebagai replika dengan replikasi asinkron.
- Implementasi *binary log* untuk *point-in-time recovery*.

- Konfigurasi parameter MariaDB untuk optimasi performa sesuai dengan karakteristik beban kerja.
2. Deployment PHPMyAdmin untuk manajemen Database
    - Instalasi Apache *web server* sebagai prasyarat untuk menjalankan PHPMyAdmin.
    - Instalasi PHP dan ekstensi yang diperlukan.
  3. Deployment Redis Cache
    - Instalasi Redis pada kontainer LXC102
    - Konfigurasi Redis dan *System Control Services*
  4. Manajemen dan Pemantauan
    - Konfigurasi pemantauan khusus untuk database menggunakan Prometheus MySQL Exporter.
    - Pembuatan dashboard Grafana untuk visualisasi metrik *database*.

### 3.2.5.3 Migrasi dan Deployment Aplikasi

Setelah infrastruktur dasar dan database layer siap, dilakukan migrasi dan *deployment* aplikasi-aplikasi yang dibutuhkan untuk mendukung operasional perusahaan. Proses ini dilakukan secara bertahap dengan pendekatan yang terstruktur untuk meminimalisir gangguan pada operasional bisnis.

#### 1. Deployment Nextcloud untuk *File Sharing*

Untuk memenuhi kebutuhan berbagi file dan kolaborasi, diimplementasikan Nextcloud menggunakan *template* TurnKey.

- Pembuatan Kontainer dengan *Template* TurnKey Nextcloud

```
pct create 400 local:vztmpl/turnkey-nextcloud-18.1-bullseye-
amd64.tar.gz --hostname nextcloud --cores 4 --memory 8192 --
swap 2048 --rootfs local-lvm:50 --net0
name=eth0,bridge=vibr0,ip=192.168.0.40/24,gw=192.168.0.1
```

Dengan adanya *template* TurnKey Nextcloud, pengguna hanya perlu melakukan konfigurasi awal, seperti nama, kredensial dan melakukan *setup database*.

- Integrasi dengan ZFS Pool

```
pct set 400 -mp0 /mnt/nextcloud
data,mp=/var/www/nextcloud/data
```

Penambahan *ZFS storage pool* dilakukan supaya Nextcloud menyimpan data user di *ZFS storage* yang sudah ada, bukan menyimpan di SSD.

```
chown -R www-data:www-data /var/www/nextcloud/data
```

*ZFS storage pool* yang sudah di pasangkan juga harus diubah izin aksesnya supaya Nextcloud dapat mengakses *pool*-nya.

```
chown -R www-data:www-data /var/www/nextcloud/data
```

- Migrasi Database

Migrasi database dari bawaan SQLite pada *template* Nextcloud ke MariaDB pada kontainer LXC100. Database yang sudah ada dicadangkan dahulu sebelum memulai.

```
cp /var/www/nextcloud/config/config.php
/var/www/nextcloud/config/config.php.bak
```

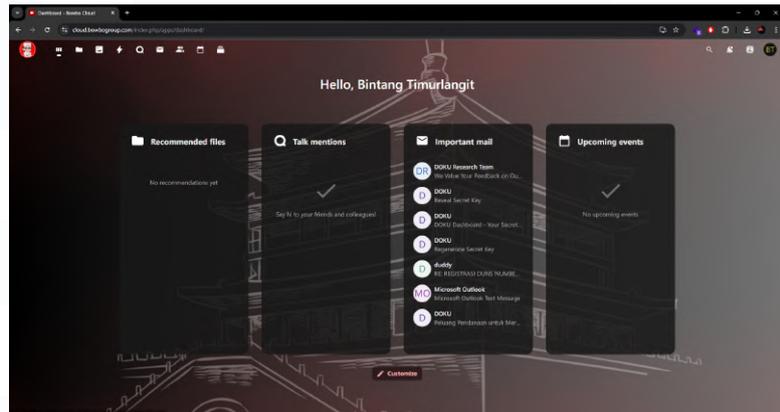
Lalu pada kontaner `database-primary` atau LXC100, dibuatkan database dengan kredensial yang sama.

```
mysql -h 192.168.0.10 -u root -p -e "CREATE DATABASE
nextcloud; GRANT ALL PRIVILEGES ON nextcloud.* TO
'nextcloud'@'%' IDENTIFIED BY 'secure_password';"
```

Setelah terbuat, maka file konfigurasi di Nextcloud disesuaikan agar dapat mengakses database tersebut.

```
// Modifikasi pada file config.php
```

```
'dbtype' => 'mysql',
'dbname' => 'nextcloud',
'dbhost' => '192.168.0.10:3306',
'dbuser' => 'nextcloud',
'dbpassword' => 'secure_password',
```



Gambar 3.8 Hasil tangkapan layar dari Nextcloud

Gambar 3.8 *Dashboard* Nextcloud menampilkan antarmuka pengguna yang intuitif dengan empat panel utama: Recommended files, Talk mentions, Important mail, dan Upcoming events. Interface ini menyediakan akses cepat ke berbagai fitur kolaborasi dan penyimpanan file, dengan desain yang bersih dan navigasi yang mudah melalui menu bar di bagian atas halaman.

## 2. Implementasi ERPNext dengan Frappe

### - Pembuatan *Custom Image Docker*

Pembuatan *custom image* diperlukan karena keperluan operasional kantor yang memerlukan beberapa aplikasi Frappe yang berbeda, sedangkan instalasi manual hanya membawa aplikasi Frappe yaitu ERPNext.

Solusi dari permasalahan ini adalah dengan melakukan pembuatan *image* Docker yang dikhususkan. Docker memungkinkan aplikasi berjalan dalam kontainer yang

terisolasi, memastikan konsistensi di berbagai lingkungan [12]. Dimulai dengan melakukan penambahan aplikasi dengan meng-edit file `apps.json`.

```
[
 {
 "url": "https://github.com/frappe/erpnext",
 "branch": "version-15"
 },
 {
 "url": "https://github.com/frappe/payments",
 "branch": "version-15"
 },
 {
 "url": "https://{{ PAT
}}@git.example.com/project/repository.git",
 "branch": "main"
 }
]
```

Setelah file tersebut dibuat, pengguna melakukan perintah untuk memulai pembuatan *custom image* Docker.

```
docker build \
--build-arg=FRAPPE_PATH=https://github.com/frappe/frappe
--build-arg=FRAPPE_BRANCH=version-15 \
--build-arg=PYTHON_VERSION=3.11.9 \
--build-arg=NODE_VERSION=18.20.2 \
--build-arg=APPS_JSON_BASE64=$APPS_JSON_BASE64 \
--tag=ghcr.io/user/repo/custom:1.0.0 \
--file=images/custom/Kontainerfile .
```

Setelah *custom image* selesai dibuat, pengguna mengubah konfigurasi Docker agar *database* dan *cache* dapat terhubung.

- Pembuatan Custom `.yaml` untuk Docker

a. Perubahan *Image* dan Konfigurasi Dasar

```
Versi Original
image: frappe/erpnext:v15.54.3

Versi Modifikasi
image: bowbogroup/frappe:1.5.0
```

b. Eksternalisasi Database dan Redis

Penghapusan *Service* Database & Redis Internal:

```
Service yang dihapus dari versi original
db:
 image: mariadb:10.6
 networks:
 - frappe_network
 healthcheck:
 test: mysqladmin ping -h localhost --
 password=*****
 interval: 1s
 retries: 20
 deploy:
 restart_policy:
 condition: on-failure
 command:
 - --character-set-server=utf8mb4
 - --collation-server=utf8mb4_unicode_ci
 - --skip-character-set-client-handshake
 - --skip-innodb-read-only-compressed
 environment:
 MYSQL_ROOT_PASSWORD: *****
 MARIADB_ROOT_PASSWORD: *****
 volumes:
 - db-data:/var/lib/mysql
```

```

Service yang dihapus dari versi original
redis-queue:
 image: redis:6.2-alpine
 networks:
 - frappe_network
 deploy:
 restart_policy:
 condition: on-failure
 volumes:
 - redis-queue-data:/data

redis-cache:
 image: redis:6.2-alpine
 networks:
 - frappe_network
 deploy:
 restart_policy:
 condition: on-failure

```

Perubahan signifikan ini menunjukkan migrasi dari database dan Redis yang di-*deploy* sebagai kontainer dalam *stack* yang sama, menjadi penggunaan layanan eksternal yang telah di-*deploy* sebelumnya pada kontainer LXC100 (MariaDB) dan LXC102 (Redis).

Pendekatan ini memberikan beberapa keuntungan:

- Memungkinkan penggunaan database yang sama untuk berbagai aplikasi.
- Memudahkan manajemen dan pencadangan *database* secara terpusat.
- Meningkatkan performa dengan menggunakan kontainer yang dioptimasi khusus untuk *database*.

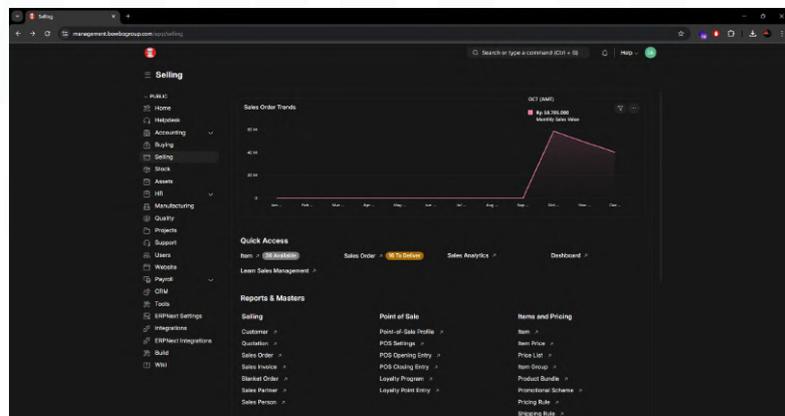
### c. Modifikasi Service `create-site`

```
Versi Original
bench new-site --mariadb-user-host-login-scope=%' --
admin-password=admin --db-root-username=root --db-root-
password=admin --install-app erpnext --set-default frontend;

Versi Modifikasi
bench new-site --mariadb-user-host-login-scope=%' --
admin-password=admin --db-root-username=root --db-root-
password=***** --install-app erpnext --install-app hrms
--install-app wiki --install-app helpdesk --install-app
print_designer --set-default frontend;
```

Modifikasi pada *command* `create-site` menunjukkan penambahan beberapa aplikasi Frappe tambahan yang diinstal bersamaan dengan ERPNext:

- hrms: Modul Human Resource Management System.
- wiki: Modul untuk dokumentasi internal dan *knowledge base*.
- helpdesk: Modul untuk manajemen tiket dan dukungan pelanggan.
- print\_designer: Modul untuk kustomisasi format cetak dokumen.



Gambar 3.9 Hasil tangkapan layar dari ERPNext

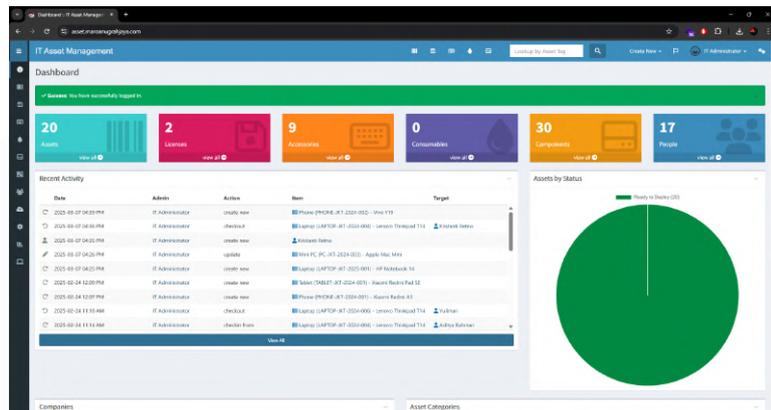
Gambar 3.9 *Dashboard* ERPNext yang diimplementasikan menggunakan *custom Docker image*, menampilkan antarmuka modul Selling dengan navigasi komprehensif di sidebar kiri mencakup berbagai modul seperti *Helpdesk*, *Accounting*, HR, dan *Manufacturing*. Panel utama menampilkan *Quick Access* untuk fungsi-fungsi umum serta Reports & Masters yang terbagi dalam tiga kategori: Selling, Point of Sale, dan Items and Pricing. Interface yang modern dengan *layout* yang terorganisir untuk kemudahan navigasi dan penggunaan.

### 3. Deployment Aplikasi SnipeIT

Instalasi SnipeIT diperlukan untuk melakukan manajemen asset tim IT sendiri yang sudah menumpuk, hal ini memudahkan koordinasi barang-barang yang keluar masuk untuk dipinjam sebagai asset kantor.

TurnKey juga sudah menyediakan *bootable image* untuk kontainer supaya dapat melakukan instalasi SnipeIT dengan mudah.

- Download *template* yang sudah disediakan pada halaman resmi TurnKey.
- Pada *first boot*, aplikasi akan terinstall dengan sendirinya.
- Pengguna hanya perlu melakukan konfigurasi kredensial admin untuk pertama kali.



Gambar 3.10 Hasil tangkapan layar dari SnipeIT

Gambar 3.10 *Dashboard* SnipeIT menampilkan ringkasan komprehensif manajemen aset IT perusahaan dengan enam kategori utama: *Assets*, *Licenses*, *Accessories*, *Consumables*, *Components*, dan *People*. Panel *Recent Activity* menunjukkan *log* aktivitas terbaru seperti pembuatan aset baru dan *checkout* perangkat, sementara grafik pie di sebelah kanan memvisualisasikan status aset dengan dominasi status "Ready to Deploy".

#### 4. Deployment Aplikasi Passbolt

Banyaknya berbagai kata sandi perusahaan yang harus disimpan, menyimpannya dalam *file* atau *notepad* bukanlah menjadi suatu langkah yang bijak. Maka dari itu, instalasi Passbolt dilakukan supaya memudahkan manajemen kata sandi. Passbolt menyediakan *script* untuk instalasi dengan mudah mengikuti *link* yang sudah disediakan pada website resminya.

Mengunduh *script* instalasi yang sudah disediakan:

```
curl -LO https://download.passbolt.com/ce/installer/passbolt-repo-setup.ce.sh
```

Mengunduh file *checksum* SHA512 untuk verifikasi integrasi file instalasi.

```
curl -LO https://github.com/passbolt/passbolt-dep-
scripts/releases/latest/download/passbolt-ce-SHA512SUM.txt
```

Instalasi aplikasi Passbolt,

```
sha512sum -c passbolt-ce-SHA512SUM.txt && sudo bash
./passbolt-repo-setup.ce.sh || echo "Bad checksum. Aborting" && rm
-f passbolt-repo-setup.ce.sh
```

## 5. Deployment Proyek pada *Backend* dan *Frontend*

Proses migrasi dan *deployment* aplikasi dilakukan dengan pendekatan yang terstruktur, dimulai dari aplikasi non-kritis dan secara bertahap menuju aplikasi yang lebih kritis. Pendekatan ini memungkinkan identifikasi dan resolusi masalah pada tahap awal, serta memberikan waktu adaptasi bagi pengguna terhadap sistem baru.

Hasil dari *deployment* layanan dan aplikasi ini adalah infrastruktur digital yang terintegrasi dan terpusat, yang mendukung seluruh aspek operasional perusahaan, dari manajemen dokumen hingga proses bisnis inti.

### 3.2.6 Monitoring dan Pemeliharaan Sistem

Implementasi sistem pemantauan dan pemeliharaan merupakan aspek kritis untuk memastikan keandalan, performa, dan keamanan infrastruktur server. Pendekatan proaktif dalam pemantauan dan pemeliharaan memungkinkan identifikasi dini terhadap potensi masalah sebelum berdampak pada operasional bisnis.

#### 3.2.6.1 Implementasi *Stack Monitoring*

Untuk memastikan visibilitas komprehensif terhadap seluruh aspek infrastruktur, diimplementasikan *stack monitoring* berbasis Prometheus dan Grafana pada kontainer LXC301:

1. Prometheus

Diimplementasikan sebagai *time-series database* untuk pengumpulan dan penyimpanan metrik dari berbagai komponen infrastruktur.

```
apt update && apt install -y prometheus
```

2. Node Exporter

Dipasang pada setiap kontainer dan host Proxmox untuk mengumpulkan metrik sistem dasar seperti CPU, memori, disk I/O, dan *network traffic*.

```
apt install -y prometheus-node-exporter
```

3. Grafana

Sebagai platform visualisasi untuk membuat *dashboard* interaktif yang menampilkan metrik dari Prometheus.

```
apt install -y grafana
```

4. Alertmanager

Dikonfigurasi untuk mengirimkan notifikasi ketika kondisi tertentu terpenuhi, seperti penggunaan disk yang tinggi atau layanan yang tidak responsif.

```
apt install -y prometheus-alertmanager
```

### 3.2.6.2 Dashboard dan Visualisasi

Beberapa dashboard Grafana dikonfigurasi untuk memberikan visibilitas terhadap berbagai aspek infrastruktur:

1. Dashboard *Host System*: Menampilkan metrik host Proxmox, termasuk penggunaan CPU, memori, disk I/O, dan *network traffic*.
2. Dashboard Kontainer: Memberikan visibilitas terhadap performa setiap *kontainer*, memungkinkan identifikasi kontainer yang mengkonsumsi daya komputasi berlebihan.

3. Dashboard *Database*: Menampilkan metrik spesifik database seperti *query per second*, *slow queries*, *connection count*, dan replikasi lag.

### 3.2.6.3 Implementasi Alerting

Sistem *alerting* dikonfigurasi untuk memberikan notifikasi proaktif ketika terjadi kondisi yang memerlukan perhatian:

1. Konfigurasi *threshold* untuk berbagai metrik, seperti:
  - Penggunaan CPU lebih dari 80 persen selama lebih dari 5 menit.
  - Penggunaan memori lebih dari 90 persen.
  - Penggunaan disk lebih dari 85 persen.
  - *Service downtime* lebih dari 1 menit.
2. Implementasi *notification channel* untuk memastikan alert diterima tepat waktu melalui email.

## 3.3 Kendala yang Ditemukan

Selama proses implementasi *private cloud infrastructure* menggunakan Proxmox VE di PT Maro Anugrah Jaya, beberapa kendala teknis dan operasional ditemukan:

### 1. Pembatasan Akses Port Internet

Penyedia layanan internet (ISP) yang digunakan perusahaan menerapkan pembatasan akses pada beberapa port jaringan. Hal ini menyebabkan kesulitan mengekspos layanan tertentu ke internet, terutama untuk layanan yang memerlukan port spesifik seperti SSH (22), HTTPS (443), dan beberapa port kustom untuk aplikasi bisnis. Pembatasan ini menghambat implementasi akses jarak jauh yang aman dan terenkripsi ke infrastruktur server.

## 2. Kompatibilitas Hardware dengan GPU Passthrough

Implementasi GPU *passthrough* untuk VM Windows mengalami beberapa kendala teknis. Meskipun perangkat keras server mendukung teknologi virtualisasi dan IOMMU, konfigurasi GPU *passthrough* tidak berfungsi sebagaimana mestinya pada percobaan awal.

Beberapa masalah yang ditemui meliputi:

- VM tidak dapat mengakses GPU secara langsung meskipun konfigurasi VFIO sudah diterapkan.
- Error "IOMMU group" yang mengindikasikan masalah pada isolasi perangkat PCI.
- Konflik antara *driver host* dan konfigurasi *passthrough*.
- VM mengalami *crash* saat mencoba mengakses GPU.

## 3. Kesulitan untuk pemasangan aplikasi Frappe Apps

Implementasi ERPNext dan aplikasi Frappe lainnya menghadapi beberapa tantangan teknis:

- Pendekatan awal dengan instalasi manual aplikasi di dalam kontainer terbukti tidak efisien.
- Proses instalasi manual membutuhkan waktu yang lama dan sering mengalami kegagalan.
- Dependensi yang kompleks antara berbagai komponen Frappe *framework* sulit dikelola secara manual.
- Konflik versi antara aplikasi yang berbeda dalam ekosistem Frappe.
- Kesulitan dalam mengintegrasikan aplikasi tambahan seperti HRMS, Wiki, dan Helpdesk dengan ERPNext *core*.

### 3.4 Solusi atas Kendala yang Ditemukan

#### 1. Solusi untuk Keterbatasan Port Internet

Untuk mengatasi pembatasan port oleh ISP, beberapa pendekatan diterapkan Cloudflare Tunneling:

- Menggunakan Cloudflare Tunnel untuk mengekspos layanan internal tanpa memerlukan *port forwarding* tradisional.
- Konfigurasi Cloudflare Argo Tunnel untuk membuat koneksi outbound yang aman dari server ke jaringan Cloudflare.
- Implementasi autentikasi Cloudflare Access untuk menambahkan lapisan keamanan tambahan.

#### 2. Solusi untuk Kompatibilitas GPU *Passthrough*

Setelah melakukan serangkaian *troubleshooting* dan eksperimen, GPU *passthrough* berhasil diimplementasikan dengan pendekatan berikut:

- Penyesuaian Konfigurasi BIOS dengan mengaktifkan fitur virtualisasi lanjutan di BIOS server.
- Mengaktifkan *Access Control Services (ACS)* untuk isolasi perangkat PCI yang lebih baik.
- Menonaktifkan fitur yang dapat mengganggu IOMMU seperti SR-IOV dan AER.

Penyempurnaan Konfigurasi Kernel:

- Menerapkan parameter kernel tambahan:  
``pcie_acs_override=downstream,multifunction``
- Menggunakan kernel versi yang lebih baru yang memiliki dukungan IOMMU yang lebih baik.
- Implementasi *patch ACS override* untuk mengatasi keterbatasan perangkat keras.

Isolasi GPU yang Tepat:

- Mengidentifikasi semua perangkat dalam IOMMU group yang sama dengan GPU.
- Melakukan *passthrough* seluruh grup IOMMU, bukan hanya GPU.
- Memastikan *audio controller* GPU juga di-*passthrough* bersama dengan GPU.

### 3. Solusi untuk Pemasangan Aplikasi Frappe Apps

Untuk mengatasi kesulitan dalam implementasi aplikasi Frappe, pendekatan kontainerisasi diberlakukan dengan langkah berikut:

- Mengembangkan *custom Docker image* yang mencakup semua aplikasi Frappe yang diperlukan.
- Menggunakan Docker Compose untuk mengelola dependensi dan konfigurasi.
- Implementasi *multi-stage build* untuk mengoptimalkan ukuran *image*.

