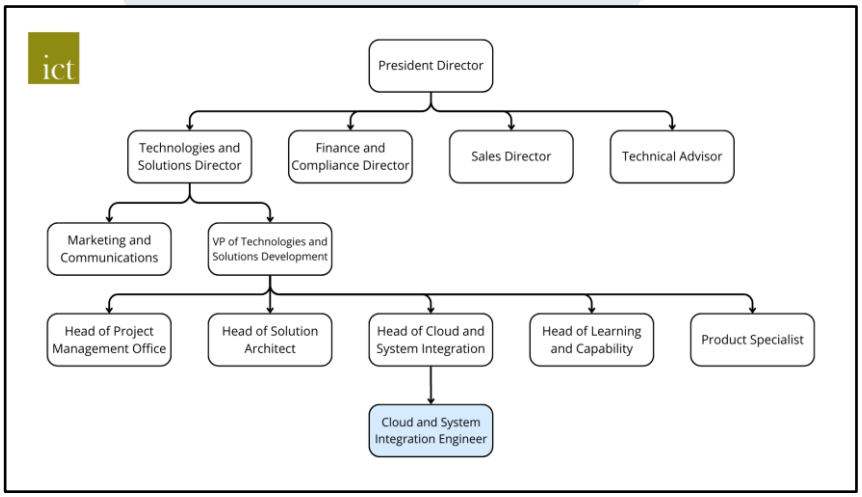
BAB III
PELAKSANAAN KERJA

3.1 Kedudukan dan Koordinasi

Selama program kerja magang di PT InfraCom Technology, posisi yang diberikan adalah sebagai anggota tim *Cloud & System Integration Engineer Intern*.

3.1.1 Kedudukan

Penulis menempati posisi sebagai anggota tim yang berada di bawah divisi atau unit yang menangani terkait penyimpanan dan integrasi sistem. Kedudukan ini menempatkan penulis sebagai bagian dari struktur organisasi yang bertanggung jawab mendukung operasional sistem, integrasi layanan, serta pengembangan infrastruktur TI. Bagan struktur organisasi divisi yang relevan ditampilkan pada gambar berikut.



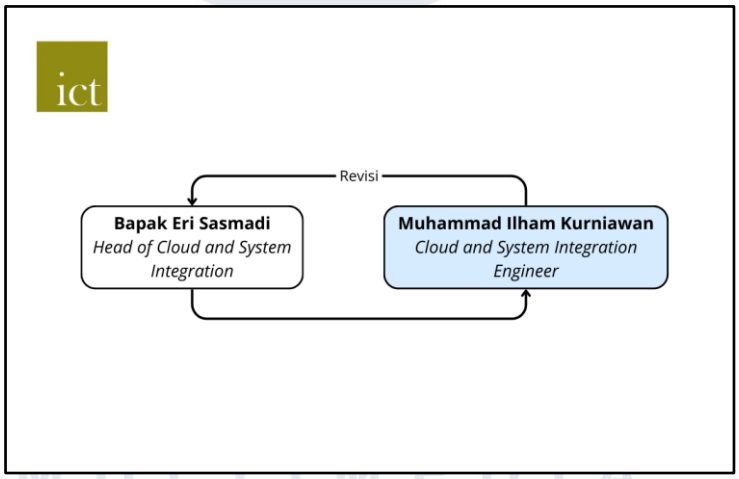
Gambar 3.1 Struktur Organisasi PT Infracom Technology

Gambar 3.1 menggambarkan ruang lingkup tugas yang menjadi tanggung jawab utama, yaitu memberikan dukungan terhadap integrasi infrastruktur cloud yang digunakan perusahaan. Tanggung jawab ini tidak hanya mencakup proses integrasi semata, tetapi juga memastikan bahwa seluruh komponen sistem dapat berfungsi secara optimal dan saling terhubung dengan baik. Selain itu, peran ini melibatkan upaya berkelanjutan untuk meningkatkan efisiensi serta stabilitas sistem, sehingga performa

layanan teknologi informasi dapat terus ditingkatkan. Dukungan yang diberikan juga berkontribusi langsung pada kelancaran berbagai aktivitas operasional dan bisnis perusahaan, karena infrastruktur TI yang andal menjadi pondasi penting bagi setiap proses kerja yang berjalan. Dengan demikian, tugas tersebut berperan strategis dalam menjaga keberlangsungan layanan dan mendukung transformasi digital perusahaan secara keseluruhan.

3.1.2 Koordinasi

Alur koordinasi selama masa magang disusun agar proses kerja berjalan efektif, terarah, dan kolaboratif. Setiap aktivitas yang dilakukan mengikuti mekanisme koordinasi yang melibatkan hubungan vertikal dengan atasan maupun hubungan horizontal dengan rekan satu tim, sehingga setiap tugas dapat diselesaikan sesuai target dan standar yang telah ditetapkan. Koordinasi vertikal dilakukan melalui komunikasi langsung dengan *Head of Cloud & System Integration* serta para *engineer* senior yang membimbing penulis dalam menjalankan tugas.



Gambar 3.2 Bagan Alur Koordinasi

Arahan kerja, pembagian tanggung jawab, serta penjelasan teknis diberikan secara berkesinambungan. Penulis juga melaporkan perkembangan pekerjaan, kendala yang muncul, serta hasil yang telah dicapai secara berkala untuk memastikan seluruh aktivitas tetap selaras dengan rencana tim. Umpan balik dari atasan menjadi dasar perbaikan dan

penyesuaian langkah berikutnya. Di sisi lain, koordinasi horizontal berlangsung melalui kerja sama harian dengan anggota tim *Cloud & System Integration Engineer* lainnya. Kolaborasi ini mencakup diskusi teknis, berbagi informasi terkait temuan atau permasalahan sistem, serta penyelarasan pekerjaan yang saling berkaitan. Pola kerja seperti ini tidak hanya membantu mempercepat penyelesaian tugas, tetapi juga menciptakan lingkungan kerja yang suportif dan terpadu. Seluruh kegiatan komunikasi dan kerja sama dalam tim memanfaatkan berbagai media kolaboratif agar proses koordinasi berjalan efektif. Pertemuan langsung di kantor biasanya digunakan untuk membahas hal-hal strategis dan mendapatkan arahan teknis secara lebih mendalam. Sementara itu, komunikasi operasional sehari-hari banyak dilakukan melalui platform digital seperti Microsoft Teams dan WhatsApp, yang digunakan untuk menyampaikan pembaruan tugas, mengkoordinasikan pekerjaan, hingga menangani masalah teknis secara cepat. Dengan dukungan alat-alat tersebut, interaksi antar anggota tim tetap terjaga dengan baik meskipun berada di lokasi yang berbeda.

3.2 Tugas yang Dilakukan

Tugas utama yang diberikan selama masa magang adalah berperan dalam mendukung pengerjaan kebutuhan teknis terkait layanan cloud computing yang diajukan oleh klien. Namun, sebelum terlibat langsung dalam aktivitas tersebut, penulis terlebih dahulu mengikuti tahap penguatan kompetensi. Pada tahap ini, perusahaan memberikan waktu sekitar satu setengah bulan untuk mempelajari konsep, layanan, serta praktik teknis yang berkaitan dengan cloud computing. Proses belajar ini mencakup pendalaman materi dasar hingga pemahaman mengenai implementasi layanan cloud di lingkungan perusahaan. Setelah periode belajar mandiri dan bimbingan teknis tersebut selesai, penulis diwajibkan mengikuti ujian sertifikasi resmi sebagai bentuk validasi terhadap kompetensi yang telah diperoleh. Setelah dinyatakan memenuhi standar melalui sertifikasi, barulah penulis mulai dilibatkan dalam berbagai tugas teknis yang berkaitan langsung dengan kebutuhan klien. Aktivitas ini meliputi asistensi dalam konfigurasi layanan cloud, analisis kebutuhan infrastruktur, hingga dukungan teknis pada proses

integrasi sistem. Semua tugas yang dikerjakan selama masa magang dirangkum secara ringkas dalam Tabel 3.1 sebagai dokumentasi formal kegiatan yang telah dilaksanakan selama periode magang.

Table 3.1 Tugas Kerja Magang

No	Tugas	Tanggal Mulai	Tanggal Selesai
1	Pembelajaran Cloud Lanjutan	16 Juni 2025	10 September 2025
	AWS	16 Juni 2025	31 Agustus 2025
	OCI	1 Juli 2025	10 September 2025
2	Project Monitoring Service Klien	9 Juli 2025	9 September 2025
	Laporan Bulanan	9 Juli 2025	9 September 2025
	Laporan Preventive Maintenance	9 Juli 2025	9 September 2025
3	Project Backup Otomatis	20 Juni 2025	31 Agustus 2025
	Implementasi Backup Otomatis	20 Juni 2025	31 Agustus 2025

3.3 Uraian Pelaksanaan Kerja

Bagian ini berisi gambaran umum mengenai berbagai aktivitas, tanggung jawab, dan proses kerja yang dijalankan penulis selama periode magang.

3.3.1 Proses Pelaksanaan

Pada bagian ini dijelaskan tahapan-tahapan yang dijalani selama pelaksanaan magang, mulai dari proses pembelajaran dasar hingga keterlibatan langsung dalam kegiatan teknis.

3.3.1.1 Pembelajaran Cloud

Kegiatan pembelajaran mandiri menjadi salah satu tahap krusial dalam proses pengembangan kompetensi selama masa magang, terutama pada bidang cloud computing yang berkembang sangat cepat dan menuntut pemahaman teknis yang kuat. Tahap ini dilaksanakan secara sistematis untuk memperdalam pemahaman konsep, meningkatkan kemampuan analisis, serta mengasah

keterampilan praktis yang diperlukan dalam mendukung kebutuhan proyek perusahaan. Melalui proses ini, pengetahuan dasar yang dimiliki sebelumnya diperluas dan diperkuat sehingga penulis dapat bekerja dengan lebih efektif, terarah, dan profesional saat terlibat dalam kegiatan operasional maupun teknis.

Pembelajaran mandiri dilakukan dengan memanfaatkan berbagai sumber terpercaya, seperti dokumentasi resmi AWS, Google Cloud, serta referensi teknis lainnya yang memberikan penjelasan komprehensif mengenai fitur, konfigurasi, dan praktik terbaik dalam penggunaan layanan cloud. Selain itu, penulis juga mengikuti berbagai kursus online melalui platform seperti AWS Skill Builder, Udemy, YouTube, dan CloudAcademy untuk memperoleh pemahaman yang lebih terstruktur terkait konsep fundamental, arsitektur layanan, serta proses implementasi cloud pada skenario nyata. Seluruh materi tersebut kemudian diperdalam melalui praktik langsung atau hands-on lab, mencakup simulasi deployment, konfigurasi monitoring, automasi proses backup, hingga pengujian keamanan dasar untuk memastikan penguasaan konsep dapat diterapkan secara nyata.

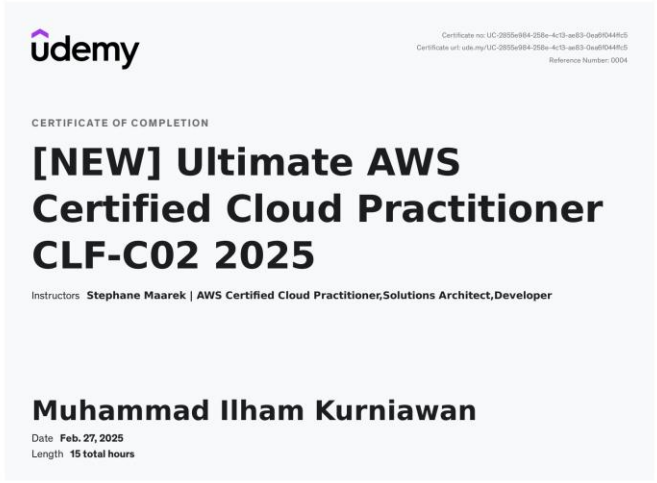
Melalui rangkaian kegiatan ini, penulis dapat mempersiapkan diri dengan lebih matang sebelum menangani tugas-tugas proyek, baik yang bersifat operasional seperti pemantauan sistem dan penyusunan laporan, maupun yang lebih teknis seperti automasi alur kerja dan optimasi konfigurasi layanan. Proses pembelajaran mandiri ini tidak hanya meningkatkan kemampuan teknis, tetapi juga menumbuhkan kemandirian, ketelitian, dan kemampuan berpikir kritis dalam menghadapi dinamika kebutuhan proyek di lingkungan kerja. Selain itu, materi pembelajaran juga diarahkan untuk memahami karakteristik berbagai penyedia layanan cloud, mencakup konsep arsitektur, model layanan, mekanisme keamanan, serta

praktik implementasi yang berbeda-beda di setiap platform, sehingga penulis memiliki perspektif yang lebih luas dan mampu beradaptasi dengan kebutuhan sistem berbasis *multi-cloud*.

1. **Amazon Web Services (AWS)**

   Mempelajari layanan-layanan fundamental dari AWS dilakukan secara sistematis dan bertahap dengan tujuan membangun pemahaman yang komprehensif, mulai dari konsep dasar hingga kemampuan merancang solusi yang lebih kompleks. Setiap tahapan pembelajaran disusun untuk saling melengkapi sehingga dapat menjawab berbagai kebutuhan dan tantangan teknis yang muncul selama periode magang. Pendekatan bertahap ini juga membantu dalam membangun fondasi yang kuat sebelum mempelajari layanan yang lebih spesifik, seperti monitoring, automasi, maupun arsitektur cloud tingkat lanjut. Langkah awal dalam memasuki ekosistem AWS dimulai dari persiapan untuk memperoleh sertifikasi AWS Certified Cloud Practitioner (CLF-C02). Tahap ini menjadi pondasi penting karena tidak hanya memberikan gambaran mengenai fungsi dan kategori layanan AWS, tetapi juga menekankan mengapa dan bagaimana layanan tersebut membawa nilai tambah bagi kebutuhan bisnis modern. Pembelajaran dilakukan melalui kursus online di Udemy yang secara terstruktur membahas empat domain utama, yaitu konsep cloud computing, aspek keamanan dan kepatuhan, teknologi inti AWS, serta mekanisme penagihan dan pengelolaan biaya. Melalui pendekatan ini, penulis mendapatkan pemahaman yang menyeluruh sekaligus praktis mengenai bagaimana layanan AWS dikelola, dioperasikan, serta dioptimalkan. Pada proses pembelajarannya, salah satu konsep fundamental yang dipelajari adalah AWS Shared Responsibility Model, yaitu model tanggung jawab bersama

antara AWS dan pelanggan. AWS bertanggung jawab atas keamanan of the cloud, mencakup infrastruktur fisik, hardware, jaringan, hingga fasilitas global. Sementara itu, pelanggan bertanggung jawab atas keamanan in the cloud, seperti konfigurasi layanan, manajemen identitas dan akses, enkripsi data, serta pengaturan firewall. Selain itu, pemahaman terhadap struktur harga dan alat pendukung seperti AWS Cost Explorer, Budgets, dan Pricing Calculator menjadi aspek penting karena efisiensi biaya merupakan salah satu pilar utama dalam adopsi cloud. Dengan memahami mekanisme biaya sejak awal, pengguna dapat merancang solusi yang tidak hanya berjalan dengan baik tetapi juga optimal dari sisi pengeluaran.


udemy

CERTIFICATE OF COMPLETION

[NEW] Ultimate AWS Certified Cloud Practitioner CLF-C02 2025

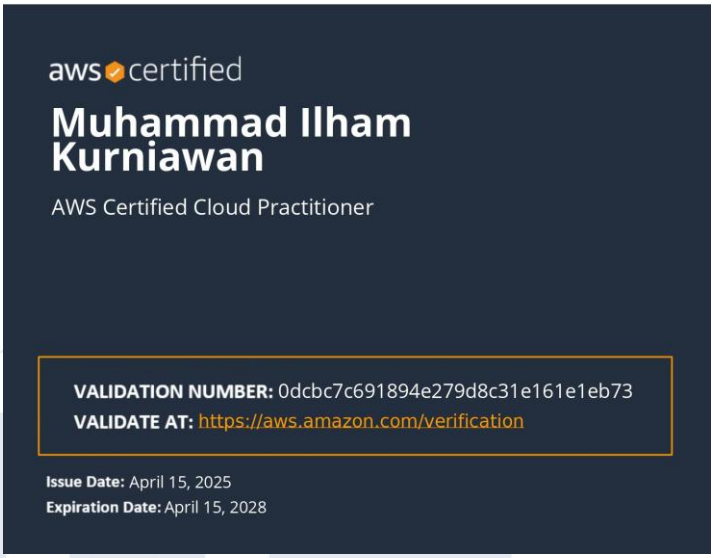
Instructors Stephane Maarek | AWS Certified Cloud Practitioner,Solutions Architect,Developer

Muhammad Ilham Kurniawan

Date Feb. 27, 2025
Length 15 total hours

Gambar 3.3 Sertifikat Udemy AWS Certified Cloud Practitioner

Gambar 3.4 AWS Certified Cloud Practitioner Foundational



Gambar 3.5 Sertifikat AWS Cloud Practitioner Foundational

Tahapan ini diakhiri dengan keberhasilan lulus ujian sertifikasi resmi AWS *Certified Cloud Practitioner*. Sertifikasi ini menjadi validasi formal atas pengetahuan dasar yang solid dan membangun kepercayaan diri untuk melangkah ke topik yang lebih kompleks, dengan bukti sertifikat bisa dilihat pada Gambar 3.5. Setelah memiliki pondasi yang kuat, *supervisor* memberikan tugas untuk mencoba melakukan hands on lab dengan membuat *3-tier Architecture* dengan dimulai membuat dari sisi *network, security, virtual machine,* dan *database*. dengan *design architecture* sebagai berikut. Pembelajaran yang dilakukan bertujuan untuk meningkatkan kemampuan yang tidak hanya mengeksekusi tugas, tetapi juga berkontribusi pada diskusi desain arsitektur, memberikan rekomendasi, dan pada akhirnya memberikan nilai lebih bagi tim dan perusahaan.

2. **Oracle Cloud Infrastructure (OCI)**

Sebagai langkah untuk memperluas kompetensi di bidang multi-cloud, penulis juga mempelajari layanan dan arsitektur Oracle Cloud Infrastructure (OCI) sebagai pelengkap dari pemahaman yang telah dibangun pada platform cloud lainnya. Pendalaman materi mengenai OCI dilakukan karena platform ini banyak digunakan oleh perusahaan berskala besar, khususnya untuk kebutuhan sistem dengan performa tinggi, seperti pengelolaan database kritikal, proses transaksi berskala besar, serta aplikasi yang menuntut tingkat latency rendah dan stabil. Selain itu, OCI dikenal memiliki pendekatan arsitektur yang berfokus pada konsistensi performa, keamanan yang tersegmentasi dengan baik, serta desain jaringan yang lebih sederhana melalui konsep flat network. Melalui proses pembelajaran ini, penulis bertujuan untuk memahami nilai lebih yang ditawarkan OCI serta melakukan perbandingan secara objektif terkait perbedaan arsitektur, kinerja, dan model layanan di antara berbagai penyedia cloud utama. Dengan demikian, kemampuan analitis dan pemahaman teknis dalam konteks lingkungan multi-cloud dapat berkembang lebih komprehensif. Seluruh proses pembelajaran dilakukan melalui Oracle MyLearn, yakni platform edukasi resmi yang menyediakan materi pelatihan komprehensif dengan alur pembelajaran yang terstruktur dan fleksibel untuk dipelajari secara mandiri. Dalam proses ini, penulis mengikuti kurikulum Oracle Cloud Infrastructure Foundations, sebuah program yang dirancang untuk membangun pemahaman mendasar mengenai layanan inti OCI, baik dari sisi konsep maupun penerapannya. Materi yang dipelajari mencakup aspek jaringan, komputasi, penyimpanan, keamanan, manajemen identitas, serta prinsip desain arsitektur yang digunakan OCI. Pembelajaran juga

menekankan pada pemetaan konsep antara OCI dan penyedia cloud lain sehingga wawasan yang diperoleh dapat memperkaya kemampuan dalam melakukan service mapping, mengenali kesetaraan fitur, serta memahami perbedaan pendekatan arsitektur antar platform cloud. Setelah menyelesaikan seluruh modul pada Oracle MyLearn, penulis melanjutkan ke tahap sertifikasi resmi yang diselenggarakan oleh Oracle. Proses persiapan mencakup latihan soal, peninjauan ulang konsep inti, dan praktik langsung melalui hands-on lab untuk memperkuat pemahaman. Dengan persiapan tersebut, penulis berhasil lulus ujian dan memperoleh sertifikasi Oracle Cloud Infrastructure 2025 Foundations Associate. Pencapaian ini menjadi validasi atas penguasaan dasar layanan OCI sekaligus memperkuat kompetensi sebagai praktisi yang mampu beroperasi dalam ekosistem multi-cloud, sebagaimana ditunjukkan pada Gambar 3.6.

Gambar 3.6 Sertifikat Oracle Certified Foundations Associate

Gambar 3.7 Sertifikat Oracle Certified Architect Associate

Gambar 3.8 Sertifikat Oracle Certified Migration Architect Professional

Pencapaian sertifikasi ini tidak hanya menandai keberhasilan menyelesaikan rangkaian materi pembelajaran, tetapi juga menjadi bukti nyata atas komitmen penulis dalam memperluas kemampuan profesional di bidang *cloud computing*. Setelah merampungkan program pelatihan dan memperoleh sertifikasi *Oracle Cloud Infrastructure 2025 Foundations Associate* sebagai dasar kompetensi, proses pengembangan kemudian

dilanjutkan ke tingkat lanjutan untuk memperdalam pemahaman mengenai arsitektur, operasi, dan implementasi teknologi cloud pada skala enterprise. Tahap berikutnya dilakukan dengan memperkuat penguasaan konsep melalui sertifikasi *Oracle Cloud Infrastructure Architect Associate*. Sertifikasi ini memvalidasi kemampuan dalam merancang solusi cloud yang aman, elastis, memiliki skalabilitas tinggi, serta mengikuti *best practice* arsitektur OCI. Materi yang diujikan mencakup pengelolaan jaringan tingkat lanjut, perencanaan kapasitas, konfigurasi komponen komputasi dan penyimpanan, serta penerapan kontrol keamanan yang sejalan dengan prinsip *zero trust*. Pengembangan kompetensi kemudian diteruskan dengan meraih sertifikasi *Oracle Cloud Infrastructure Migration Architect*, yang berfokus pada perancangan dan pelaksanaan proses migrasi beban kerja dari sistem *on-premise* maupun platform cloud lain menuju OCI. Pada tahap ini, penulis menguasai kemampuan melakukan asesmen kebutuhan, menentukan strategi migrasi yang tepat, merancang proses *cutover* serta *downtime plan*, hingga melakukan validasi hasil migrasi untuk memastikan sistem berjalan optimal. Melalui proses pembelajaran berjenjang tersebut, mulai dari Foundations, dilanjutkan ke Architect Associate, dan kemudian Migration Specialist, kompetensi yang dibangun tidak hanya berhenti pada pemahaman teori, tetapi juga berkembang menjadi keterampilan praktis yang siap diterapkan dalam konteks proyek nyata. Ditambah dengan penguasaan fundamental AWS yang telah dimiliki sebelumnya, penulis memperoleh perspektif yang lebih luas terhadap implementasi solusi multi-cloud. Kemampuan memahami kelebihan, batasan, serta karakteristik teknis dari masing-masing cloud provider menjadi nilai strategis dalam

mendukung kebutuhan industri modern, yang semakin banyak mengadopsi pendekatan multi-cloud untuk meningkatkan fleksibilitas, efisiensi, dan keandalan layanan TI. Keseluruhan pencapaian ini mencerminkan bahwa proses pengembangan kompetensi dilakukan secara sistematis dan konsisten, sehingga penulis lebih siap untuk berkontribusi dalam tugas-tugas teknis yang lebih kompleks, seperti perancangan arsitektur, optimalisasi performa, desain *disaster recovery*, hingga migrasi cloud berskala besar.

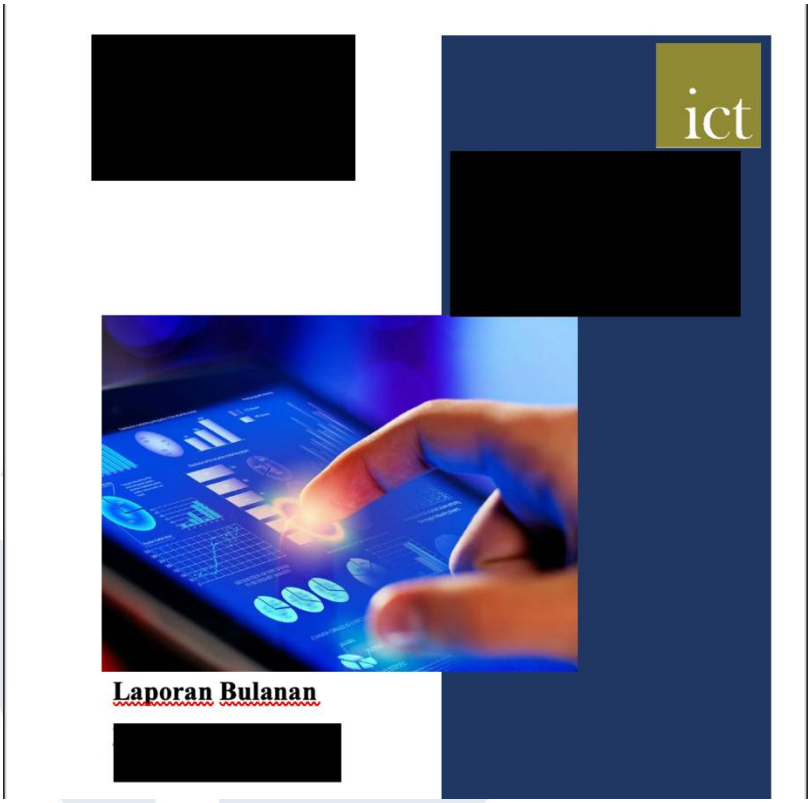
3.3.1.2 Project Monitoring Services Client

Setelah memahami konsep dasar cloud computing dan mekanisme kerja layanan berbasis cloud, saya kemudian terlibat secara aktif dalam proyek pemantauan layanan (service monitoring) untuk salah satu klien perusahaan. Proyek ini bertujuan memastikan bahwa layanan, aplikasi, dan infrastruktur yang digunakan klien selalu berada dalam kondisi optimal, baik dari aspek ketersediaan (availability), kinerja (performance), maupun keandalan (reliability). Pemantauan layanan menjadi komponen penting dalam operasional teknologi informasi karena berperan dalam mendeteksi potensi gangguan secara dini, menyediakan data historis untuk analisis tren, serta mendukung pengambilan keputusan strategis terkait kapasitas maupun perencanaan perbaikan sistem. Dalam pelaksanaan kegiatan tersebut, saya diberikan tanggung jawab untuk membantu menyusun *monthly report* dan *preventive maintenance report* sebagai bagian dari siklus pemantauan rutin sesuai kesepakatan layanan dengan klien.

1. ***Monthly Report***

   Laporan Bulanan merupakan dokumen analitis yang memuat rangkuman menyeluruh mengenai performa dan kondisi layanan klien selama periode satu bulan. Tidak seperti laporan

mingguan yang umumnya bersifat operasional dan taktis, laporan bulanan disusun dengan fokus pada perspektif strategis untuk memberikan wawasan mendalam kepada pihak manajemen, tim teknis senior, maupun klien. Isi laporan meliputi analisis tren penggunaan sumber daya, pola kejadian (incident patterns), perbandingan performa antar periode, serta evaluasi metrik kunci seperti *CPU utilization, memory usage, disk I/O, network throughput,* dan *availability*. Dalam proses penyusunan laporan ini, tugas yang diberikan yaitu untuk melakukan pengambilan data dan tangkapan layar (screenshot) secara rutin dari sistem monitoring Amazon CloudWatch, termasuk grafik-grafik metrik yang relevan. Data mentah tersebut kemudian diolah menjadi tabel dan visualisasi yang lebih mudah dipahami. Selain itu, menyusun deskripsi penjelasan, interpretasi tren, serta mengidentifikasi potensi masalah mengacu pada template laporan yang telah ditentukan oleh perusahaan. Draft laporan yang telah disusun selanjutnya diajukan kepada supervisor untuk diperiksa sebelum disampaikan kepada klien. Penyusunan laporan bulanan ini memberikan pengalaman yang signifikan dalam memahami bagaimana monitoring cloud bekerja secara praktis, bagaimana membaca pola performa, serta bagaimana memberikan insight berbasis data untuk mendukung keputusan operasional maupun strategis.



Gambar 3.9 Cover Laporan Bulanan

2. *Preventive Maintenance Report*

Preventive maintenance report merupakan dokumen teknis yang digunakan untuk mencatat seluruh aktivitas pemeliharaan preventif pada lingkungan infrastruktur klien. Tujuan utama laporan ini adalah memastikan bahwa setiap komponen sistem tetap berada dalam kondisi optimal serta mengurangi risiko terjadinya gangguan atau insiden di masa mendatang. Berbeda dengan tindakan reaktif, preventive maintenance bersifat proaktif, sehingga seluruh aktivitas yang dicatat merupakan bentuk pencegahan sebelum masalah muncul. Dalam proses penyusunan *preventive maintenance report*, tugas yang diberikan mencakup pelaksanaan prosedur pemeliharaan, pemeriksaan status layanan, serta verifikasi beberapa parameter teknis seperti versi perangkat lunak, status patching, kapasitas penyimpanan, keamanan akses, dan konsistensi konfigurasi.

Setiap hasil pemeriksaan kemudian didokumentasikan ke dalam laporan tersebut, termasuk temuan yang muncul selama proses verifikasi. Apabila terdapat potensi masalah, temuan tersebut dicatat sebagai *finding* dan dipantau hingga status tindak lanjutnya dinyatakan selesai (*closed*). Pelaksanaan aktivitas preventive maintenance ini berperan penting dalam memastikan bahwa infrastruktur klien tetap memenuhi standar operasional yang ditetapkan, meminimalkan risiko *downtime*, serta mendukung operasional bisnis yang lebih stabil. Melalui kegiatan ini, diperoleh pengalaman langsung dalam melakukan pengecekan sistematis, mendokumentasikan hasil pemeliharaan, serta memahami praktik terbaik dalam menjaga keandalan layanan berbasis cloud.

3.3.1.3 Implementasi Backup Otomatis

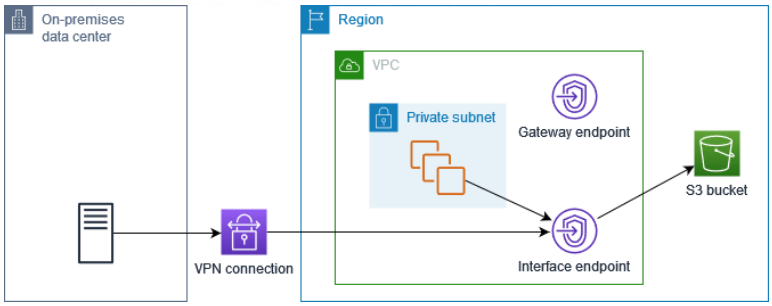
Pada periode magang yang penulis jalani, salah satu tugas yang diberikan adalah merancang dan mengimplementasikan sistem *automatic backup* dari lingkungan *on-premise* ke Amazon S3. Proses backup yang sebelumnya digunakan masih bersifat manual sehingga berpotensi menimbulkan keterlambatan, *human error*, serta tidak memiliki mekanisme keamanan yang memadai ketika data ditransfer melalui jaringan internet publik. Untuk memenuhi kebutuhan tersebut, perusahaan menginstruksikan agar penulis membangun solusi backup yang otomatis, terjadwal, dan aman dengan memanfaatkan Amazon S3, AWS CLI, serta koneksi privat melalui VPN *Site-to-Site*. Dengan pendekatan ini, seluruh proses pemindahan data dilakukan melalui jalur privat tanpa menggunakan IP publik, sehingga risiko kebocoran data dapat ditekan secara signifikan. Dalam pelaksanaan proyek, penulis melakukan berbagai aktivitas mulai dari menyiapkan lingkungan AWS, mengkonfigurasi VPN, memasang perangkat pendukung pada server lokal, membuat script *automated backup*, menguji konektivitas, hingga melakukan validasi

hasil unggahan ke Amazon S3. Seluruh langkah kerja terdokumentasi dengan baik dan disertai bukti gambar *(placeholder)* yang dapat diganti sesuai kebutuhan laporan. Proyek ini menjadi salah satu bagian penting dari kegiatan magang karena memberikan pengalaman langsung dalam menerapkan konsep *cloud computing*, keamanan jaringan, serta otomasi sistem pada lingkungan kerja nyata.

1. **Arsitektur**

   Arsitektur sistem backup yang digunakan dalam proyek ini dirancang untuk memastikan proses pemindahan data dari lingkungan lokal menuju layanan Amazon Web Services (AWS) dapat dilakukan secara aman, efisien, dan terstruktur. Arsitektur ini terdiri dari beberapa komponen utama, yaitu server lokal berbasis Virtual Machine (VM), perangkat jaringan seperti router atau firewall yang mendukung protokol IPsec VPN, AWS Customer Gateway, AWS Virtual Private Gateway, serta Amazon S3 Bucket sebagai tujuan akhir penyimpanan data. Dalam arsitektur ini, transfer data dilakukan melalui *tunnel* VPN terenkripsi berbasis IPsec yang menghubungkan jaringan lokal dengan infrastruktur AWS. Penggunaan IPsec VPN memungkinkan pembentukan kanal komunikasi privat yang terisolasi dari jaringan publik, sehingga seluruh proses transmisi data memiliki tingkat keamanan yang tinggi. Selain itu, mekanisme autentikasi antar perangkat pada Customer Gateway dan Virtual Private Gateway memastikan hanya sumber yang terotorisasi yang dapat mengakses jaringan AWS. Pendekatan ini sangat penting untuk menjaga integritas dan kerahasiaan data, terutama ketika menangani file backup yang berisi informasi penting atau sensitif. Arsitektur juga dirancang untuk mengatur alur trafik secara sistematis. Ketika server lokal mengeksekusi proses backup, data tidak dikirimkan langsung

ke internet, melainkan diarahkan terlebih dahulu ke perangkat jaringan lokal, kemudian masuk ke dalam *tunnel* VPN menuju Virtual Private Gateway di AWS. Setelah tiba di VPC AWS, trafik diteruskan ke S3 Gateway Endpoint agar server lokal dapat mengakses Amazon S3 tanpa perlu koneksi publik. Hal ini memastikan proses pengiriman data berlangsung sepenuhnya melalui jalur privat. Selain memprioritaskan aspek keamanan, arsitektur ini juga memperhatikan kebutuhan skalabilitas dan *fault tolerance*. Struktur yang digunakan memungkinkan perusahaan untuk menambahkan server tambahan tanpa perubahan besar pada konfigurasi inti. Apabila terjadi gangguan pada sisi jaringan tertentu, sistem tetap dapat dialihkan melalui jalur redundan atau dapat dipulihkan dengan cepat. Dengan demikian, arsitektur ini layak diterapkan pada lingkungan produksi yang membutuhkan ketersediaan sistem backup secara berkelanjutan. Secara keseluruhan, arsitektur ini memberikan landasan teknis yang stabil bagi implementasi sistem backup otomatis. Dengan memanfaatkan integrasi antara jaringan privat berbasis VPN dan layanan penyimpanan AWS, sistem ini mampu memberikan solusi yang aman, terukur, dan sesuai kebutuhan operasional perusahaan selama kegiatan magang berlangsung.



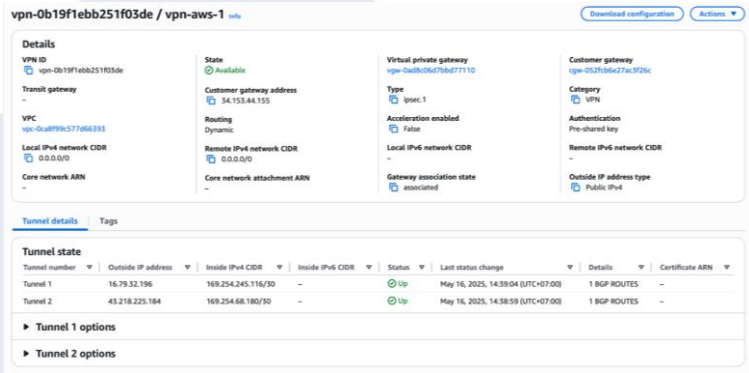
Gambar 3.10 Arsitektur Sistem

2. Konfigurasi VPN Site-to-Site

Tahap awal dalam implementasi sistem backup otomatis adalah membangun konektivitas privat yang aman antara infrastruktur lokal dan layanan AWS melalui mekanisme VPN Site-to-Site (S2S). Konektivitas ini menjadi komponen fundamental karena seluruh proses transfer file backup dilakukan melalui tunnel terenkripsi yang tidak melewati internet publik. Dengan demikian, integritas, kerahasiaan, serta keamanan data dapat dipertahankan sesuai dengan standar yang umum digunakan dalam lingkungan produksi. Pada tahap ini, penulis merancang dan menerapkan konfigurasi S2S VPN secara menyeluruh, mulai dari definisi endpoint, penentuan skema routing, hingga konfigurasi parameter keamanan pada protokol IPsec. Konfigurasi dimulai dengan pembuatan Customer Gateway (CGW) pada sisi AWS. CGW merepresentasikan perangkat jaringan lokal, baik berupa router, firewall, maupun server yang memiliki kemampuan IPsec, serta memiliki alamat IP publik yang berfungsi sebagai endpoint untuk terhubung ke AWS. Pada tahap ini, penulis menentukan jenis routing yang digunakan yaitu routing statis dengan mendefinisikan jaringan internal lokal yang harus dapat diakses oleh AWS. Setelah CGW dibuat, penulis melanjutkan dengan membangun Virtual Private Gateway (VGW), yaitu perangkat virtual dalam AWS yang berfungsi sebagai titik terminasi VPN pada sisi cloud. VGW kemudian di-attach ke Virtual Private Cloud (VPC) yang menjadi lokasi bucket S3 Endpoint dan komponen AWS lainnya yang terlibat dalam sistem backup. Tahap berikutnya adalah pembuatan VPN Connection antara CGW dan VGW. AWS secara otomatis menghasilkan dua tunnel IPsec sebagai redundant channel untuk memastikan konektivitas tetap tersedia meskipun

salah satu tunnel mengalami gangguan atau proses rekeying. Setiap tunnel memiliki parameter konfigurasi yang berbeda, termasuk alamat endpoint AWS, pre-shared key, algoritma kriptografi, masa aktif Security Association, serta pengaturan *Internet Key Exchange*. File konfigurasi VPN yang disediakan AWS kemudian dapat diunduh dan diterapkan pada perangkat lokal yang mendukung protokol IPsec, tanpa bergantung pada implementasi perangkat lunak tertentu, sehingga memberikan fleksibilitas dalam pemilihan platform. Pada tahap implementasi, penulis menyesuaikan parameter IKE Phase 1, seperti algoritma enkripsi AES-256, algoritma integritas SHA-256, serta Diffie-Hellman Group 14, untuk pertukaran kunci yang aman. Sementara itu, pada IKE Phase 2, digunakan algoritma ESP AES-256-SHA256 dengan pengaturan lifetime yang konsisten antara sisi AWS dan lokal, untuk mencegah pemutusan tunnel akibat ketidaksinkronan masa aktif SA. Konfigurasi ini disusun sedemikian rupa agar enkripsi tetap kuat sekaligus menjaga stabilitas koneksi tunnel dalam jangka panjang. Pengaturan tambahan, seperti Dead Peer Detection dan auto-rekey, diterapkan sesuai kemampuan perangkat jaringan yang digunakan. Setelah seluruh konfigurasi diterapkan, penulis melakukan aktivasi tunnel dan verifikasi koneksi. Proses ini mencakup pengecekan fase inisiasi IKE, pembentukan SA IPsec, serta uji ketersediaan kedua tunnel sebagai jalur aktif maupun cadangan. Selain itu, penulis menambahkan *static routing* di sisi lokal agar jaringan internal dapat mengakses AWS melalui interface IPsec. Pengujian dilakukan menggunakan *ping*, *traceroute*, serta uji akses ke S3 melalui S3 VPC Endpoint untuk memastikan seluruh trafik berjalan melalui jalur privat dan terenkripsi. Dengan konfigurasi

tersebut, koneksi privat antara server lokal dan AWS dapat berfungsi optimal. Implementasi VPN *Site-to-Site* ini memastikan keamanan proses backup, meningkatkan keandalan dan stabilitas jaringan, serta menjadi fondasi utama agar sistem backup otomatis dapat berjalan tanpa hambatan dan tanpa ketergantungan pada internet publik.



Gambar 3.11 Konfigurasi VPN Site-to-Site

3. Persiapan Server Lokal

Sebelum proses backup otomatis dapat dilakukan, server lokal atau *virtual machine* (VM) yang menjadi sumber data harus dipersiapkan secara menyeluruh agar proses transfer ke Amazon S3 dapat berjalan dengan lancar dan aman. Persiapan ini mencakup beberapa aspek teknis yang krusial, mulai dari instalasi perangkat lunak pendukung hingga pengaturan kapasitas penyimpanan. Salah satu komponen utama yang perlu dipasang adalah AWS Command Line Interface (AWS CLI), yang akan digunakan untuk melakukan operasi transfer file secara otomatis dari server lokal menuju bucket S3. Instalasi AWS CLI dilakukan sesuai dengan versi sistem operasi yang digunakan, baik Linux maupun Windows, dan diikuti dengan konfigurasi kredensial yang valid, termasuk Access Key ID, Secret Access Key, serta region default, agar server dapat berkomunikasi dengan layanan AWS dengan aman. Selain instalasi perangkat

lunak, penulis juga memastikan bahwa kapasitas penyimpanan pada server lokal atau VM cukup untuk menampung seluruh snapshot, dump file database, atau file backup lain sebelum dikirim ke Amazon S3. Hal ini penting untuk mencegah kegagalan proses backup akibat keterbatasan ruang penyimpanan, terutama ketika ukuran file backup relatif besar. Sebelum proses pengiriman dilakukan, data backup biasanya dikemas ke dalam format arsip seperti .tar.gz atau .zip untuk mengurangi ukuran file, mempermudah manajemen, serta mempercepat proses transfer melalui koneksi VPN yang telah dibuat sebelumnya. Pengemasan data ini juga membantu menjaga integritas file selama proses pengiriman, karena seluruh file dapat divalidasi sebagai satu kesatuan sebelum diunggah. Tahap berikutnya adalah validasi akun AWS dan uji konektivitas antara server lokal dengan bucket S3. Penulis melakukan pengujian sederhana menggunakan perintah aws s3 ls untuk memastikan server dapat mengakses bucket yang telah ditentukan, serta melakukan uji unggah file kecil untuk memastikan kredensial dan pengaturan jaringan berjalan sesuai harapan. Langkah-langkah ini menjadi bagian penting dalam memastikan bahwa proses backup otomatis akan berjalan lancar ketika dijadwalkan menggunakan cron job atau mekanisme otomatisasi lainnya. Dengan persiapan yang matang, server lokal atau VM tidak hanya siap untuk melakukan backup rutin, tetapi juga mampu menjamin keamanan dan integritas data selama proses transfer ke cloud.

```
# DONWLOAD INSTALLER
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
# UNZIP
unzip awscliv2.zip
# INSTALL
sudo ./aws/install
# CHECK VERSION
aws --version
```

Gambar 3.12 Instalasi AWS CLI

4. Implementasi Script Backup

Proses backup otomatis pada proyek ini dikembangkan menggunakan Bash scripting di lingkungan server Linux. Bash dipilih karena sifatnya yang ringan, fleksibel, dan mampu menjalankan perintah sistem secara langsung. Script ini memanfaatkan AWS CLI, terutama perintah aws s3 cp dan aws s3 sync, untuk mengunggah file backup dari server lokal ke bucket Amazon S3. Pendekatan ini memungkinkan integrasi mudah dengan cron scheduler, sehingga backup dapat dijalankan secara otomatis tanpa intervensi manual, meningkatkan efisiensi operasional dan mengurangi risiko kesalahan manusia. Script backup dilengkapi dengan mekanisme logging yang mendetail, mencatat setiap file yang berhasil atau gagal diunggah, beserta tanggal dan waktu eksekusi. Selain itu, script melakukan pengecekan untuk menghindari duplikasi file, sehingga efisiensi penyimpanan tetap terjaga. Script ini juga mendukung berbagai jenis database perusahaan, yaitu MySQL, Microsoft SQL Server (MSSQL), dan Oracle Database, dengan masing-masing backup diarahkan ke folder terstruktur di S3 sesuai jenis database dan kategori backup (harian, mingguan, bulanan). Dengan desain modular dan penggunaan fungsi-fungsi Bash yang sistematis, script ini mudah dimodifikasi untuk perubahan struktur direktori, penambahan database, atau kebijakan backup baru. Implementasi ini menjamin proses backup berjalan otomatis, aman, efisien, dan dapat diaudit, sehingga mendukung tujuan

perusahaan dalam menjaga integritas, keamanan, dan keberlanjutan data kritis.

a. Konfigurasi Awal dan Variabel Global

Bagian ini mendefinisikan lokasi direktori backup untuk masing-masing database, folder atau file yang dikecualikan dari backup, bucket S3 tujuan, serta file log untuk mencatat aktivitas backup. Variabel UPLOAD_COUNT digunakan untuk menghitung jumlah file yang berhasil diunggah. Variabel DB_TARGET memungkinkan script dijalankan untuk satu database tertentu atau semua sekaligus.

```
# ========== CONFIGURATIONS ==========
ORACLE_SRC="/home/server/database/oracle"
MYSQL_SRC="/home/server/database/mysql"
MSSQL_SRC="/home/server/database/mssql"

ORACLE_SRC_EXC="skipthisfolder"
MYSQL_SRC_EXC="excluded-db"
MSSQL_SRC_EXC="app1/test"

S3_BUCKET="s3://ict-backup"
S3_PATH_PREFIX="backup"

LOGFILE="/var/log/upload_all_$(date +%Y%m%d_%H%M%S).log"
exec > >(tee -a "$LOGFILE") 2>&1

UPLOAD_COUNT=0

# ========== RUNTIME TARGET SELECTION ==========
DB_TARGET="$1"
[[ -z "$DB_TARGET" ]] && DB_TARGET="all"
DB_TARGET=$(echo "$DB_TARGET" | tr '[:upper:]' '[:lower:]')

echo "[INFO] Starting full database backup copy script at $(date)"
```

Gambar 3.13 Konfigurasi Awal dan Variabel Global

b. Fungsi dan Logika Utama Script

Fungsi-fungsi berikut merupakan komponen modular yang membentuk alur utama dari *script backup* otomatis. Setiap fungsi memiliki peran spesifik yang mendukung proses klasifikasi backup, validasi berkas, hingga pengunggahan data ke Amazon S3. Secara keseluruhan, fungsi-fungsi ini bekerja secara terpadu untuk memastikan proses backup berjalan terstruktur, aman, dan hanya memproses berkas yang benar-benar relevan. Gambar-gambar berikut menampilkan struktur, alur

kerja, serta implementasi masing-masing fungsi dalam *script*.

```
# ========== FUNCTIONS ==========
get_week_of_month() {
    day=$(date -d "$1" +%d)
    echo $(((day - 1) / 7 + 1))
}

get_weekday() {
    date -d "$1" +%u
}
```

Gambar 3.14 Script Kategori Backup

Selain itu, script ini dilengkapi mekanisme *retry* yang diterapkan pada proses pengunggahan file maupun folder ke Amazon S3. Jika terjadi kegagalan saat proses unggah, baik karena kendala jaringan maupun gangguan sementara pada layanan, *script* akan mencoba kembali hingga tiga kali sebelum menetapkan unggahan tersebut sebagai gagal. Pendekatan ini memastikan proses backup tetap andal dan meminimalkan risiko kehilangan data akibat gangguan yang bersifat sementara.

```
upload_file_s3() {
    local file="$1"
    local app="$2"
    local cat="$3"
    local db="$4"
    local s3_path="${S3_BUCKET}/${S3_PATH_PREFIX}/${db}/${app}/${cat}/"

    echo "[INFO] Uploading file $file to $s3_path"
    local max_retry=3
    local attempt=0
    until [ $attempt -ge $max_retry ]; do
        aws s3 cp "$file" "$s3_path" --only-show-errors
        if [ $? -eq 0 ]; then
            echo "[SUCCESS] Uploaded: $file"
            UPLOAD_COUNT=$((UPLOAD_COUNT+1))
            return 0
        fi
        echo "[RETRY] Failed to upload $file (attempt $((attempt+1)))"
        attempt=$((attempt+1))
        sleep 5
    done
    echo "[ERROR] Failed to upload $file after $max_retry attempts"
    return 1
}

upload_folder_s3() {
    local folder="$1"
    local app="$2"
    local cat="$3"
    local db="$4"
    local s3_path="${S3_BUCKET}/${S3_PATH_PREFIX}/${db}/${app}/${cat}/"

    echo "[INFO] Uploading folder $folder to $s3_path"
    local max_retry=3
    local attempt=0
    until [ $attempt -ge $max_retry ]; do
        aws s3 cp "$folder" "$s3_path" --recursive --only-show-errors
        if [ $? -eq 0 ]; then
            echo "[SUCCESS] Uploaded folder: $folder"
            UPLOAD_COUNT=$((UPLOAD_COUNT+1))
            return 0
        fi
        echo "[RETRY] Failed to upload folder $folder (attempt $((attempt+1)))"
        attempt=$((attempt+1))
        sleep 5
    done
    echo "[ERROR] Failed to upload folder $folder after $max_retry attempts"
    return 1
}
```

Gambar 3.15 Script Melakukan Unggahan

```
is_excluded() {
    local path="$1"
    EXCLUDES=(
        "$ORACLE_SRC/$ORACLE_SRC_EXC"
        "$MYSQL_SRC/$MYSQL_SRC_EXC"
        "$MSSQL_SRC/$MSSQL_SRC_EXC"
    )
    for exclude in "${EXCLUDES[@]}"; do
        if [[ "$path" == "$exclude"* ]]; then
            return 0
        fi
    done
    return 1
}
```

Gambar 3.16 Script Pemeriksaan Pengecualian

```
# ========== DATE LOGIC ==========
TODAY=$(date +%Y-%m-%d)
YESTERDAY=$(date -d "yesterday" +%Y-%m-%d)
echo "[INFO] Today: $TODAY | Processing backup from: $YESTERDAY"
```

Gambar 3.17 Script Memilih File Backup

c. Proses *Upload Backup per Database*

Proses unggahan (upload) backup pada sistem ini dilakukan secara terstruktur untuk setiap jenis database agar manajemen file backup menjadi lebih rapi dan mudah diawasi. Dengan membedakan mekanisme unggahan berdasarkan jenis database, risiko kesalahan atau kehilangan data dapat diminimalkan, sekaligus mempermudah proses pemulihan ketika diperlukan. Untuk Oracle, script yang digunakan secara otomatis menyeleksi folder hasil backup RMAN yang dibuat sesuai jadwal. Script ini memfilter file berdasarkan tanggal pembuatan backup sehingga hanya file terbaru yang akan diunggah, sekaligus menentukan kategori backup, apakah termasuk weekly backup atau monthly backup, sehingga pengelolaan arsip menjadi lebih sistematis. Selain itu, script juga memeriksa struktur folder dan memastikan tidak ada file duplikat yang diunggah, sehingga efisiensi penyimpanan tetap terjaga.

```
# ========== ORACLE ==========
if [[ "$DB_TARGET" == "all" || "$DB_TARGET" == "oracle" ]]; then
    if [[ -d "$ORACLE_SRC" ]]; then
        echo "[INFO] Oracle: Uploading full backup folders"
        find "$ORACLE_SRC" -mindepth 2 -maxdepth 2 -type d -name "rman_*" ! -name "*lv1*" | while read -r dir; do
            is_excluded "$dir" && { echo "[SKIP] Oracle excluded: $dir"; continue; }

            folder_name=$(basename "$dir")
            date_str=$(echo "$folder_name" | grep -oE '[0-9]{2}-[0-9]{2}-[0-9]{4}')
            [[ -z "$date_str" ]] && echo "[SKIP] $folder_name no date" && continue
            file_date=$(date -d "$(echo "$date_str" | awk -F- '{print $3"-"$2"-"$1}')" +%Y-%m-%d 2>/dev/null)
            [[ $? -ne 0 ]] && echo "[SKIP] Invalid date in $folder_name" && continue
            [[ "$file_date" == "$YESTERDAY" ]] || continue

            weekday=$(get_weekday "$file_date")
            week_of_month=$(get_week_of_month "$file_date")
            category="weekly"
            [[ "$weekday" == "1" && "$week_of_month" -eq 4 ]] && category="monthly"
            app_name=$(realpath --relative-to="$ORACLE_SRC" "$(dirname "$dir")")
            upload_folder_s3 "$dir" "$app_name" "$category" "oracle"
        done
    else
        echo "[SKIP] Oracle directory missing"
    fi
fi
```

Gambar 3.18 Oracle Database Script

Pada MySQL, script bekerja dengan cara menyeleksi file dump berformat .sql berdasarkan tanggal pembuatan. Selain memastikan file yang diunggah adalah backup terbaru, script juga membandingkan daftar file dengan daftar pengecualian (exclusion list) sehingga file yang tidak perlu diunggah tidak ikut terproses. Hal ini membantu mengurangi penggunaan ruang penyimpanan cloud yang tidak perlu dan menjaga proses upload tetap efisien.

```
# ========== MYSQL ==========
if [[ "$DB_TARGET" == "all" || "$DB_TARGET" == "mysql" ]]; then
    if [[ -d "$MYSQL_SRC" ]]; then
        echo "[INFO] MySQL: Uploading full backup files"
        find "$MYSQL_SRC" -type f | while read -r file; do
            is_excluded "$file" && { echo "[SKIP] MySQL excluded: $file"; continue; }
            filename=$(basename "$file")
            date_str=$(echo "$filename" | grep -oE '[0-9]{8}')
            [[ -z "$date_str" ]] && echo "[SKIP] $filename no date" && continue
            file_date=$(date -d "$date_str" +%Y-%m-%d 2>/dev/null)
            [[ $? -ne 0 ]] && echo "[SKIP] Invalid date in $filename" && continue
            [[ "$file_date" == "$YESTERDAY" ]] || continue
            weekday=$(get_weekday "$file_date")
            week_of_month=$(get_week_of_month "$file_date")
            category="weekly"
            [[ "$weekday" == "1" && "$week_of_month" -eq 4 ]] && category="monthly"
            app_name=$(realpath --relative-to="$MYSQL_SRC" "$(dirname "$file")")
            upload_file_s3 "$file" "$app_name" "$category" "mysql"
        done
    else
        echo "[SKIP] MySQL directory missing"
    fi
fi
```

Gambar 3.19 MySQL Script

Sementara itu, untuk MSSQL, script mengunggah file berformat .bak dengan langkah awal memeriksa tanggal pembuatan file serta kategori backup yang bersangkutan. Dengan memisahkan kategori backup,

baik itu harian, mingguan, atau bulanan, sistem dapat menjaga keteraturan penyimpanan data dan mempermudah proses monitoring. Script juga memeriksa integritas file sebelum proses unggah dilakukan, sehingga setiap file yang dikirim ke cloud telah tervalidasi dan siap digunakan untuk proses restore bila dibutuhkan.

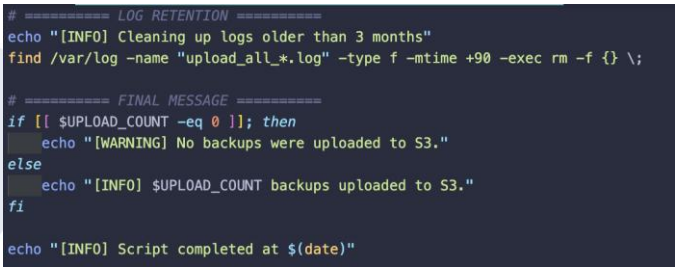
Gambar 3.20 MSSQL Script

Dengan mekanisme unggahan yang berbeda-beda sesuai karakteristik masing-masing database, seluruh proses ini tidak hanya meningkatkan efisiensi dan keandalan penyimpanan data, tetapi juga meminimalkan risiko kesalahan manusia, duplikasi file, dan kehilangan data. Pendekatan ini memastikan setiap backup terkelola dengan baik, mudah dilacak, serta dapat diandalkan ketika proses pemulihan data diperlukan.

d. Retensi Log dan Akhir Eksekusi

Bagian ini berfungsi untuk melakukan pembersihan file log yang telah berusia lebih dari tiga bulan, sehingga ruang penyimpanan pada server tetap terjaga dan tidak dibebani oleh akumulasi log yang

tidak lagi diperlukan. Praktik pembersihan ini penting untuk mencegah penurunan kinerja sistem akibat penggunaan storage yang berlebihan, sekaligus memastikan bahwa hanya log yang relevan dan masih diperlukan yang tetap disimpan. Setelah seluruh proses pembersihan dan unggahan backup selesai dilakukan, script akan menampilkan ringkasan berupa jumlah file yang berhasil diunggah. Informasi ini memberikan gambaran yang jelas kepada administrator mengenai status keseluruhan proses backup, membantu dalam evaluasi harian maupun pelacakan apabila terjadi ketidaksesuaian atau indikasi kegagalan pada salah satu tahapan. Dengan demikian, bagian ini tidak hanya menjaga efisiensi penyimpanan server, tetapi juga mendukung transparansi dan akurasi dalam pemantauan proses backup secara menyeluruh.



Gambar 3.21 Retensi Log

5. Konfigurasi Cronjob

Untuk memastikan proses backup berjalan secara otomatis dan terjadwal, cron digunakan sebagai scheduler pada sistem Linux. Cron memungkinkan penjadwalan eksekusi script backup secara periodik, baik harian, mingguan, maupun sesuai kebutuhan organisasi. Dengan memanfaatkan cron, seluruh proses backup dapat dijalankan tanpa intervensi manual, sehingga meminimalkan risiko kelalaian manusia

dan memastikan kontinuitas keamanan data. Pada implementasi proyek ini, penulis melakukan konfigurasi cron melalui file /etc/crontab atau menggunakan user-specific crontab, dengan memastikan script backup memiliki permission eksekusi yang benar. Secara khusus, script backup dijadwalkan untuk dijalankan setiap hari Selasa pukul 1 pagi untuk melakukan unggahan file backup dari MySQL, MSSQL, dan Oracle Database ke Amazon S3. Penjadwalan ini dipilih agar proses backup berjalan di luar jam operasional utama, sehingga tidak mengganggu performa server dan aktivitas pengguna. Selain itu, cron dilengkapi dengan mekanisme logging agar setiap eksekusi dapat dipantau, termasuk informasi keberhasilan atau kegagalan proses. Penulis juga membahas beberapa tips dan strategi troubleshooting cron yang sering muncul pada lingkungan produksi, seperti masalah PATH environment, hak akses file, serta konflik antara user cron dan root cron. Dengan penanganan yang tepat, cron menjadi komponen yang andal untuk mengotomatiskan proses backup, menjaga integritas data, dan mendukung operasional perusahaan dalam jangka panjang. Baris perintah berikut digunakan untuk mengkonfigurasi penjadwalan otomatis pada sistem berbasis Linux menggunakan *cron daemon*.

Gambar 3.22 Konfigurasi Cronjob

6. Pengujian Sistem Backup

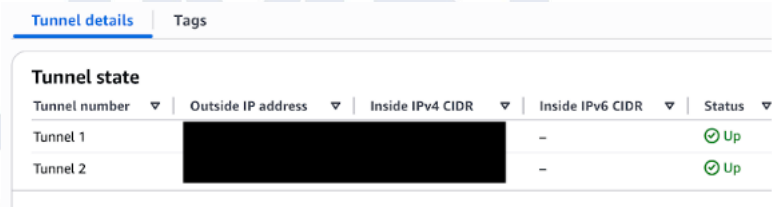
Pengujian sistem backup otomatis dilakukan untuk memastikan bahwa seluruh proses, mulai dari pengambilan data pada server lokal hingga penyimpanan di Amazon S3, berjalan sesuai dengan perancangan yang telah ditetapkan. Pengujian ini bertujuan untuk memverifikasi keandalan

sistem, integritas data, serta kesiapan mekanisme backup dalam mendukung kebutuhan operasional perusahaan.

a. Pengujian Konektivitas VPN Site-to-Site

Pengujian pertama dilakukan pada aspek konektivitas jaringan dengan memverifikasi kestabilan VPN Site-to-Site antara infrastruktur lokal dan Amazon Web Services (AWS). Pengujian ini mencakup pengecekan status tunnel VPN melalui AWS Management Console, memastikan kedua tunnel berada dalam kondisi aktif (UP), serta memverifikasi keberhasilan proses Internet Key Exchange (IKE) dan pembentukan Security Association (SA).

Konektivitas VPN yang stabil merupakan prasyarat utama agar proses transfer data backup dapat dilakukan secara aman dan berkelanjutan. Hasil pengujian menunjukkan bahwa koneksi VPN Site-to-Site berada dalam kondisi stabil selama proses backup berlangsung, tanpa terjadinya pemutusan tunnel maupun degradasi koneksi yang signifikan.



Gambar 3.23 Status VPN

b. Pengujian Proses Backup ke Amazon S3

Pengujian selanjutnya berfokus pada proses pengiriman file backup dari server lokal ke Amazon S3 menggunakan AWS Command Line Interface (CLI). File yang diuji meliputi backup database MySQL, Microsoft SQL Server (MSSQL), dan Oracle Database yang telah dikompresi dalam format arsip. Selama proses pengujian,

dicatat ukuran file backup, waktu mulai dan selesai proses upload, serta jumlah percobaan ulang (retry) apabila terjadi kegagalan transfer.

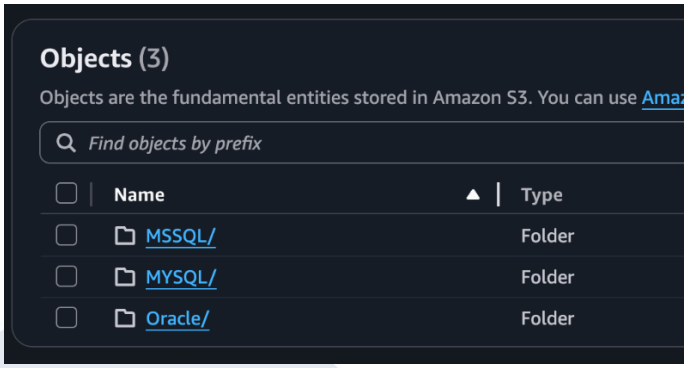
Hasil pengujian menunjukkan bahwa proses backup dapat berjalan dengan baik untuk file berukuran ratusan megabyte hingga beberapa gigabyte. Waktu transfer bervariasi tergantung pada ukuran file dan kondisi bandwidth jaringan, namun secara umum proses upload dapat diselesaikan tanpa gangguan yang berarti.

Gambar 3.24 Proses Upload

c. Validasi File Backup di Amazon S3

Setelah proses upload selesai, dilakukan validasi untuk memastikan bahwa file backup telah tersimpan dengan lengkap dan sesuai dengan struktur direktori yang telah dirancang. Validasi dilakukan dengan memeriksa isi bucket Amazon S3 melalui AWS Management Console, serta mencocokkan jumlah dan ukuran file antara server lokal dan bucket S3.

Hasil validasi menunjukkan bahwa seluruh file backup database MySQL, MSSQL, dan Oracle berhasil tersimpan di Amazon S3 tanpa adanya perbedaan ukuran file maupun kehilangan data. Hal ini menunjukkan bahwa proses transfer data berlangsung secara utuh.

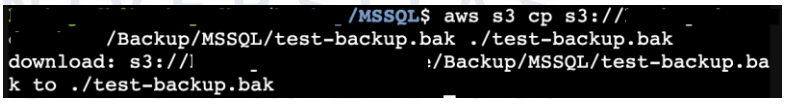


Gambar 3.25 Struktur Folder

d. Pengujian Proses Restore dari Amazon S3 ke Server Lokal

Selain pengujian proses backup, dilakukan pula pengujian proses *restore* data dari Amazon S3 ke server lokal. Pengujian ini bertujuan untuk memastikan bahwa file backup yang tersimpan di S3 dapat digunakan kembali apabila terjadi kegagalan sistem atau kehilangan data pada server utama.

Proses restore dilakukan dengan mengunduh file backup dari Amazon S3 menggunakan AWS CLI, kemudian mengekstrak file arsip dan melakukan *restore* database. Hasil pengujian menunjukkan bahwa proses download file dari S3 berjalan dengan lancar dan seluruh database dapat direstore dengan baik tanpa kehilangan data maupun kesalahan struktur data.



Gambar 3.26 Proses Restore dari Cloud

e. Evaluasi Hasil Pengujian

Untuk mengevaluasi keberhasilan sistem backup secara objektif, ditetapkan beberapa kriteria keberhasilan yang dapat diukur. Kriteria ini digunakan sebagai acuan dalam menilai apakah sistem backup telah memenuhi

kebutuhan operasional perusahaan, baik dari sisi waktu pemulihan, konsistensi data, maupun tingkat keberhasilan proses.

Table 3.2 Kriteria Pengujian

Parameter	Target
Recovery Point Objective (RPO)	≤ 1 hari
Recovery Time Objective (RTO)	≤ 60 menit
Backup Window	Di luar jam operasional
Retensi Backup	7 hari
Tingkat Keberhasilan Backup	100%

Berdasarkan kriteria tersebut, dilakukan pencatatan hasil pengujian secara kuantitatif yang meliputi ukuran file backup, waktu proses backup dan restore, serta status keberhasilan transfer data.

Table 3.3 Hasil Pengujian

Jenis Database	Ukuran	Waktu Backup	Waktu Restore	Retry	Status
MySQL	350 MB	4 menit	3 menit	0	Berhasil
MSSQL	1,2 GB	9 menit	7 menit	1	Berhasil
Oracle	2,8 GB	15 menit	12 menit	0	Berhasil

Seluruh aktivitas backup dan restore juga dicatat dalam file log pada server lokal yang berisi informasi waktu eksekusi, status proses, serta pesan kesalahan apabila terjadi kegagalan. Berdasarkan hasil pengujian dan data kuantitatif yang diperoleh, sistem backup otomatis terbukti mampu memenuhi kriteria keberhasilan yang telah ditetapkan. Proses backup dan restore berjalan sesuai target RPO dan RTO, tanpa kehilangan data maupun kegagalan transfer, sehingga sistem ini dinilai layak untuk diterapkan pada lingkungan operasional perusahaan.

7. **Analisis Keamanan**

Keamanan data merupakan aspek yang sangat kritis dalam implementasi sistem backup, mengingat data yang diunggah ke cloud seringkali bersifat sensitif dan memiliki nilai strategis bagi organisasi. Dalam proyek ini, beberapa mekanisme keamanan diterapkan untuk memastikan bahwa seluruh proses transfer, penyimpanan, dan pemulihan data dapat berjalan dengan aman, integritas terjaga, dan risiko kebocoran diminimalkan. Pertama, VPN Site-to-Site (S2S) digunakan untuk membangun jalur komunikasi privat antara server lokal dan AWS. Dengan adanya tunnel IPsec yang terenkripsi, data backup tidak melewati internet publik secara langsung, sehingga mengurangi risiko intersepsi atau serangan *Man-in-the-Middle (MITM)*. Jalur ini juga mendukung *redundancy* dengan dua tunnel terpisah, sehingga apabila satu tunnel mengalami gangguan, tunnel kedua dapat tetap menjaga kontinuitas transfer data. Pengaturan protokol IPsec, algoritma enkripsi AES-256, dan integritas SHA-256 memastikan bahwa data tetap rahasia dan autentik selama proses pengiriman. Kedua, Amazon S3 sebagai tujuan penyimpanan menyediakan berbagai mekanisme perlindungan data. Salah satunya adalah enkripsi sisi server, baik melalui SSE-S3 maupun SSE-KMS, yang memastikan bahwa data yang tersimpan dalam bucket terenkripsi secara otomatis. Selain itu, pengaturan bucket policy dapat diterapkan secara ketat, misalnya hanya mengizinkan akses dari IP tertentu, VPC tertentu, atau akun IAM tertentu. Hal ini membatasi ruang lingkup akses dan meminimalkan kemungkinan penyalahgunaan kredensial atau akses tidak sah.Selanjutnya, aspek mitigasi risiko terhadap data corruption juga diperhatikan. Proses backup

mencakup validasi checksum sebelum dan sesudah transfer, sehingga integritas file dapat diverifikasi. Setiap file yang diunggah dicatat dalam log dan dibandingkan dengan file asli, sehingga potensi korupsi dapat segera terdeteksi dan ditindaklanjuti. Selain itu, risiko kebocoran credential juga diminimalkan dengan penerapan prinsip keamanan minimal (*least privilege*). IAM Role dan IAM Policy di AWS dikonfigurasi agar hanya memberikan izin yang diperlukan untuk melakukan upload dan akses backup, tanpa memberikan hak administratif berlebih yang dapat disalahgunakan. Kredensial pada server lokal disimpan dengan aman menggunakan file konfigurasi yang hanya dapat diakses oleh pengguna dengan hak eksekusi tertentu, sehingga mengurangi kemungkinan kompromi. Dengan kombinasi mekanisme keamanan di sisi jaringan (VPN), enkripsi di sisi penyimpanan (S3), pengaturan hak akses dan bucket policy, serta validasi integritas data, sistem backup ini dirancang untuk memenuhi standar praktik keamanan industri. Pendekatan ini memastikan bahwa data sensitif, termasuk database MySQL, MSSQL, dan Oracle, tetap terlindungi dari berbagai ancaman mulai dari intersepsi, korupsi, hingga penyalahgunaan kredensial, sekaligus mendukung keberlanjutan dan keandalan operasional perusahaan.

8. Analisa Performa dan Optimasi

Performa proses backup merupakan faktor penting yang menentukan efisiensi operasional dan keberlanjutan sistem penyimpanan data. Dalam implementasi sistem backup otomatis ini, performa dipengaruhi oleh beberapa variabel utama, antara lain bandwidth jaringan, ukuran file backup, metode kompresi, serta kinerja storage lokal pada server atau

VM. Bandwidth jaringan membatasi kecepatan transfer file ke Amazon S3, sehingga pemilihan jalur koneksi yang stabil dan cepat menjadi aspek krusial. Sementara ukuran file backup yang besar dapat memperpanjang waktu transfer, sehingga strategi pengemasan dan kompresi data menjadi pertimbangan penting untuk mengurangi ukuran file tanpa mengorbankan integritas data. Analisis performa dilakukan dengan membandingkan berbagai metode kompresi, seperti gzip, bzip2, dan zip, serta menguji dampaknya terhadap waktu transfer, penggunaan CPU, dan ukuran akhir file. Selain itu, variasi ukuran file backup diuji untuk memahami bagaimana sistem merespons file yang kecil versus file berukuran besar, sehingga dapat ditentukan konfigurasi optimal untuk skala data yang berbeda. Selain metode kompresi, beberapa strategi optimasi juga diterapkan untuk meningkatkan efisiensi proses backup. Pertama, incremental backup memungkinkan hanya file atau perubahan data terbaru yang dikirim ke S3, sehingga mengurangi redudansi dan mempercepat proses transfer. Kedua, deduplication digunakan untuk menghindari pengunggahan file yang sama berkali-kali, baik di sisi lokal maupun cloud. Ketiga, parallel upload memanfaatkan kemampuan AWS CLI atau script Bash untuk melakukan unggahan beberapa file secara bersamaan, memaksimalkan throughput jaringan dan memperpendek waktu total backup. Dengan pendekatan analisis performa ini, penulis dapat mengidentifikasi bottleneck pada sistem, memilih strategi kompresi dan transfer yang optimal, serta memastikan bahwa backup dari berbagai database termasuk MySQL, MSSQL, dan Oracle dapat dilakukan secara efisien, aman, dan dapat diandalkan. Analisis ini juga menjadi dasar bagi rekomendasi

peningkatan performa sistem di masa depan, terutama pada skenario produksi dengan volume data yang lebih besar.

3.3.2 Kendala yang Ditemukan

Dalam proses pelaksanaan kerja magang, berbagai kendala dan tantangan tidak dapat dihindari, baik dari sisi teknis maupun non-teknis. Situasi ini menjadi bagian penting dari proses pembelajaran, karena menuntut penulis untuk beradaptasi dengan cepat terhadap dinamika dunia kerja yang sesungguhnya, terutama di lingkungan teknologi yang kompleks dan terus berkembang. Berbagai kendala tersebut juga membantu penulis mengasah kemampuan berpikir kritis, problem solving, serta komunikasi dalam tim. Selain itu, pengalaman menghadapi hambatan ini memberikan gambaran nyata mengenai bagaimana setiap proses kerja membutuhkan ketelitian, koordinasi, dan kesiapan dalam menghadapi perubahan yang tidak terduga. Adapun beberapa kendala utama yang dihadapi selama masa magang adalah sebagai berikut.

1. **Kendala Teknis Implementasi Cloud**

   Selama proses implementasi infrastruktur cloud, salah satu kendala utama yang saya hadapi adalah tantangan teknis yang kompleks dalam konfigurasi layanan, terutama yang melibatkan Site-to-Site VPN, pengaturan IPsec, routing statis, serta integrasi koneksi privat menuju layanan seperti Amazon S3. Kendala ini muncul karena setiap lingkungan memiliki konfigurasi unik, sehingga pengaturan yang terlihat sederhana dalam dokumentasi resmi sering kali tidak langsung berhasil ketika diterapkan pada sistem nyata. Misalnya, perbedaan parameter antara perangkat lokal dan konfigurasi yang disyaratkan AWS, seperti algoritma enkripsi, IKE version, SA lifetime, maupun pengaturan routing, sering menyebabkan tunnel IPsec gagal ter-establish, sehingga memerlukan proses troubleshooting berulang untuk menemukan kombinasi pengaturan yang tepat. Selain itu, pengaturan koneksi privat ke S3 melalui VPC Endpoint juga menimbulkan tantangan tersendiri. Pada percobaan

awal, koneksi tidak berjalan sebagaimana mestinya karena beberapa isu, seperti akses yang belum tersinkronisasi, route yang belum dikenali, atau policy yang belum dikonfigurasi dengan benar. Hal ini memaksa penulis untuk melakukan pengecekan dan penyesuaian konfigurasi berkali-kali, termasuk berkoordinasi dengan engineer senior untuk memastikan setiap parameter sesuai dengan kondisi infrastruktur aktual. Tantangan ini menunjukkan bahwa implementasi cloud tidak hanya sekadar mengikuti dokumentasi, tetapi juga membutuhkan pemahaman mendalam terhadap lingkungan operasional, kemampuan troubleshooting, serta ketelitian tinggi dalam menyesuaikan konfigurasi dengan realita lapangan.

2. **Kurangnya Pemahaman Awal tentang Cloud**

   Kendala berikutnya muncul dari sisi kurangnya familiaritas penulis dengan konsep cloud computing secara mendalam, khususnya layanan-layanan yang tersedia di AWS dan OCI. Meskipun memiliki pemahaman dasar, proses mempelajari arsitektur cloud secara menyeluruh, mulai dari komponen seperti S3, VPC, VPN Gateway, hingga mekanisme routing dan pengaturan keamanan, membutuhkan waktu adaptasi yang cukup intensif. Setiap layanan memiliki terminologi, karakteristik, dan konfigurasi spesifik yang harus dipahami secara teori maupun praktik. Selain itu, menyelaraskan antara konsep teoritis yang dipelajari di kampus dengan kebutuhan implementasi nyata di lingkungan produksi juga menjadi tantangan tersendiri. Praktik terbaik terkait keamanan, otomasi, dan desain jaringan seringkali berbeda ketika dihadapkan dengan kondisi operasional perusahaan, sehingga penyesuaian dan pemahaman kontekstual sangat diperlukan. Proses ini membutuhkan ketelitian, konsistensi, serta bimbingan dari engineer senior agar pengetahuan yang diperoleh tidak hanya bersifat akademis, tetapi benar-benar relevan dengan kebutuhan kerja. Tantangan ini wajar

terjadi karena ruang lingkup cloud computing sangat luas dan terus berkembang, sehingga mahasiswa atau intern dituntut untuk memiliki kemampuan adaptasi dan kemauan belajar yang tinggi.

3. **Tantangan dalam Penyusunan Laporan Perusahaan**

   Tantangan lain yang dihadapi selama masa magang adalah dalam proses penyusunan *Monthly Report* dan *Preventive Maintenance Report* yang menjadi bagian penting dari aktivitas monitoring layanan. Penyusunan laporan ini membutuhkan kemampuan untuk mengolah berbagai jenis data teknis, seperti metrik availability, performa sistem, alert yang muncul, kapasitas sumber daya, serta rekapan log aktivitas. Setiap data memiliki konteks dan karakteristik yang berbeda sehingga proses analisisnya harus dilakukan secara teliti agar informasi yang dihasilkan akurat dan dapat dipertanggungjawabkan. Selain itu, penulis harus mampu menafsirkan hasil pengecekan teknis tersebut ke dalam format laporan yang sesuai dengan standar perusahaan, yang tidak hanya bersifat informatif tetapi juga mudah dipahami oleh pihak manajemen maupun klien. Di sisi lain, tingkat ketelitian yang tinggi sangat diperlukan karena laporan monitoring berfungsi sebagai dasar pengambilan keputusan, termasuk dalam hal tindak lanjut permasalahan atau potensi gangguan (finding) yang teridentifikasi. Kesalahan kecil dalam pencatatan, analisis, atau penyajian data dapat berdampak pada interpretasi yang keliru dan mempengaruhi rekomendasi teknis. Oleh karena itu, proses penyusunan laporan biasanya juga melibatkan validasi dari engineer senior untuk memastikan seluruh informasi telah disajikan secara lengkap, tepat, dan sesuai standar operasional. Tantangan ini memberikan pengalaman penting dalam bekerja secara sistematis dan detail pada konteks laporan teknis profesional.

3.3.3 Solusi Atas Kendala yang Ditemukan

Selama menjalani masa magang, sejumlah kendala dan tantangan muncul dari berbagai sisi, baik teknis maupun non-teknis, yang menuntut penulis untuk terus beradaptasi dan berpikir kritis dalam menghadapi setiap situasi. Kendala teknis muncul misalnya pada konfigurasi layanan cloud, pengaturan jaringan, atau proses troubleshooting, sementara kendala non-teknis berkaitan dengan manajemen waktu, komunikasi dalam tim, serta adaptasi terhadap budaya kerja perusahaan. Untuk memastikan kegiatan pembelajaran dan pelaksanaan tugas tetap berjalan dengan baik, diperlukan usaha aktif dalam menemukan solusi, baik melalui pencarian informasi secara mandiri, eksplorasi dokumentasi teknis, maupun dengan meminta arahan dan bimbingan dari supervisor serta dukungan dari rekan satu tim. Setiap solusi yang diterapkan bersifat kontekstual, menyesuaikan dengan kondisi permasalahan yang dihadapi, sekaligus diarahkan untuk meningkatkan kompetensi profesional dan efisiensi kerja secara berkelanjutan. Proses ini tidak hanya membantu menyelesaikan masalah yang muncul, tetapi juga memperkuat kemampuan analisis, pemecahan masalah, serta keterampilan kolaborasi di lingkungan kerja yang dinamis. Dengan pendekatan ini, penulis mampu mengoptimalkan pengalaman magang sekaligus memperoleh pemahaman yang lebih mendalam mengenai praktik kerja profesional. Uraian lebih lanjut mengenai penyelesaian kendala-kendala tersebut dijelaskan pada bagian berikut.

1. **Implementasi Cloud**

   Untuk mengatasi kendala teknis yang muncul selama konfigurasi infrastruktur cloud, langkah awal yang saya lakukan adalah memperdalam pemahaman mengenai arsitektur jaringan perusahaan secara menyeluruh, serta menelaah parameter teknis yang harus disesuaikan dengan standar AWS. Proses ini dimulai dengan meninjau ulang dokumentasi resmi AWS terkait Site-to-Site VPN, pengaturan IPsec, dan konfigurasi VPC, lalu membandingkannya dengan konfigurasi perangkat lokal yang tersedia di lingkungan

kerja. Dengan cara ini, saya dapat mengidentifikasi perbedaan kritis seperti algoritma enkripsi yang digunakan, versi IKE, SA lifetime, dan kebijakan routing yang dapat menyebabkan tunnel IPsec gagal terhubung. Setelah perbedaan parameter ini teridentifikasi, saya melakukan serangkaian uji coba konfigurasi secara bertahap. Pertama, memastikan fase IKE dan IPsec negotiation berhasil, kemudian memeriksa log perangkat secara detail untuk menemukan titik kegagalan pada proses handshake. Langkah berikutnya melibatkan penyesuaian parameter secara iteratif, dengan selalu berkonsultasi bersama supervisor maupun engineer senior untuk memastikan setiap perubahan tidak menimbulkan konflik konfigurasi baru. Selain troubleshooting VPN, saya juga melakukan verifikasi menyeluruh terhadap konfigurasi VPC Endpoint yang digunakan untuk koneksi privat ke layanan Amazon S3. Proses ini meliputi pengecekan ulang route table, memastikan policy endpoint telah mengizinkan akses yang sesuai, sinkronisasi ulang credential dan permission IAM, serta melakukan pengujian koneksi di environment terisolasi agar alur data berjalan lancar tanpa melalui internet publik. Pendekatan kombinasi antara eksplorasi mandiri, pemanfaatan dokumentasi teknis, observasi log sistem, serta diskusi dengan tim membuat proses troubleshooting menjadi lebih terstruktur dan sistematis. Strategi ini tidak hanya meminimalkan risiko error berulang, tetapi juga meningkatkan pemahaman saya terhadap karakteristik jaringan perusahaan dan memastikan implementasi infrastruktur cloud berjalan stabil, aman, dan sesuai standar operasional.

2. **Pemahaman terkait Cloud**

   Untuk mengatasi keterbatasan pemahaman awal mengenai cloud computing, saya membangun fondasi belajar yang lebih terstruktur melalui kombinasi dokumentasi resmi, hands-on lab, dan diskusi teknis dengan engineer senior. Proses pembelajaran dimulai dari

memahami layanan inti seperti S3, VPC, IAM, VPN Gateway, serta mekanisme routing antar jaringan cloud. Setelah fondasi ini cukup kuat, saya memperluas pengetahuan ke aspek yang lebih kompleks, termasuk praktik keamanan, desain high availability, optimasi performa, serta skenario integrasi antar layanan cloud. Setiap konsep yang dipelajari langsung saya praktikkan pada environment pengujian agar pengalaman belajar tidak hanya bersifat teoritis, tetapi juga relevan dengan kondisi operasional nyata. Saya juga memanfaatkan tutorial internal perusahaan, playbook konfigurasi, dan dokumentasi best practice untuk memahami standar implementasi yang berlaku di lingkungan profesional, yang sering kali berbeda dari materi akademik. Proses ini diperkuat dengan meminta umpan balik secara aktif ketika melakukan konfigurasi, terutama untuk memastikan langkah yang diambil sesuai dengan kebijakan keamanan dan kebutuhan operasional klien. Pendekatan belajar iteratif seperti ini membantu mempercepat adaptasi, meningkatkan konsistensi pemahaman, serta memastikan kompetensi yang saya bangun dapat diaplikasikan secara efektif pada proyek-proyek nyata. Dengan demikian, kendala minimnya familiaritas awal terhadap teknologi cloud dapat diatasi secara bertahap melalui kombinasi belajar mandiri, praktik langsung, dan bimbingan profesional yang terarah.

3. **Penyusunan Laporan Perusahaan**

Dalam menghadapi tantangan penyusunan Monthly Report dan Preventive Maintenance Report, saya menerapkan pendekatan kerja yang lebih sistematis dan berbasis proses. Langkah pertama adalah membuat alur kerja pengumpulan data yang jelas dan terstruktur. Sumber data yang digunakan meliputi dashboard monitoring, log server, sistem alerting, serta laporan internal yang tersedia. Untuk memastikan tidak ada data yang terlewat, saya menggunakan checklist sebagai panduan langkah demi langkah. Metode ini

membuat proses pengumpulan data menjadi lebih efisien, sekaligus meminimalkan risiko ketidakkonsistenan atau kesalahan input. Selanjutnya, saya menyusun template analisis untuk metrik utama, seperti availability, performa sistem, kapasitas, dan temuan (finding). Template ini membantu mempercepat proses interpretasi data, memastikan laporan memiliki format yang konsisten, dan memudahkan pihak manajemen atau klien dalam memahami isi laporan. Selain itu, saya memanfaatkan sesi diskusi dengan engineer senior untuk menafsirkan hasil pengecekan teknis, sehingga laporan yang dihasilkan tidak sekadar angka, tetapi juga menyertakan analisis kontekstual dan rekomendasi tindakan yang relevan. Proses cross-check dan validasi menjadi langkah penting sebelum laporan diserahkan, guna memastikan tidak ada kesalahan pencatatan, penulisan, maupun analisis. Dalam beberapa kasus, saya melakukan revisi berdasarkan umpan balik tim agar penyajian informasi lebih jelas, akurat, dan mudah dipahami. Dengan menerapkan pendekatan kerja yang konsisten ini, tantangan dalam penyusunan laporan teknis dapat diatasi secara bertahap, sehingga saya mampu meningkatkan ketelitian, akurasi, serta kualitas keseluruhan dari laporan yang dihasilkan. Strategi ini juga menanamkan disiplin kerja, pemahaman analisis data, serta kemampuan menyajikan informasi secara profesional dan sistematis.